KONVENS 2014

proceedings

October, 8-10 2014

University of Hildesheim,
Germany

Stiftung Universität Hildesheim
2003

The 12th edition of Konvens, the bi-annual German conference is sponsored by the German Society for Computational Linguistics (GSCL), by the special interest group on computational linguistics within the German Linguistics Society (DGfS-CL), and by the Austrian Society for Artificial Intelligence (ÖGAI). We also received support from the German Institute for International Pedagogic Research (Deutsches Institut für Internationale Pädagogische Forschung, DIPF), from the Common Language Resources and Technoloy Infrastructure project CLARIN-D and from the University of Hildesheim.

GSCL

dgfs
Deutsche Gesellschaft
für Sprachwissenschaft

ÖGAI

CLARIN-D

Universitätsverlag Hildesheim
Marienburger Platz 22
31141 Hildesheim
verlag@uni-hildesheim.de
ISBN 10: 3-934105-46-7
ISBN 13: 978-3-934105-46-1
Hildesheim 2014

This is an electronic publication, it is available via `http://www.uni-hildesheim.de/konvens2014`

PROCEEDINGS OF THE 12TH EDITION OF THE KONVENS CONFERENCE

Josef Ruppenhofer and Gertrud Faaß (eds.)

Hildesheim, Germany
October 8 – 10, 2014

Dear Participants of KONVENS 2014,
dear Reader,

it is our pleasure to welcome all attendees of the 12th KONVENS, Konferenz zur Verarbeitung Natürlicher Sprache, and of the co-located workshops in Hildesheim and to make the texts of all contributed papers available to our readership.

Being organized jointly by the German and Austrian community in the field of computational linguistics, as represented by the professional institutions GSCL, Gesellschaft für Sprachtechnologie und Computerlinguistik, ÖGAI, Österreichische Gesellschaft für Artificial Intelligence, and the section on computational linguistics of DGfS, Deutsche Gesellschaft für Sprachwissenschaft, KONVENS has been throughout its history, and continues to be, a privileged forum for the exchange of new ideas, approaches and techniques in the field, bringing together theoretical research, applied work and evaluations.

The 2014 issue of KONVENS is even more a forum for exchange: its main topic is the interaction between Computational Linguistics and Information Science, and the synergies such interaction, cooperation and integrated views can produce. This topic at the crossroads of different research traditions which deal with natural language as a container of knowledge, and with methods to extract and manage knowledge that is linguistically represented is close to the heart of many researchers at the Institut für Informationswissenschaft und Sprachtechnologie of Universität Hildesheim: it has long been one of the institute's research topics, and it has received even more attention over the last few years.

The main conference papers deal with this topic from different points of view, involving flat as well as deep representations, automatic methods targeting annotation and hybrid symbolic and statistical processing, as well as new Machine Learning-based approaches, but also the creation of language resources for both machines and humans, and methods for testing the latter to optimize their human-machine interaction properties. In line with the general topic, KONVENS-2014 focuses on areas of research which involve this cooperation of information science and computational linguistics: for example learning-based approaches, (cross-lingual) Information Retrieval, Sentiment Analysis, paraphrasing or dictionary and corpus creation, management and usability.

The workshops hosted at this iteration of KONVENS also reflect the interaction of, and common themes shared between, Computational Linguistics and Information Science: a focus on on evaluation, represented by shared tasks on Named Entity Recognition (GermEval) and on Sentiment Analysis (GESTALT); a growing interest in the processing of non-canonical text such as that found in social media (NLP4CMC) or patent documents (IPaMin); multi-disciplinary research which combines Information Science, Computer Aided Language Learning, Natural Language Processing, and E-Lexicography with the objective of creating language learning and training systems that provide intelligent feedback based on rich knowledge (ISCALPEL).

As organizers, we are grateful to all contributors and to the invited speakers, Janyce Wiebe, Jacques Savoy, Hinrich Schütze and Benno Stein. We would also like to express our gratitude to all those who lent their time and expertise to the reviewing process, sometimes at short notice. A big thank you is also owed to the organizers of the workshops that KONVENS is hosting this year and to the presenter of Friday's tutorial. Finally, we want to specifically acknowledge all the locals who made the conference and this volume happen: Gertrud Faaß and Josef Ruppenhofer, Fritz Kliche and Stefanie Elbeshausen, Julia Jürgens and Gabriele Irle, and the student assistants Max Billmeier, Melanie Dick, Julian Hocker, Victoria Wandt, and Marie Zollmann.

Christa Womser-Hacker and Ulrich Heid

CONTENTS

On the effect of word frequency on distributional similarity

Christian Wartena
Hochschule Hannover
Department of Information and Communication
Expo Plaza 12, 30539 Hannover, Germany
Christian.Wartena@hs-hannover.de

Abstract

The dependency of word similarity in vector space models on the frequency of words has been noted in a few studies, but has received very little attention. We study the influence of word frequency in a set of 10 000 randomly selected word pairs for a number of different combinations of feature weighting schemes and similarity measures. We find that the similarity of word pairs for all methods, except for the one using singular value decomposition to reduce the dimensionality of the feature space, is determined to a large extent by the frequency of the words. In a binary classification task of pairs of synonyms and unrelated words we find that for all similarity measures the results can be improved when we correct for the frequency bias.

1 Introduction

Distributional similarity has become a widely accepted method to estimate the semantic similarity of words by analyzing large amounts of texts. The basic idea of distributional similarity is that words occurring in similar contexts have a similar meaning. However, implementations of the idea differ by choosing different features to represent the context of a word, by different approaches to determine feature weights and by different similarity measures to compare the contexts. A number of recent studies (Bullinaria and Levy, 2007; Bullinaria and Levy, 2012; Kiela and Clark, 2014) shed light on the influence of a number of design choices on the performance of distributional similarity in various tasks.

Usually it is assumed that a minimum number of occurrences of a word is needed to build a reliable distributional model of the word. Ferret (2010) e.g. observes that results become significantly worse when less than 100 occurrences of a word are available.

Besides the fact, that a minimum number of occurrences is required to get any reliable information about a word at all, another problem is the fact that similarity measures tend to have a frequency bias. Weeds et al. (2004) evaluated a number of combinations of feature weighting schemes and similarity measures and found that each combination has a frequency bias: when we look for the words that are most similar to a given word, most measures prefer more frequent words. A few measures have a bias towards less frequent words or words with a frequency similar to the target word. The larger the difference in frequency between the most frequent and the least frequent word included in some test set is, the stronger the influence of the frequency bias will become. Thus the frequency bias poses a further burden upon the inclusion of infrequent words in a task.

Experiments in which the quality of distributional methods is tested usually involve many words for which information in lexical resources is available and that occur quite frequently in large corpora. However, if we look at the distribution of words in a corpus the vast majority of words occurs only very rarely. E.g. according to Barroni et al. (2009) the large ukWaC corpus contains about $1.529 \cdot 10^6$ different word forms tagged as common noun by the TreeTagger (Schmid, 1995), $1.414 \cdot 10^6$ of which occur less than 20 times. In most studies a minimum number of 20, 100 or sometimes even 1000 occurrences of a word is assumed to be necessary to compute reliable similarities. Thus for most words distributional similarity cannot be used.

One of the practical applications of distributional similarity that is often mentioned, is automatic updating and extension of a thesaurus with new terminology (Crouch, 1990; Curran and Moens, 2002; Turney and Pantel, 2010). One of the typical properties of new terminology is, that we do not yet have many occurrences of the terms in our corpus. Thus, the methods developed are in fact not suited for this useful application. For many other applications a similar situation holds. Thus, if we want to make distributional similarity more useful for applications, we need to improve the way we can deal with infrequent words. Before we can improve methods for infrequent words, we need to better understand, how various implementations of distributional similarity depend on word frequency.

In the present paper we study the frequency bias in more detail for 6 different similarity methods. First we compare the methods on a standard task, the synonymy task that has been included in the Test of English as a Foreign Language (TOEFL). In two experiments we then compute the similarity of pairs of English words with different frequencies using the ukWaC corpus. In the first experiment we compute the similarity of 10 000 arbitrary word pairs in which the frequency of the first word is kept constant and the frequency of the second word varies. In this experiment we can observe for each method, how the similarity depends on the word frequency. In the second experiment we investigate the behavior of the methods in a task in which 10 000 pairs of synonyms and non-synonyms have to be ranked. For this test a set of word pairs was used that was selected from Wordnet without putting restrictions on the frequency of the involved words in some corpus. Finally, we show how much the results of each method can be improved by taking into account the similarity expected on the base of the frequency of the words.

In section 2 we discuss related work. In section 3 we present the details of the distributional methods compared. Section 4 describes the data and the experiments used to study the influence of word frequency on word similarity for each method. The results of the experiments are given and discussed in section 5.

2 Related Work

Despite the importance of being able to deal with infrequent words, the problem has received very little attention. Ferret (2010) computes the similarity of huge amounts of word pairs in order to extract synonyms from a mid-sized corpus. He systematically investigates the results for low frequent, mid frequent and highly frequent words using cosine similarity and pointwise mutual information for feature weighting. He concludes that the results for the low frequent words (less than 100 occurrences) are useless.

Kazama et al. (2010) propose a method to extract word pairs with a high likelihood to be semantically related. They argue that, given two word pairs with the same (distributional) similarity, the pair with more frequent words should become a higher likelihood to be semantically related. The rationale behind this is, that we have more observations and thus a more reliable estimation of the similarity. Thus their method becomes robust when dealing with sparse data. However, if the task is not to extract pairs of related words from a corpus, but to decide whether two given words are related or not, we do not want to decide that the words are unrelated just by the fact that we do not have enough observations.

Already Patel et al. (1998) found a clear correlation between the frequency of words and their similarity. However, they were more interested in corpus size than in word frequencies. As mentioned above, Weeds et al. (2004) study the frequency bias for several methods in the case that similar words for a given word are sought. They do not consider the direct dependency of the similarity values on the frequency of the words, but study the frequency of the most similar words that are found, in relation to the frequency of the target word.

In two previous studies we investigated the dependency of the similarity of a pair of two words on the frequency of these words (Wartena, 2013a; Wartena, 2013b). In these studies we could improve the results for two different tasks substantially by using the difference between the similarity predicted by the frequency of the words and the actual measured similarity. However, in both studies only random indexing was considered. Random indexing is an efficient method for dimensionality reduction using random projection of features into a small size feature space. However, the results using random indexing are probably not as good as those obtained with other dimensionality reduction methods and the method is not very popular in the field of distributional semantics. In the present study we extend the previous studies and also include other similarity measures and different feature weighting schemes.

3 Overview of used similarity methods

The computation of distributional similarity of two words always involves two steps: first distributional models for each word are built by collecting information about the contexts in which the words occur. Subsequently these models are compared to access the similarity of the words. As the models are usually vectors in a high dimensional feature space or probability density distributions a number of well known similarity measures can be used. Also for the construction of the models a number of choices has to be made: it has to be decided which context information is used; several possibilities exist for the weighting of the context features and finally some dimensionality reduction techniques might be applied.

In order to study the effects of word frequency on distributional similarity it is not feasible to explore all possible combinations of choices for context features, weighting method, dimensionality reduction technique and similarity measure. Fortunately, a few recent studies have investigated the effect of various design choices and combinations of choices for different tasks and corpora in a systematic way (Bullinaria and Levy, 2007; Bullinaria and Levy, 2012; Kiela and Clark, 2014). In the present study we will include a number of methods that turned out to be successful in the mentioned studies.

In the simplest case we use just the frequencies of context words as a feature vector in combination with cosine similarity. We will refer to this configuration as *plain_cos*. We also include the same method in combination with the Jensen-Shannon divergence, which we call *plain_jsd*. A successful weighting scheme turned out to be pointwise mutual information (PMI) between a word and a feature. As it makes no sense to use an information theoretical measure like Jensen-Shannon divergence (JSD) for weighted features, we use PMI only in combination with cosine similarity and refer to this combination as *plain_pmi*. We also consider the variant where the feature space of the last method is reduced using singular value decomposition (*svd*). Alternatively, we use random indexing to reduce the feature space. We use random indexing both in combination with cosine similarity (*ri_cos*) and with JSD (*ri_jsd*)

In the following we will discuss the various parameters for each configuration in more detail.

3.1 Context features

As context features we use the lemmata of words in the context window of the target word. Sometimes a combination of a word and its syntactic relation to the target word is used. However, it is not clear whether inclusion of syntactic dependencies systematically improves the quality of the feature vectors (Giesbrecht, 2010; Kiela and Clark, 2014). Both Bullinaria and Levy (2012) and Kiela and Clark (2014) show that lemmatization always improves the results, though both studies do not agree about the effect of stemming. Here we use in all cases the lemmata as context features, but we compute the context models for surface forms of the words. Thus, we never lemmatize the words in the test sets. For the method that uses singular value decomposition (SVD) we have to include context vectors of words that are not part of the test. For these additional words we use lemmata as well.

Inclusion of function words and other highly frequent words put a heavy load on all subsequent computations and might even have a negative effect on the performance (Bullinaria and Levy, 2012). Thus we decided to exclude all closed-class words (determiners, conjunctions, prepositions, etc.). Furthermore we exclude all words from a small standard stop word list (taken from Lucene). After removal of these words we take two words to the left and to the right of each word within the same sentence as context features.

Very infrequent words do not contribute very much to the context vectors. Thus, after selecting the context words, we remove those words that fall outside a given frequency range in the corpus. For all experiments we use the ukWaC corpus (Baroni et al., 2009). For the conditions *plain_cos*, *plain_jsd*, *plain_pmi* and *svd* we kept only words that occur at least 5000 times and at most 1 000 000 times in this corpus in the first experiment. This gives us 16 617 context words that are used as features. In the random word pair experiment and in the synonym ranking task (both involving much more words for which context vectors have to be computed) we kept words occurring at least 10 000 times and at most 1 000 000 times, resulting in a set of 10 800 context features. For random indexing using much more words is no problem and also improved the results in preliminary experiments. Thus we take words in the frequency range from 5 to 1 000 000 occurrences, resulting in 935 405 different features.

3.2 Feature weighting

Positive Pointwise Mutual Information (PPMI) is a popular feature weighting scheme and it was shown both in the studies of Bullinaria and Levy (2012) and of Kiela and Clark (2014) that PPMI in combination with several similarity measures gives optimal results. PPMI is defined as the maximum of 0 and the pairwise mutual information. We use the PPMI for feature weighting in *plan_pmi* and *svd*. For all other configurations raw feature counts are used.

3.3 Dimensionality Reduction

Given the huge amount of different words that can appear in the context of a word, we always will end up with very high dimensional and very sparse feature spaces. Therefor, often some form of dimensionality reduction technique is used. Moreover, techniques like singular value decomposition (SVD) will find the most important underlying factors determining the use of a word and separate them from less important factors that are probably not related to the meaning of the word.

We use SVD in one condition. First we construct the full co-occurrence matrix of $16\,617 \times 16\,617$ with almost $78 \cdot 10^6$ non-zero entries for the TOEFL-Test. In the random word pair experiment and in the synonym ranking task, in which we used less context features, the size of the matrix is $20\,788 \times 10\,800$ and $18\,145 \times 10\,800$, respectively. Subsequently we compute the positive pairwise mutual information (PPMI) for each word/feature pair and adjust the values in the co-occurrence matrix. Using the svdlib library from the semantic vectors package (`https://code.google.com/p/semanticvectors/`) we compute matrices U, S and V such that $M = USV^T$, where U and V are orthogonal matrices and S is a diagonal matrices of the singular values of M where M is the original word-lemma matrix of PPMI values. We now can use the rows of US as feature vectors for the words. By truncating the rows we can restrict the comparison of the feature vectors to the most important principle components. We will use the first 5 000 components in the experiments below.

Bullinari and Levy (2012) found that results can be improved when the influence of the first components is reduced. To do so, they either simply leave out the first n principal components or reduce the weights of the most important features by using the matrix $X = US^P$ instead of $X = US$, where P is called Caron's P. Following Bullinaria and Levy we use a value of $0,25$ for Caron's P.

An alternative way to reduce the number of dimensions is random projection. Random projection was introduced for distributional similarity by Karlgren and Sahlren (2001) under the name random indexing. Random indexing has the great advantage that it is computationally very cheap and there is no need to build the full co-occurrence matrix. Each feature is represented by a n-dimensional vector with a 1 at k random positions and 0 at all other positions. In the following we set $n = 10\,000$ and $k = 10$. This vector can be seen as a *fingerprint* of the feature. The context of a word is represented by the sum of the vectors of all words found in its context.

The advantage of this method is that the number of dimensions can be chosen freely and no additional computation for dimension reduction is needed. Random Indexing is not used very widely and not included in a number of overview studies. However, Random Indexing was shown to yield competitive results at the 2013 Semeval phrasal semantics task (Korkontzelos et al., 2013).

3.4 Similarity Measures

Various similarity measures have been used for distributional semantics. If we use vectors of simple word occurrences cosine similarity is an obvious choice. In the studies of Bullinaria and Levy (2007) and of Kiela and Clark (2014) this measure performed very well in combination with various weighting schemes and for various tasks. We use cosine similarity for the conditions *plain_cos*, *plain_pmi*, *svd* and *ri_cos*.

Alternatively, we can see the distributional model of a word as a probability distribution over words that can appear in the context of that word. Then it is natural to use a information theoretic similarity measure. Since we usually want a symmetric measure the most commonly used measure is the Jensen Shannon Divergence (JSD). JSD was shown to give also very good results, especially in combination with unweighted features (Bullinaria and Levy, 2007; Kiela and Clark, 2014). We use JSD in the conditions *plain_jsd* and *ri_jsd*.

4 Data and Experiments

For all experiments described below we compute the context vectors on the ukWaC-Corpus (Baroni et al., 2009). First we examine how each method performs on the widely used TOEFL synonym test. Then we study the influence of word frequency on a set of 10 000 randomly selected word pairs. Finally, we compare the methods is a test in which 10 000 pairs of synonyms and unrelated words have to be ranked.

4.1 TOEFL Synonym Test

One of the most widely used tests to evaluate semantic similarity is the synonymy task that has been included in the Test of English as a Foreign language (TOEFL) (Landauer and Dumais, 1997). The test consists of 80 words and for each word four potential synonyms. In total 391 words are involved. The task is to decide which of the four candidates is the synonym. When we choose always the candidate with the largest distributional similarity, we see how well the chosen measure reflects semantic similarity. We include this test to get an impression of the quality of the methods included in the following experiments.

4.2 Random Word Pairs Experiment

For our first experiment to access the behavior of the similarity measures for words with different numbers of observations we have extracted 10 000 word pairs from the ukWaC corpus in the following way: we selected 100 words that occur at least 1000 and at most 1005 times in the corpus and that have a part-of-speech tag from an open word class, consist of at least 3 letters and do not contain special characters. These words are used as the first component of the word pairs. Next we randomly selected 10 000 words from the corpus with the same criteria but in a frequency range from 5 to 1 000 000. This was done by ordering all words according to their frequency and picking words with a fixed interval from that list. Thus the frequency distribution of these words is the same the that of all words in the corpus. Finally, these 10 000 words were assigned to the previously selected words to obtain 10 000 word pairs.

For these pairs we compute the similarity for each method. In order to see to what degree the similarity of a pair of words depends on the frequency of the words, we predict the similarity for each pair by taking the average similarity of 100 word pairs with the same or almost the same frequency. To do so, we order the all pairs according to the frequency of their second word (the frequency of the first word of each pair is always the same) [1]. Now we compute the average similarity of 50 pairs before and 50 pairs after the pair under consideration. Finally, we compute the coefficient of determination as follows:

$$R^2 = 1 - \frac{\sum_i (sim_i - \overline{sim_i})^2}{\sum_i (sim_i - \overline{sim})^2} \qquad (1)$$

where sim_i is the found similarity of the i-th pair, $\overline{sim_i}$ predicted similarity (moving average) for that pair and $\overline{sim}$ is the average similarity of all pairs.

[1] In case the the frequencies of the second word are identical, we order the pairs alphabetically. However, any other ordering did not influence the results presented below within the precision of two decimals.

4.3 Synonym Ranking Task

In the last experiment we want to investigate how much each method can be improved when we correct for the frequency bias.

Association tasks in which a word has to be associated with one word from a small list of words, have been used in many studies on distributional similarity. However, for some applications we are confronted with a completely different situation. A possible application is to add terminology extracted from a corpus to an existing thesaurus. Each term now is either a synonym of one of many thesaurus terms, or it is new concept for which no synonyms are present in the thesaurus. In fact for each pair we have to decide whether the words are synonym or not.

Another problem of the TOEFL test and some other tests is the small size: the TOEFL set has 80 pairs, the Rubinstein-Goodenough set consists of 65 pairs (Rubenstein and Goodenough, 1965), the Finkelstein's WordSim-353 set consists of 353 pairs (Finkelstein et al., 2001). Moreover, some data focus more on word associations than on synonymy. Finally, many larger generated data sets have a strong frequency bias. E.g. for their Wordnet Based Similarity Test, with questions similar to those from the TOEFL test, Freitag et al. (2005) have chosen only words occurring at least 1000 times in the North American News corpus (about 1 billion words); for a lexical entailment task Zhitomirsky- Geffet and Dagan (2009) use only words occurring at least 500 times in a 18 Million word corpus; for their distance comparison Bullinaria and Levy (2007) select 200 words "that are well distributed in the corpus" and the test set for two word phrases constructed by Mitchell and Lapata (2010) consists of phrases occurring at least 100 times in the British National Corpus (100 million words).

In an application in which e.g. new terminology has to be mapped onto an existing thesaurus, we do not want to exclude infrequent words. In contrary: the new and rare words are the most interesting ones. Therefor we use in our last experiment a data set of almost 10 000 word pairs in which no infrequent words are excluded. We have used this data set before in a similar experiment (Wartena, 2013a)[2]. This list of pairs consists of single words taken from Wordnet (Miller, 1995) that occur at least two times in the British National Corpus and at least once in the ukWaC corpus. The data set contains 849 pairs for which the Jaccard coefficient of the sets of Wordnet senses of the words is at least 0.7. These word pairs are considered to be synonyms. As non-synonyms 8967 word pairs are included that share no senses.

The task now is to decide for each pair, whether the words are synonym or not. We evaluate similarity measures for this task by ranking the pairs according to the similarity of the words. An ideal ranking, of course, would put all synonyms on top. To what extent this is the case is indicated by the area under the ROC curve (AUC).

For the pairs in this data set we also want to predict the similarity using the word frequency. The situation is a bit more complicated than before, since the frequency of both words is variable. Here we follow our previous finding that the similarity is determined mostly by the frequency of the least frequent word (Wartena, 2013b). We thus take the moving average of the similarity when the pairs are ordered according to the minimum of the word counts as prediction. Finally, we rank the pairs according to their residual values, assuming that a pair is likely to be semantically related if the observed distributional similarity is larger than we would expect from the frequency of the words.

5 Results

Though our implementation of random indexing is not exactly the same as that described by Karlgren and Sahlren (2001) (e.g. we so not use lower weights for more distant words) and though we use a different corpus, we get the same result on the TOEFL synonym task. Best results are obtained using SVD. However, the results fall clearly back behind those obtained by Bullinaria and Levy (2012), despite the fact that we roughly made the same choices for all parameters.

The results of the random word pair experiment are given in Table 2. The similarities based on SVD are almost independent of the frequency of

[2] The data set is available at http://nbn-resolving.de/urn:nbn:de:bsz:960-opus-4077.

Table 1: Results of 6 different distributional similarity methods on the TOEFL synonym task using the ukWaC- Corpus

Method	Fraction correct
plain_cos	0.675
plain_jsd	0.688
plain_pmi	0.788
svd	0.863
ri_cos	0.725
ri_jsd	0.650

Table 2: Dependency of 6 different distributional similarity methods for 10 000 random pairs of words on the frequency of the words.

Method	R^2
plain_cos	0.20
plain_jsd	0.77
plain_pmi	0.47
svd	0.10
ri_cos	0.39
ri_jsd	0.87

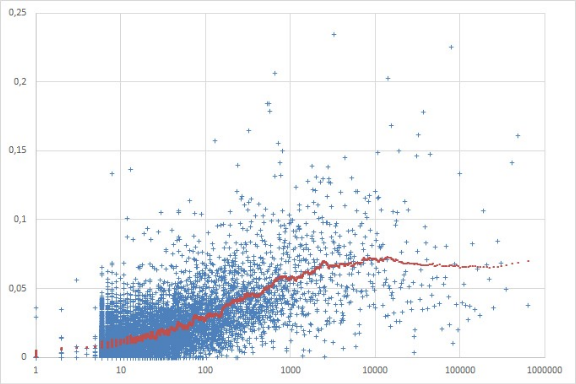


Figure 1: Similarity of wordpairs using the *plain_pmi* configuration in dependence of the frequency of the second word. The first word in each pair always occurs between 1000 and 1005 times in the corpus. The y-axis is represents the cosine similarity, the x-axis the number of occurrences of the second word. The solid (red) line is the moving average in a window of 100 word pairs.

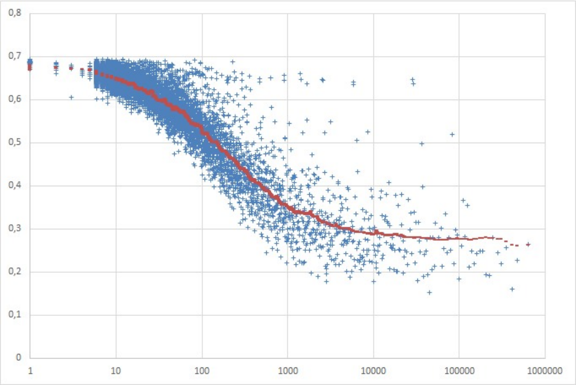


Figure 2: Similarity of wordpairs using the *ri_jsd* configuration in dependence of the frequency of the second word. The y-axis is represents the Jensen-Shanon divergence of the context vectors of the words, the x-axis the number of occurrences of the second word. The solid (red) line is the moving average in a window of 100 word pairs

the words. Especially the similarities computed using the Jensen-Shannon divergence are highly determined by the frequency. Interestingly we see that the R^2 value for *plain_pmi* is much larger than for *plain_cos*. The dependency of the similarity on the frequency of the second word is illustrated exemplary in Figure 1 and 2 for the configurations *plain_pmi* and *ri_cos*. We see that the moving average for the methods using cosine similarity is roughly logarithmic function of the frequency. For the JSD the moving average follows a kind of asymmetric sigmoid curve.

In the synonym ranking task (Table 3) we do not find any surprises: as in the case of the TOEFL-test the best result is obtained with the *svd*-configuration, the second best with *plain_pmi* and results based on the Jensen-Shannon divergence are worst. The dependency on the word frequency, measured by coefficient of determination, also confirms the results of the previous experiment, though the absolute values are a bit different. Remarkable, however, are the results of the ranking by the residual values. The results of all methods could be improved. The largest improvements, of course, are found for the methods with the largest dependency on the word frequency. The differences between the methods now become much smaller. The methods *svd* and *plain_pmi* now give the same results.

Finally, we also want to know how the 6 methods perform for word pairs involving an infrequent word. Table 4 gives the results for all pairs with at least one word occurring less than 100 times in the ukWaC corpus. We observe that the results for the methods that have a strong fre-

Table 3: Results of 6 different distributional similarity methods on ranking 10 000 pairs of synonyms and non-synonyms task using the ukWaC- Corpus. The first column gives the results of ranking the pairs according to their similarity. The second column shows the dependency of the similarity of the word pairs on their frequency expressed the R^2 value of the moving average. The last column gives the results when the pairs are ranked according to the residual values.

Method	AUC (sim)	R^2	AUC (res)
plain_cos	0.66	0.22	0.77
plain_jsd	0.43	0.86	0.72
plain_pmi	0.67	0.33	0.85
svd	0.81	0.04	0.85
ri_cos	0.60	0.28	0.72
ri_jsd	0.41	0.94	0.70

Table 4: Results of 6 different distributional similarity methods on ranking 1953 pairs of synonyms and non-synonyms from which at least one word occurs less than 100 times in the ukWaC-Corpus. The first column gives the results of ranking the pairs according to their similarity. The second column gives the results when the pairs are ranked according to the residual values.

Method	AUC (sim)	AUC (res)
plain_cos	0.65	0.71
plain_jsd	0.53	0.64
plain_pmi	0.73	0.81
svd	0.80	0.82
ri_cos	0.59	0.65
ri_jsd	0.50	0.61

quency bias is better than the results on the complete data set. This is as expected, since the frequency range is clearly reduced in this subset. When we rank the pairs using the residual values, the results of all methods stay behind those on the complete data set.

6 Discussion

We clearly see that all methods become better when more data are available. However, all methods have the potential to make good predictions for less frequent words. The method using SVD is only slightly worse on the less frequent data. Thus we see that the best methods still give useful results for infrequent words, contradicting the findings of Ferret (2010).

For the cosine similarity and the JSD the dependency on the word frequency can intuitively be understood as follows. The cosine depends only on the dimensions for which both vectors have a non-zero value. If the vectors become less sparse, since we have seen more different contexts, it is not surprising that the cosine tends to become larger. The JSD also depends only on the dimensions for which both vectors have a non-zero value. This can be seen if we rewrite the JSD for two probability density functions p and q as

$$\begin{aligned} \mathrm{JSD}(p,q) &= \tfrac{1}{2}D(p||\tfrac{1}{2}p+\tfrac{1}{2}q) + \tfrac{1}{2}D(q||\tfrac{1}{2}p+\tfrac{1}{2}q) \\ &= \log 2 + \tfrac{1}{2} \sum_{t:p(t)\neq 0 \,\wedge\, q(t)\neq 0} \Big(p(t)\log\big(\tfrac{p(t)}{p(t)+q(t)}\big) \\ &+ q(t)\log\big(\tfrac{q(t)}{p(t)+q(t)}\big)\Big), \qquad (2) \end{aligned}$$

where $D(p,q)$ is the Kullback-Liebler divergence of p and q. The differences between cosine and JSD cannot be explained that easily. If we weight the features using PPMI the influence of words just occurring a few times in the context of a word is reduced. Thus the similarity caused by irrelevant words just randomly occurring in the context of both words when we consider enough data, is reduced. The influence of irrelevant features is further reduced when SVD is used.

Furthermore we see that the dependency on the frequency for the methods using random indexing is larger than for the corresponding plain methods. For random indexing we included much more (infrequent) context words as features. Thus there are more factors that potentially cause the differences.

The data set of the ranking task was first used by Wartena (2013a). When we compare the results with the results presented there, we see that we get exactly the same result for *ri_jsd*, though the configuration is somewhat different: we use 10 000 dimensions and a window of 4 words, whereas Wartena (2013a) used 20 000 dimensions and used all words in the sentence as features. For the *ri_cos* method the results are worse than those presented there. Wartena (2013a) also gives a ranking by the residual values. The results given there are much better than those found here and even slightly better than those found using SVD. The difference between the both studies is, that the modeling in Wartena (2013a) is not based on the frequency of the words but on the number of non-zero values in the feature vectors.

Of course, it would be easy to obtain better results, by using other additional features for the ranking. E.g. the synonyms in the data set tend to have a lower frequency than the unrelated word pairs. Moreover, many synonyms are just spelling variants, that could be detected easily using edit distance or bigram overlap.

7 Conclusions and Future Work

Though the dependency of word similarities in distributional models on their frequencies is already known since a decade, the issue has received little attention. In the present paper we investigated the influence of word frequency on 6 different methods to compute distributional similarity. Thus the paper extends previous work in which only random indexing was considered or in which a frequency bias was observed for various methods but in which the correlation between frequency and similarity was not investigated in more detail.

We find that all tested methods except the one using SVD for dimensionality reduction are strongly dependent on frequency of the words. We find the dependency as well for cosine similarity as for Jensen-Shannon divergence. The dependencies are found consistently on two different data sets. The second data set consist of pairs of synonyms and unrelated words. We have shown that the methods that are strongly dependent on word frequency nevertheless have the potential to discriminate between pairs of synonyms and unrelated words, when we do not use the absolute similarity but the similarity relative to the similarity expected on the base of the word frequency.

The superiority of the method using point wise mutual information for feature weighting, SVD for dimensionality reduction and the cosine as similarity measure for feature vectors was already found in a number of other studies. However, the present study reveals one of the factors that are responsible for the performance differences: the distortion by the word frequencies.

We now could conclude that we know which method to use. However, SVD is computationally demanding and not feasible in all situations. The fact that we have shown that other methods can give similar results when we correct for the frequency bias, encourages us to search for similarity measures and feature weighting schemes that are less sensitive for word frequency. A different direction that we will pursue is smoothing of the feature vectors of infrequent words in order to compensate for the effects of a low number of observations.

References

Marco Baroni, S. Bernardini, A. Ferraresi, and E. Zanchetta. 2009. The WaCky Wide Web: A Collection of Very Large Linguistically Processed Web-Crawled Corpora. *Language Resources and Evaluation 43 (3): 209-226*, 43(3):209–226.

John A. Bullinaria and Joseph P. Levy. 2007. Extracting semantic representations from word co-occurrence statistics: A computational study. (39):510–526.

John A. Bullinaria and Joseph P. Levy. 2012. Extracting Semantic Representations from Word Co-occurrence Statistics: Stop-lists, Stemming and SVD. (44):890–907.

Carolyn J. Crouch. 1990. An approach to the automatic construction of global thesauri. *Information Processing & Management*, 5:629–640.

James R. Curran and Marc Moens. 2002. Improvements in Automatic Thesaurus Extraction. In *Unsupervised Lexical Acquisition: Proceedings of the Workshop of the ACL Special Interest Group on the Lexicon (SIGLAX)*, pages 59–66.

Olivier Ferret. 2010. Testing Semantic Similarity Measures for Extracting Synonyms from a Corpus. In Guy de Pauw, editor, *LREC 2010, Seventh International Conference on Language Resources and Evaluation*, pages 3338–3343.

Lev Finkelstein, Evgeniy Gabrilovich, Yossi Matias, Ehud Rivlin, Zach Solan, Gadi Wolfman, and Eytan Ruppin. 2001. Placing Search in Context: The Concept Revisited. In *Proceedings of the 10th International Conference on World Wide Web*, WWW '01, pages 406–414, New York and NY and USA. ACM.

Dayne Freitag, Matthias Blume, John Byrnes, Edmond Chow, Sadik Kapadia, Richard Rohwer, and Zhiqiang Wang. 2005. New Experiments in Distributional Representations of Synonymy. In *Proceedings of the Ninth Conference on Computational Natural Language Learning*, CONLL '05, pages 25–32, Stroudsburg and PA and USA. Association for Computational Linguistics.

Eugenie Giesbrecht. 2010. Towards a Matrix-based Distributional Model of meaning. In *Proceedings* *of the NAACL HLT 2010 Student Research Workshop*, pages 23–28, Los Angeles, California. ACL.

Jussi Karlgren and Magnus Sahlgren. 2001. From Words to Understanding. In *Foundations of Real-World Intelligence*, pages 294–308. CSLI Publications, Stanford and California.

Jun'ichi Kazama, Stijn de Saeger, Kow Kuroda, Masaki Murata, and Kentaro Torisawa. 2010. A Baysesian Method for Robust Estimation of Distributional Similarities. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 247–356.

Douwe Kiela and Stephen Clark. 2014. A Systematic Study of Semantic Vector Space Model Parameters. In *2nd Workshop on Continuous Vector Space Models and their Compositionality (CVSC)*.

Ioannis Korkontzelos, Torsten Zesch, Fabio Massimo Zanzotto, and Chris Biemann. 2013. SemEval-2013 Task 5: Evaluating Phrasal Semantics. In *Proceedings of the 7th International Workshop on Semantic Evaluation (Semeval 2013)*.

Thomas K. Landauer and Susan T Dumais. 1997. A solution to Plato's problem: the latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review*, 104(2):211–240.

George A. Miller. 1995. WordNet: A Lexical Database for English. *Communications of the ACM*, 38(11):39–41.

Jeff Mitchell and Mirella Lapata. 2010. Composition in distributional models of semantics. *Cognitive science*, 34(8):1388–1429.

Malti Patel, John A. Bullinaria, and Joseph P. Levy. 1998. Extracting semantic representations from large text corpora. In *4th Neural Computation and Psychology Workshop, London, 9–11 April 1997*, pages 199–212.

Herbert Rubenstein and John B. Goodenough. 1965. Contextual Correlates of Synonymy. *Commun. ACM*, 8(10):627–633.

Helmut Schmid. 1995. Improvements in Part-of-Speech Tagging with an Application to German. In *Proceedings of the ACL SIGDAT-Workshop*.

Peter D. Turney and Patrick Pantel. 2010. From Frequency to Meaning: Vector Space Models of Semantics. *Journal of Artificial Intelligence Research*, (37):141–188.

Christian Wartena. 2013a. Distributional similarity of words with different frequencies. In *Proceedings of the 13th Dutch-Belgian Workshop on Information Retrieval*.

Christian Wartena. 2013b. HsH: Estimating semantic similarity of words and short phrases with frequency normalized distance measures. In *Proceedings of the 6th International Workshop on Semantic Evaluation (SemEval 2012), Atlanta, Georgia, USA*.

Julie Weeds, David J. Weir, and Diana McCarthy. 2004. Characterising Measures of Lexical Distributional Similarity. In *COLING 2004, Proceedings of the 20th International Conference on Computational Linguistics*.

Maayan Zhitomirsky-Geffet and Ido Dagan. 2009. Bootstrapping Distributional Feature Vector Quality. *Computational Linguistics*, 25(3):435–461.

A Language-independent Sense Clustering Approach for Enhanced WSD

Michael Matuschek*, Tristan Miller*, and Iryna Gurevych*†

*Ubiquitous Knowledge Processing Lab (UKP-TUDA)
Department of Computer Science, Technische Universität Darmstadt
http://www.ukp.tu-darmstadt.de/
†Ubiquitous Knowledge Processing Lab (UKP-DIPF)
German Institute for Educational Research
http://www.dipf.de/

Abstract

We present a method for clustering word senses of a lexical-semantic resource by mapping them to those of another sense inventory. This is a promising way of reducing polysemy in sense inventories and consequently improving word sense disambiguation performance. In contrast to previous approaches, we use Dijkstra-WSA, a parameterizable alignment algorithm which is largely resource- and language-agnostic. To demonstrate this, we apply our technique to GermaNet, the German equivalent to WordNet. The GermaNet sense clusterings we induce through alignments to various collaboratively constructed resources achieve a significant boost in accuracy, even though our method is far less complex and less dependent on language-specific knowledge than past approaches.

1 Introduction

Lexical-semantic resources (LSRs) are a prerequisite for many key natural language processing tasks. However, it is nowadays widely recognized that not every resource is equally well suited for each task. For word sense disambiguation (WSD), which is the focus in this paper, the Princeton WordNet (Fellbaum, 1998) is the predominant sense inventory for English because of its free availability, its comprehensiveness, and its use in dozens of previous studies and data sets. For German, GermaNet (Hamp and Feldweg, 1997) is the German equivalent to WordNet and has positioned itself as the reference resource for WSD, although systematic investigation of German WSD has only recently begun (Broscheit et al., 2010; Henrich and Hinrichs, 2012).

There is much evidence to suggest that the sense distinctions of expert-built wordnets are far subtler than what is typically necessary for real-world NLP applications, and sometimes even too subtle for human annotators to consistently recognize. This point has been made specifically for WordNet (Ide and Wilks, 2006), but is just as applicable to other expert-built resources (Jorgensen, 1990). This makes improving upon experimental results difficult, while at the same time the downstream benefits of improving WSD on these LSRs are often not clearly visible.

Using a different sense inventory could solve the problems inherent to expert-built LSRs, and recently collaboratively constructed resources, such as Wiktionary and Wikipedia, have been suggested (Mihalcea, 2007). These resources are attractive because they are large, freely available in many languages, and under continuous improvement. However, they still contain considerable gaps in coverage, few large-scale sense-annotated corpora use them, and for some word categories their senses are also rather fine-grained. Much prior work has therefore focused instead on enhancing wordnets by decreasing their granularity through (semi-)automatic clustering of their senses. However, until now, the focus of attention has almost exclusively been the English WordNet. While it has been shown that such clustering significantly enhances both human interannotator agreement (Palmer et al., 2007) and automatic WSD performance (Snow et al., 2007), the previous approaches had been specifically tailored towards this resource, making the applicability to other LSRs, let alone other languages, difficult.

In this paper, we describe a new, fully automated approach to the granularity problem which taps the benefits of collaboratively constructed LSRs without the drawbacks of using them as wholesale replacements for other LSRs. Specifically, we induce a clustering of a resource's senses by first mapping them to those in the other resources, and then grouping source senses which map to the same target sense. This results in a coarse-grained sense inventory. In contrast to previous alignment-based clustering techniques, we use Dijkstra-WSA, a state-of-the-art sense alignment algorithm which is highly parameterizable as well as resource- and language-agnostic. This allows us to produce clusterings based on several different German resource alignments, for which we conduct in-depth analyses and evaluations. To demonstrate the language-independence of our approach, we produce clusters for both GermaNet and WordNet, though our algorithm is easily applicable to many resource pairs.

2 Related work

Clustering fine-grained sense distinctions into coarser units has been a perennial topic in WSD. Past approaches have included using text- and metadata-based heuristics to derive similarity scores for sense pairs in electronic dictionaries (Dolan, 1994; Chen and Chang, 1998), exploiting semantic hierarchies to group senses by proximity or ancestry (Peters et al., 1998; Buitelaar, 2000; Mihalcea and Moldovan, 2001; Tomuro, 2001; Ide, 2006), grouping senses which lexicalize identically when manually translated (Resnik and Yarowsky, 2000), using distributional similarity of senses (Agirre and Lopez de Lacalle, 2003; McCarthy, 2006), exploiting disagreements between human annotators of sense-tagged data (Chklovski and Mihalcea, 2003), heuristically mapping senses to learned semantic classes (Kohomban and Lee, 2005), and deep analysis of syntactic patterns and predicate–argument structures (Palmer et al., 2004; Palmer et al., 2007).

Comparison of these approaches is hampered by the fact that evaluations often are not provided in the papers, are applicable only for the particular LSR used in the experiment, do not provide a random baseline for reference, and/or provide only intrinsic measures such as "reduction in average polysemy" which do not directly speak to the clusterings' correctness or utility for a particular task. Though many of the above authors cite improved WSD as a motivation for the work, most of them do not actually investigate how their clusterings impact state-of-the-art disambiguation systems. The only exception is Palmer et al. (2007), who compare results of a state-of-the-art WSD system, as well as human interannotator agreement, on both fine-grained and clustered senses. To ensure that the measured improvement was not due solely to the reduced number of sense choices for each word, they also evaluate a random clustering of the same granularity.

Apart from the above-noted approaches, there has also been interest recently in techniques which reduce WordNet's sense granularity by aligning it to another, more coarse-grained resource at the level of word senses. Navigli (2006) induces a sense mapping between WordNet and the *Oxford Dictionary of English* (Soanes and Stevenson, 2003) on the basis of lexical overlaps and semantic relationships between pairs of sense glosses. WordNet senses which align to the same *Oxford* sense are clustered together. The evaluation is similar to that later used by Palmer et al. (2007), except that rather than actually running a WSD algorithm, Navigli expediently takes the raw results of a Senseval WSD competition (Snyder and Palmer, 2004) and does a coarse-grained rescoring of them. The improvement in accuracy is reported relative to that of a random clustering, though unlike in Palmer et al. (2007) there is no indication that the granularity of the random clusters was controlled. It is therefore hard to say whether the clustering really had any benefit.

Snow et al. (2007) and Bhagwani et al. (2013) extend Navigli's approach by training machine learning classifiers to decide whether two senses should be merged. They make use of a variety of features derived from WordNet as well as external sources, such as the aforementioned *Oxford*–WordNet mapping. They also improve upon Navigli's evaluation technique in two important ways: first, they ensure their baseline random clustering has the same granularity as their induced clustering, and second, the random clustering performance is computed precisely rather than estimated stochastically. While their methods result in an improvement over their baseline, they do require a fair amount of annotated training data, and their features are largely tailored towards WordNet-specific information types. This makes the methods' transferability to resources lacking this information rather difficult.

In this paper, we go beyond this previous work in two ways. First, we employ Dijkstra-WSA (Matuschek and Gurevych, 2013), a state-of-the-art alignment algorithm with the attractive property of being largely resource- and even language-agnostic. This makes the alignment (and hence, the clustering approach) easily applicable to many different resource combinations, though we expect its performance to be competitive with far more complex and resource-specific approaches.

Second, thanks to the flexibility of Dijkstra-WSA, we can perform a deeper comparative analysis of alignment-based clusterings against not one but three different LSRs. We investigate how the different properties of these resources influence the alignments and clusterings, particularly with respect to accuracy across parts of speech. This is the first time such a detailed analysis is presented. We focus on collaboratively constructed LSRs, as their emergence has led to an ongoing discussion about their quality and usefulness (Zesch et al., 2007; Meyer and Gurevych, 2012; Krizhanovsky, 2012; Gurevych and Kim, 2012; Hovy et al., 2013). Our work aims to contribute to this discussion by investigating the crucial aspects of granularity and coverage.

3 Alignment-based clustering

3.1 Task description

Word sense clustering is the process, be it manual or automatic, of identifying senses in an LSR which are similar to the extent that they could be considered the same, slight variants of each other, or perhaps subsenses of the same broader sense. Its purpose is to merge these senses (i.e., to consider the set of clustered senses as a single new sense) so as to facilitate usage of the sense inventory in applications which benefit from a lower degree of polysemy, such as machine translation, where lexical ambiguity is often preserved across certain language pairs, making fine-grained disambiguation superfluous. For example, the two WordNet senses of *ruin*—"destroy completely; damage irreparably" and "reduce to ruins"—are very closely related and could be used interchangeably in many contexts.

One way to achieve such a clustering is *word sense alignment* (WSA), or *alignment* for short. An alignment is formally defined as a list of pairs of senses from two LSRs, where the members of each pair represent the same meaning. When it is not restricted to 1:1 alignments, it is possible that a sense s in one LSR A is assigned to several senses $t_1, \ldots, t_n$ in another LSR B. Assuming that all alignments are correct, this implies that $s \in A$ is more coarse-grained and subsumes the other senses, which in turn can be considered as a sense cluster within B. For example, the aforementioned senses of *ruin* could both be aligned to the Wiktionary sense "to destroy or make something no longer usable" and thereby clustered.

3.2 Lexical-semantic resources

For our experiments we align GermaNet, a German wordnet, to three different collaboratively constructed German LSRs: Wikipedia, Wiktionary, and OmegaWiki. Our goal is to demonstrate that effective sense clustering is possible for resources in languages other than English using a language-agnostic alignment approach.

Moreover, we aim to cover two popular dictionary resources which are at different stages of development regarding size and coverage (OmegaWiki and Wiktionary) as well as the most popular collaboratively constructed encyclopedia (Wikipedia), which was not designed as a lexicographic knowledge source but is widely used in NLP nonetheless (Zesch et al., 2007; Milne and Witten, 2008). As the detailed results of the alignment are of secondary interest here (being exhaustively discussed in Matuschek and Gurevych (2013)), we focus on a discussion of the clusterings which are derived from the alignment and relate these results to the properties of the LSRs involved. For convenient usage in our clustering framework, we use the LSR versions found in the unified resource UBY (Gurevych et al., 2012).

GermaNet (Hamp and Feldweg, 1997) is an expert-built computational lexicon for German and thus the counterpart to WordNet. It is organized into synsets (over 84 500 in version 8.0, which we use) connected via semantic relations.

Wikipedia is a free, multilingual, collaboratively written online encyclopedia and one of the largest publicly available knowledge sources. Each article usually describes a distinct concept which is connected to other articles by means of hyperlinks. UBY contains a snapshot of the German edition from 16 August 2009 with around 834 000 articles.

Wiktionary is a dictionary "sister project" of Wikipedia. For each word, multiple senses can be encoded, and these are usually also represented by glosses. There are also hyperlinks which lead to synonyms, hypernyms, meronyms, etc. UBY's 6 April 2011 snapshot of the German edition contains around 72 000 entries.

OmegaWiki is another freely editable online dictionary. Unlike in Wikitionary, there are no distinct language editions; OmegaWiki is comprised of language-independent concepts ("defined meanings") which bear lexicalizations in various languages. These are connected by semantic relations as in WordNet. UBY uses a database dump from 3 January 2010, which contains slightly less than 47 000 concepts and lexicalizations in over 470 languages.

3.3 Dijkstra-WSA

Dijkstra-WSA is the graph-based word sense alignment algorithm which we use to infer the clusterings. It consists of three steps: (i) the initial construction of the graphs, (ii) the identification of valid alignments using a shortest path algorithm, and (iii) an optional similarity-based backoff for senses which could not be aligned.

Graph construction. The set of senses (or synsets, if applicable) of an LSR is represented as a set of nodes V where the set of edges $E \subseteq V \times V$ between these nodes represents semantic relatedness between them. This is called a *resource graph*. For deriving the edges, one can use semantic relations (such as hyponymy), hyperlinks (for Wikipedia), or other relatedness indicators provided by the resource. For sparse LSRs such as Wiktionary, it is a viable option to increase the density by adding edges between senses s_1 and s_2 if a monosemous term t with sense s_2 is included in the gloss of s_1. For example, one can link a sense of *Java* to *programming language* if the latter term is included in the former's definition text. This so-called *linking of monosemous lexemes* proved to significantly enhance the graph density (and hence, the recall of the alignment) with only a minor loss in precision.

Computing sense alignments. For the two resource graphs A and B, edges representing trivial alignments are introduced first. Alignments are trivial if two senses have the same attached lexeme in A and B and this lexeme is also monosemous in each resource. For example, if the noun phrase *programming language* is contained in either resource and has exactly one sense in each one, we can directly infer the alignment.

Next, we consider each still unaligned sense $s \in A$. We first retrieve the set of target senses $T \subset B$ with matching lemma and part of speech (e.g., *Java (island)* and *Java (programming language)*) and compute the shortest path to each of them with Dijkstra's shortest path algorithm (Dijkstra, 1959). The candidates in T with a distance below a certain threshold (estimated on a development set considering the graph size and density) are selected as alignment targets, and the algorithm continues until either all senses are aligned or no path can be found for the remaining senses. The intuition behind this is that the trivial alignments serve as "bridges" between A and B, such that a path starting from a sense s_1 in A traverses edges to find a nearby already aligned sense s_2, "jumps" to B using a cross-resource edge leading to t_2 and then ideally finds an appropriate target sense t_1 in the vicinity of t_2. In this example, the bridge *programming language* would enable the correct identification of two equivalent senses of *Java*. Note that our definition allows computation of one-to-many alignments, which are a prerequisite for the subsequent clustering step we describe in Section 3.1. Also note that with each successful alignment, edges are added to the graph so that a different ordering of the considered senses leads to different results; these differences were in no case statistically significant, however.

Similarity-based backoff. Alignments found by Dijkstra-WSA are complementary to those usually found by text similarity–based approaches. We therefore use a hybrid approach which first uses Dijkstra-WSA and falls back to gloss similarity for those cases where no target could be found in the graph. This significantly increases the alignment recall, so in order to better understand the consequences for our clustering system, we run Dijkstra-WSA both with and without this backoff. However, we do not employ a machine learning component; to keep the approach as knowledge-poor as possible, we follow the approach by Henrich et al. (2011) and align to the candidate with the greatest similarity.

4 Evaluation

4.1 Methodology

A common extrinsic method for evaluating sense clusterings is to take the raw assignments made by existing word sense disambiguation systems on a standard data set and then rescore them according to the clustering. That is, a system is considered to have correctly disambiguated a term not only if it chose a correct sense specified by the data set's answer key, but also if it chose any other sense in the same cluster as a correct one. Of course, any clustering whatsoever is likely to increase accuracy, simply by virtue of there being fewer answers for the system to choose among. To account for this, accuracy obtained with each clustering must be measured relative to that of a random clustering of equivalent granularity.[1]

The random clustering score for each instance in the data set can be determined mathematically. Snow et al. (2007) and Bhagwani et al. (2013) use

$$\sum_{c \in C} \frac{|c|\,(|c|-1)}{N\,(N-1)}, \tag{1}$$

where C is the set of clusters over the N senses of a given term, and $|c|$ is the number of senses in the cluster c. However, this formula is accurate only when the gold standard specifies a single correct answer for the instance. In practice, WSD data sets can specify multiple possible correct senses for an instance, and a system is considered to have correctly disambiguated the target if it selected any one of these senses. The Senseval-3 all-words corpus used by Snow et al. (2007) and Bhagwani et al. (2013) is such a data set (some 3.3% of the instances have two or more "correct" senses) so the scores they report underestimate the accuracy of the random baseline and inflate their clustering methods' reported improvement.

To arrive at a formula which works in the general case, consider that for an instance where the target word has N senses, g of which are correct in the given context, and one of which is an incorrectly chosen sense, the total number of ways of distributing these senses among the clusters is

$$N \cdot \binom{N-1}{g} = \frac{N!}{g!(N-g-1)!}. \tag{2}$$

Of these, the number of distributions which cluster the incorrectly chosen sense together with none of the correct senses is

$$\sum_{c \in C} |c| \binom{N-|c|}{g} = \sum_{c \in C} \frac{|c|\,(N-|c|)!}{g!\,(N-|c|-g)!}, \tag{3}$$

where the summation includes only those clusters where $N - |c| \geq g$. The probability that the incorrectly chosen sense is clustered together with at least one correct sense is therefore

$$1 - \sum_{c \in C} \frac{|c|\,(N-|c|)!\,(N-g-1)!}{N!\,(N-|c|-g)!} \tag{4}$$

or, recast for ease of programmatic computation,

$$1 - \sum_{c \in C} \frac{|c| \prod_{i=0}^{g-1} (N-|c|-i)}{\prod_{i=0}^{g} (N-i)}. \tag{5}$$

For the case where there really is only one correct gold-standard answer, Formula 4 becomes

$$\begin{aligned} 1 - \sum_{c \in C} \frac{|c|\,(N-|c|)}{N\,(N-1)} &= \sum_{c \in C} \frac{|c|}{N} - \sum_{c \in C} \frac{|c|\,(N-|c|)}{N\,(N-1)} \\ &= \sum_{c \in C} \frac{|c|\,(|c|-1)}{N\,(N-1)}, \end{aligned} \tag{6}$$

which agrees with Formula 1 above.

To compute the clustered scoring, including that of the random clusterings, we use the free DKPro WSD framework (Miller et al., 2013).

[1] Controlling for granularity is vital, since it is trivial to construct clusterings which effect arbitrarily high WSD accuracy. Consider the extreme case where for each word, *all* the senses are clustered together; this clustering would have 100% WSD accuracy and thus easily beat an uncontrolled random baseline, but not a granularity-controlled one.

	aff.	**imp.**	**%**
OmegaWiki (DWSA)	438	130	29.7
OmegaWiki (sim. only)	712	165	23.2
OmegaWiki (w/backoff)	872	205	23.5
Wiktionary (DWSA)	1355	311	23.0
Wiktionary (sim. only)	1463	349	23.8
Wiktionary (w/backoff)	1797	349	19.4
Wikipedia (DWSA)	773	120	15.5
Wikipedia (sim. only)	710	158	22.2
Wikipedia (w/backoff)	852	147	17.3

Table 1: Number and percentage of lexical items in the data set affected and improved by the clusterings. The slight proportional decrease in improved items in some configurations results from an improved alignment recall using the backoff.

4.2 Data sets and algorithms

To our knowledge, there are currently only two German-language sense-annotated corpora, both of the "lexical sample" variety: DeWSD (Broscheit et al., 2010) and WebCAGe (Henrich et al., 2012). At the time of writing only the latter was available to us, and so is the one used in our study. With 10 429 instances of 2719 lexical items annotated with GermaNet 8.0 senses, WebCAGe 2.0 is significantly larger and more up to date than DeWSD, which has 1154 instances of 40 lexical items annotated with GermaNet 5.1 senses. As with the Senseval-3 data set, many WebCAGe instances specify multiple gold-standard senses.

German-language WSD is still in its infancy; the only results reported so far on WebCAGe are for various weakly supervised, Lesk-like systems (Henrich and Hinrichs, 2012).[2] For our extrinsic cluster evaluation, we therefore rescore the sense assignments made by their *lsk_Ggw+Lgw* system, the best-performing system (in terms of recall and F_1) when run on the entire WebCAGe 2.0 corpus.

4.3 Experiments on GermaNet

GermaNet–OmegaWiki. When only Dijkstra-WSA is used for clustering, the clusters are small and few in number. This results in few lexical items in the data set being affected by the clustering, and is in line with the observation made in Matuschek and Gurevych (2013) that graph-based alignments usually yield good precision at the expense of recall. So although relatively few senses are aligned and subsequently clustered, the clusters seem mostly correct, which is indicated by the significant overall improvement. The first line of Table 1 shows how many of the 10 429 instances of the evaluation data set were actually affected by this clustering configuration, and of these how many saw an increase in accuracy over the random baseline (which is an indicator of the validity of the clusters).

For adjectives (the smallest part-of-speech group in the data set) there is almost no clustering at all, as for most senses Dijkstra-WSA identified no targets, or only one target. The situation was better for nouns and verbs; while the clusters are not large (usually 2–3 senses), the high-precision clustering did improve the results. Nouns especially saw a statistically significant[3] improvement over the random clustering (1.6 percentage points). The upper third of Table 2 shows the full results for this setup. The table shows the original accuracy score without clustering (*none*), the accuracy with our clustering (*WSA*), the accuracy with random clustering of equivalent granularity (*rand.*), and the difference between the latter two (±).

When gloss similarity is used in isolation, we achieve a higher alignment recall and thus larger clusters; this way, we are able to cluster a substantial number of adjectives, leading to an increase in WSD performance. However, the overall results are worse due to the lower precision for nouns.

When we employ the backoff to improve the recall of the graph-based alignment (i.e., a combination of both approaches), we get more and larger clusters (see third line of Table 1), leading to a significant improvement in WSD accuracy for nouns and verbs (Table 2). Although alignment precision for this setup was reported to be generally worse than for Dijkstra-WSA alone, the alignments are seemingly still precise enough to form meaningful clusters with only a few errors.

A good example is the verb *markieren* ("to mark"), whose only sense in OmegaWiki ("somehow tag for later reference") is aligned to two

[2] Broscheit et al. (2010) evaluate a graph-based WSD system, albeit only on the DeWSD corpus.

[3] All significance statements in this paper are based on McNemar's test at a confidence level of 5%.

			OmegaWiki			**Wiktionary**			**Wikipedia**		
		none	**rand.**	**WSA**	±	**rand.**	**WSA**	±	**rand.**	**WSA**	±
no backoff	noun	51.1	60.9	62.5	1.6*	75.1	77.2	**2.1***	75.1	76.3	1.2*
	verb	43.1	45.8	46.6	0.8*	60.1	61.8	**1.7***	—	—	—
	adj.	43.3	45.0	45.0	0.0	82.5	83.0	0.5	—	—	—
	all	48.1	55.3	56.5	1.2*	71.2	73.0	**1.8***	—	—	—
sim. only	noun	51.1	61.6	62.7	1.1*	72.3	73.8	1.4*	70.5	71.6	1.1*
	verb	43.1	55.5	56.3	0.8*	58.7	58.7	0.0	—	—	—
	adj.	43.3	61.6	62.1	0.5	65.9	66.3	0.4	—	—	—
	all	48.1	59.8	60.7	0.9*	67.8	68.7	0.9*	—	—	—
w/backoff	noun	51.1	66.9	68.5	1.6*	83.2	85.3	**2.1***	76.6	78.6	2.0*
	verb	43.1	56.0	57.3	1.3*	73.7	74.3	0.6	—	—	—
	adj.	43.3	61.1	62.0	**0.9**	87.9	87.8	−0.1	—	—	—
	all	48.1	63.3	64.7	1.4*	80.7	82.2	1.5*	—	—	—

Table 2: WSD accuracy (F-score) by POS, using clusterings derived from alignments of GermaNet to various resources, via Dijkstra-WSA without (top) and with (bottom) the similarity-based backoff, or via gloss similarity only (middle). Boldface marks best results per POS; asterisks mark statistically significant differences from the granularity-controlled random baseline.

GermaNet senses, one each for text and territorial marking. The difference in polysemy between GermaNet and OmegaWiki (see Table 3) pays off here, as the coarse OmegaWiki sense subsumes the GermaNet senses. This is exactly the intended effect when this kind of clustering is performed.

However, there are also many notable gaps in coverage (Table 3)—even some commonly used terms are missing from OmegaWiki altogether, leaving their GermaNet senses unaligned and unclustered. This underrepresentation of lemmas and senses can be attributed to the fact that OmegaWiki, in comparison to Wiktionary and Wikipedia, is in an earlier stage of development; this is especially true for the German edition.

GermaNet–Wiktionary. Unlike OmegaWiki, Wiktionary's coverage of lexical items is almost the same as GermaNet's (> 99%; see Table 3), which leads to a higher number of affected items in the test data set and, consequently, significantly better overall results in comparison to OmegaWiki in the same setup. For nouns and verbs, the clustering yields major improvements (Table 2), while the benefit for adjectives is modest. However, it comes as a surprise that the results are not even better—if for almost every lexeme alignment targets can be found, the assumption is that many clusters could be formed. This is not the case as on the test data set, the degree of polysemy is almost the same in both resources, and GermaNet is substantially less polysemous for verbs. Hence, for many senses in GermaNet there exists an equivalent sense with comparable granularity in Wiktionary, and no 1:*n* mapping can be found which would imply a clustering.

	GN	**OW**	**WKT**	**WP**
Nouns cov. (%)	100.0	20.6	99.9	80.6
Verbs cov. (%)	100.0	20.7	99.9	—
Adjs. cov. (%)	100.0	29.8	98.6	—
Items cov. (%)	100.0	21.4	99.8	45.6
Senses / noun	2.82	1.18	3.84	2.25
Senses / verb	3.70	1.31	3.59	—
Senses / adj.	2.48	1.26	3.24	—
Senses / item	3.21	1.23	3.69	2.25

Table 3: Coverage of lexical items in the test set per resource, and the degree of polysemy (i.e., the average number of senses per item).

While this impairs even better results for our clustering approach, it is also a strong indicator of the quality of the German Wiktionary. Its superiority in certain respects over the English version has already been described by Meyer (2013).

When both approaches are combined, recall is again considerably higher, but the overall results are not—more items are affected, but no more can be improved (see Table 1). Here, we apparently hit the limits of the clustering approach: While large clusters (and many affected items) are generally desirable, a certain level of precision has to be maintained for this approach to be effective.

GermaNet–Wikipedia. As Wikipedia contains almost exclusively noun concepts, our evaluation for this clustering was restricted to this part of speech (see Table 2). We observe that the results for Dijkstra-WSA alone as well as for the similarity-based approach are significantly better than random, but worse than for the other clusterings. This is explicable by the fact that the polysemy for nouns is comparable for GermaNet and Wikipedia (see Table 3). The observation made for Wiktionary that similar granularity implies many 1:1 alignments and thus few and small clusters holds here as well, as many GermaNet noun senses in the data set have a corresponding entry in Wikipedia. An example is the noun *Filter*, where GermaNet encodes three senses (filter for liquids, air filter, and polarization filter) which are all present in Wikipedia and correctly aligned. Due to its encyclopedic focus, Wikipedia also contains senses which are rather obscure and unlikely to be found in a dictionary (e.g., *Filter* is also an American rock band). Our analysis shows, however, that the alignment algorithm reliably rules them out as alignment targets so that they usually do not impair the clustering outcome.

When combining both approaches in the hybrid setup, we get the expected boost in recall, and the significantly better WSD result (+2.0 as compared to the random setup) suggests that the precision is still acceptable. This is in line with the results reported in (Matuschek and Gurevych, 2013) on the task of WordNet–Wikipedia alignment, which is comparable due to the similar structures of WordNet and GermaNet; in this setup, the hybrid approach yielded better recall while maintaining the same precision as the individual approaches.

Combined approaches. Our experiments show that clustering GermaNet against different collaboratively constructed LSRs using a state-of-the-art WSA algorithm is indeed effective: with few exceptions, the WSD results beat comparable random clusterings, and often significantly so.

A main insight was that different clusterings do not work equally well on each part of speech: while OmegaWiki works best for adjectives, Wiktionary gives the best results for nouns and verbs. Thus, we performed an additional experiment where optimal clusterings were chosen for each part of speech (the boldface results from Table 2). This clustering yields a significant improvement in WSD for each part of speech except adjectives, and achieves the strongest overall improvement (1.9 percentage points) over random clustering. This shows that our language-independent approach is effective, even though it consists solely of an alignment algorithm which does not rely on any resource-specific tuning or knowledge external to any of the resources involved. This is in strong contrast to previous work such as Snow et al. (2007), who employ further external resources, as well as features specifically tailored towards WordNet in a supervised machine learning setup.

4.4 Experiments on WordNet

To demonstrate the validity of our approach for English, we also clustered WordNet by aligning it to the English editions of the three collaboratively constructed LSRs and used the resulting coarse-grained WordNet for WSD. We rescored the raw sense assignments of the three top-performing systems in the Senseval-3 English all-words WSD task (Snyder and Palmer, 2004); the results, averaged across all systems, are shown in Table 4. In general, our observation of significantly improved WSD performance held for English as well. While there are some deviations from the results we reported for German, the observations regarding the properties of the collaboratively constructed LSRs can for the most part be transferred.

As for German, we observed that different clusterings do not work equally well on each part of speech. Thus, we also tested a configuration for English where we selected the optimal clusterings for each part of speech (the boldface results from Table 4). As with German, this clustering results in a significant improvement for each part of speech (except adverbs, though these comprise only 15 of the 2041 instances in the data set).

			OmegaWiki			Wiktionary			Wikipedia		
		none	**rand.**	**WSA**	±	**rand.**	**WSA**	±	**rand.**	**WSA**	±
no backoff	noun	69.0	70.2	71.0	0.8*	70.7	71.4	0.6	71.5	72.5	1.0*
	verb	56.4	59.5	61.2	**1.8***	63.8	64.9	1.1	—	—	—
	adj.	69.3	69.8	69.7	0.0	70.5	70.9	0.5	—	—	—
	adv.	86.7	86.7	86.7	0.0	86.7	86.7	0.0	—	—	—
	all	64.6	66.4	67.4	**1.0***	68.3	69.1	0.8*	—	—	—
w/backoff	noun	69.0	78.4	80.5	**2.2***	72.6	73.6	1.0*	73.5	74.2	0.8
	verb	56.4	69.5	66.9	−2.6*	65.4	66.5	1.0	—	—	—
	adj.	69.3	78.9	82.4	**3.4***	73.6	74.0	0.4	—	—	—
	adv.	86.7	86.7	86.7	0.0	86.7	86.7	0.0	—	—	—
	all	64.6	75.3	76.0	0.7	70.3	71.2	0.9*	—	—	—

Table 4: WSD accuracy (F-score) by POS, using clusterings derived from Dijkstra-WSA alignments of WordNet to various resources, without (top) and with (bottom) the similarity-based backoff. Boldface marks best results per POS; asterisks mark statistically significant differences from the random baseline.

5 Conclusions and future work

In this work, we presented a method for clustering fine-grained GermaNet senses by aligning them to three different collaboratively constructed sense inventories. We used Dijkstra-WSA, a language-independent alignment algorithm which is easily applicable to a variety of LSRs. We showed that a significant improvement in word sense disambiguation accuracy is possible with this method. In contrast to previous approaches, ours is substantially more flexible and generic, relying on no knowledge external to the LSRs and no resource-specific feature engineering. As evidence of this, we demonstrated that our method also performs well with the English WordNet. We also discussed the properties of the different LSRs regarding coverage and granularity, and showed that combining clusterings of different resources for different parts of speech leads to the best performance. Our clusterings will be made freely available to the research community at `https://www.ukp.tu-darmstadt.de/data/`.

One task we intend to investigate in future work is an evaluation on the forthcoming sense-annotated extension to the TüBa-D/Z corpus (Henrich et al., 2013). And as Dijkstra-WSA is applicable to arbitrary pairs of LSRs, we would also like to investigate clustering LSRs other than GermaNet and WordNet, which are by far not the only ones with a tendency towards microdistinction of senses (Jorgensen, 1990). Not only might this improve performance when these sense inventories are used for WSD, but it might also help in the curation of these resources by identifying questionable sense distinctions. This seems especially interesting for Wiktionary and OmegaWiki, which have quite different sense granularities but whose collaborative construction model allow for easy revision of entries.

Regarding improvements to the clustering approach itself, we would like to evaluate to what extent the clusters we create respect the existing taxonomic structure of the resources induced by semantic relations; for instance, merging senses on different levels of the GermaNet taxonomy could lead to circular or otherwise contradictory relations. Following Snow et al. (2007), we want to investigate how such violations of the taxonomy can be avoided in the algorithmic approach.

Acknowledgments

This work has been supported by the Volkswagen Foundation as part of the Lichtenberg Professorship Program under grant № I/82806, and by the Hessian research excellence program *Landes-Offensive zur Entwicklung Wissenschaftlich-ökonomischer Exzellenz (LOEWE)* as part of the research center “Digital Humanities”. The authors thank Verena Henrich for providing raw sense assignments for her systems on WebCAGe 2.0.

References

Eneko Agirre and Oier Lopez de Lacalle. 2003. Clustering WordNet word senses. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, and Nikolai Nikolov, editors, *Proceedings of the 4th International Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, pages 11–18, September.

Sumit Bhagwani, Shrutiranjan Satapathy, and Harish Karnick. 2013. Merging word senses. In *Proceedings of Workshop on Graph-based Methods for Natural Language Processing (TextGraphs-8)*, pages 11–19.

Samuel Broscheit, Anette Frank, Dominic Jehle, Simone Paolo Ponzetto, Danny Rehl, Anja Summa, Klaus Suttner, and Saskia Vola. 2010. Rapid bootstrapping of word sense disambiguation resources for German. In Manfred Pinkal, Ines Rehbein, Sabine Schulte im Walde, and Angelika Storrer, editors, *Proceedings of the 10th Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2010)*, pages 19–27, September.

Paul Buitelaar. 2000. Reducing lexical semantic complexity with systematic polysemous classes and underspecification. In Amit Bagga, James Pustejovsky, and Wlodek Zadrozny, editors, *Proceedings of the NAACL-ANLP 2000 Workshop on Syntactic and Semantic Complexity in Natural Language Processing Systems*, pages 14–19, April.

Jen Nan Chen and Jason S. Chang. 1998. Topical clustering of MRD senses based on information retrieval techniques. *Computational Linguistics*, 24(1):61–95.

Timothy Chklovski and Rada Mihalcea. 2003. Exploiting agreement and disagreement of human annotators for word sense disambiguation. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, Nicolas Nicolov, and Nikolai Nikolov, editors, *Proceedings of the 4th International Conference on Recent Advances in Natural Language Processing (RANLP 2003)*, September.

Edsger W. Dijkstra. 1959. A note on two problems in connexion with graphs. *Numerische Mathematik*, 1:269–271.

William B. Dolan. 1994. Word sense ambiguation: Clustering related senses. In *Proceedings of the 15th International Conference on Computational Linguistics (COLING 1994)*, volume 2, pages 712–716, August.

Christiane Fellbaum, editor. 1998. *WordNet: An Electronic Lexical Database*. MIT Press, Cambridge, MA, USA.

Iryna Gurevych and Jungi Kim, editors. 2012. *The People's Web Meets NLP: Collaboratively Constructed Language Resources*. Theory and Applications of Natural Language Processing. Springer.

Iryna Gurevych, Judith Eckle-Kohler, Silvana Hartmann, Michael Matuschek, Christian M. Meyer, and Christian Wirth. 2012. UBY – A large-scale unified lexical-semantic resource based on LMF. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 580–590.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet – A lexical-semantic net for German. In *Proceedings of the ACL Workshop on Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications*, pages 9–15.

Verena Henrich and Erhard Hinrichs. 2012. A comparative evaluation of word sense disambiguation algorithms for German. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC 2012)*, May.

Verena Henrich, Erhard Hinrichs, and Tatiana Vodolazova. 2011. Semi-automatic extension of GermaNet with sense definitions from Wiktionary. In *Proceedings of the 5th Language and Technology Conference (LTC 2011)*, pages 126–130.

Verena Henrich, Erhard Hinrichs, and Tatiana Vodolazova. 2012. WebCAGe – A Web-harvested corpus annotated with GermaNet senses. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics (EACL 2012)*, pages 387–396, April.

Verena Henrich, Erhard Hinrichs, and Reinhild Barkey. 2013. Extending the TüBa-D/Z treebank with GermaNet sense annotation. In Iryna Gurevych, Chris Biemann, and Torsten Zesch, editors, *Proceedings of the 25th Conference of the German Society for Computational Linguistics (GSCL 2013)*, volume 8105 of *Lecture Notes in Artificial Intelligence*, pages 89–96. Springer, September.

Eduard Hovy, Roberto Navigli, and Simone Paolo Ponzetto. 2013. Collaboratively built semi-structured content and artificial intelligence: The story so far. *Artificial Intelligence*, 194:2–27, January. (Introduction to the special issue "Artificial Intelligence, Wikipedia and Semi-Structured Resources").

Nancy Ide and Yorick Wilks. 2006. Making sense about sense. In Eneko Agirre and Philip Edmonds, editors, *Word Sense Disambiguation: Algorithms and Applications*, volume 33 of *Text, Speech, and Language Technology*, chapter 3. Springer.

Nancy Ide. 2006. Making senses: Bootstrapping sense-tagged lists of semantically-related words. In Alexander Gelbukh, editor, *Proceedings of the 7th International Conference on Intelligent Text Processing and Computational Linguistics (CICLing*

A Hybrid Entity-Mention Pronoun Resolution Model for German Using Markov Logic Networks

Don Tuggener
Unversity of Zurich
Institute of Computational Linguistics
tuggener@cl.uzh.ch

Manfred Klenner
Unversity of Zurich
Institute of Computational Linguistics
klenner@cl.uzh.ch

Abstract

This paper presents a hybrid pronoun resolution system for German. It uses a simple rule-driven entity-mention formalism to incrementally process discourse entities. Antecedent selection is performed based on Markov Logic Networks (MLNs). The hybrid architecture yields a cheap problem formulation in the MLNs w.r.t. inference complexity but pertains their expressiveness. We compare the system to a rule-driven baseline and an extension which uses a memory-based learner. We find that the MLN hybrid outperforms its competitors by large margins.

1 Introduction

Coreference resolution is an important tasks in many natural language processing pipelines. Several approaches have investigated the use of Markov Logic Networks (MLNs) for this task for the English language. Fewer approaches have explored MLNs for pronoun resolution, and, to our knowledge, none have explored the use of MLNs for German pronoun resolution.

We propose an architecture for the incorporation of MLNs in an entity-mention model for pronoun resolution in German. The hybrid architecture features two main benefits.

(i) The rule-driven, incremental entity-mention model provides a means to address the number of antecedent candidates, which is generally large in German due to morphological and semantic underspecification of certain key pronouns[1].

(ii) MLNs have attracted the attention of the coreference community, as global hard constraints can be used to enforce the transitivity and exclusiveness properties of coreference. Enforcing these properties poses problems in the classical mention-pair model (Soon et al., 2001, inter alia), where found pairs of coreferring NPs need to be merged to produce the coreference partition. The entity-mention model alleviates the need to express transitivity and exclusiveness in the MLNs, as the coreference partition is incrementally established during left-to-right processing and naturally adheres to these constraints. This allows us to model each pronoun occurrence as separate instance in the MLNs. Compared to other systems using MLNs, which model full documents, the hybrid architecture reduces the problem complexity for the MLN and, thereby, processing times.

We first review the incremental entity-mention model as implemented in the CorZu coreference system (Klenner and Tuggener, 2011). Next, we introduce the hybrid architecture which incorporates MLNs for antecedent selection. In the experiments section, we improve the CorZu system for pronoun resolution and establish a machine learning baseline based on TiMBL. Finally, we compare the three systems in the evaluation section[2].

[1] The pronoun *er (he)* can refer to both animate and inanimate entities, *sie (she/they)* is also underspecified in number; the possessive pronoun *sein* has ambiguous gender (masculine or neutral; *his/its*); the possessive pronoun *ihr* can be feminine, singular *(her)*, or plural *(their)*. Therefore, morphology cannot always be applied in a straight-forward way as a filter criterion for licensing antecedent candidates, which leads to large numbers of candidates.

[2] This work is licensed under a Creative Commons At-

2 Incremental discourse processing with an entity-mention model

To our knowledge, the CorZu system (Klenner and Tuggener, 2011) is the only ready-to-use system for coreference resolution for German[3]. The system implements a rule-driven entity-mention model, in which potential anaphors are compared to already established coreference sets and a buffer list which stores markables not yet in coreference sets. Algorithm 1 outlines the underlying discourse processing approach.

Algorithm 1 Incremental entity-mention model

```
 1: for $m \in Markables$ do
 2:     for $e \in CorefPartition$ do
 3:         if $e_{-1} < m \wedge compatible(e_{-1}, m)$ then
 4:             $Candidates \oplus e_{-1}$
 5:         end if
 6:     end for
 7:     for $np \in BufferList$ do
 8:         if $np < m \wedge compatible(np, m)$ then
 9:             $Candidates \oplus np$
10:         end if
11:     end for
12:     $ante \leftarrow get\_best(Candidates)$
13:     if $\exists ante$ then
14:         $disambiguate(m)$
15:         if $ante \in CorefPartition$ then
16:             $ante \oplus m$
17:         else
18:             $CorefPartition \oplus \{ante \oplus m\}$
19:         end if
20:     else
21:         $BufferList \oplus m$
22:     end if
23: end for
```

For every markable[4] m, preceding markables are gathered from the coreference partition (lines 2-5; only the last mention of an established coreference chain is accessible, i.e. e_{-1}) and the buffer list (7-11) as antecedent candidates and appended ($\oplus$) to the candidate list. A selection strategy then determines the best candidate to be the antecedent (line 12). m is then disambiguated and absorbs number, gender, animacy, and named entity type of the antecedent (line 14). If the antecedent is a member of a coreference chain, m is appended to that chain (lines 15-16). Otherwise, a new coreference chain is created and appended to the coreference partition (lines 17-18). If no antecedent is determined, m is appended to the buffer list (line 21).

This architecture is attractive for pronoun resolution in German, because only one candidate (the most recent candidate e_{-1}) is accessible from discourse old entities (i.e. candidates already in coreference chains). When disambiguating a resolved mention, all semantic and morphological properties of the chain are projected onto e_{-1}[5]. Therefore, other members of the chain need not be considered as candidates when resolving successive markables, which potentially reduces the number of candidates from the length of a chain to one.

The main focus of the work presented here lies on improving the antecedent selection strategy (line 12) in algorithm 1 for pronoun resolution. The CorZu system uses a rule-based antecedent selection strategy based on a ranking of grammatical functions which determines the salience of the antecedent candidates[6]. The salience of each grammatical function gf is calculated by a simple ratio:

$$salience(gf) = \frac{|mentions\ bearing\ gf|}{|mentions|}$$

As the CorZu system was designed for general end-to-end coreference resolution, and not for pronouns in particular, we will experiment with rule-based extensions to this strategy. Before doing so, we will present the MLN based replacement of the antecedent selection strategy, which forms the main contribution of this work.

3 Markov Logic Networks for Reference Resolution

Markov Logic Networks (Richardson and Domingos, 2006) combine the strength of first order

[3] http://www.cl.uzh.ch/research/coreferenceresolution.html

[4] NPs potentially partaking in coreference relations.

[5] While we investigate the pronoun resolution component of CorZu in this work, the system still produces full coreference chains using string matching methods to link nominal mentions. We retain this mechanism to disambiguate potentially underspecified antecedent candidates.

[6] If two candidates have the same salience, the more recent one is selected.

predicate logic and stochastic inference. First order predicate formulas no longer need be binary, but can be assigned a weight based on statistical analysis of training data. MLNs are an interesting framework for coreference resolution, as most systems combine some notion of rule-based filtering and machine learning.

3.1 Related work

Song et al. (2012) propose a supervised model for coreference resolution using MLNs and compare it to a MaxEnt system under the same conditions. Their MLN system outperforms its MaxEnt variant and beats all other machine learning-based systems of the CoNLL 2011 shared task. Poon and Domingos (2008) investigate unsupervised coreference resolution with MLNs on the MUC-6 and ACE corpora and outperform the best results reported so far. Hou et al. (2013) apply MLNs to the problem of bridging anaphora. As Chan and Lam (2008) have shown, MLNs also provide a suitable framework for separately modeling pronoun resolution.

A strong motivation for using MLNs in coreference resolution in related work is that MLNs can be used to easily and efficiently address the problem of pair clustering. Transitivity and exclusiveness constraints can be expressed and enforced in simple first order predicate logic formulas.

3.2 Our approach

There are three types of formulas involved in modeling MLNs: local, global, and hidden ones. In coreference resolution, the local formulas are used to express soft constraints on the relation between pairs of mentions (e.g. sentence distance) which are assigned a weight during learning. Global hard constraints express the transitivity, symmetry, and exclusiveness properties of coreference and guide the pair clustering which generates the coreference partition. Finally, hidden predicates list the coreference relations between the mentions (i.e. the relations that need to be inferred during resolution).

A benefit of the entity-mention model is that clustering is not needed, as the coreference partition is established incrementally during left-to-right text processing, and the model naturally adheres to the transitivity and exclusiveness constraints of coreference. Therefore, we only need one global hard constraint, namely that a pronoun has exactly one antecedent.

As related work models whole documents in MLNs as instances, the number of hidden predicates per instance I is given by the number of mentions and the lengths of the chains they are in. This equals the sum of the pairwise permutation of mentions n pertained in each chain $c_{i...m}$, which amounts to

$|hidden\_predicates| \in I = \sum_{c_i}^{c_m} \frac{n_i!}{(n_i-2)!}$.

In contrast, because we do not need to express transitivity and exclusiveness in the MLN, we model each occurrence of a pronoun in a document as an instance and infer it separately, which gives us

$|hidden\_predicates| \in I = 1$.

Additionally, we reduce the MLN's workload by outsourcing the check for compatibility of antecedent candidates and a pronoun. Antecedent candidates are generated by the entity-mention model which uses hard filtering of candidates based on morphological agreement and distance[7]. This reduces the number of predicates and formulas needed in the MLN and, thereby, its complexity, which leads to fast processing times.

If clustering and, therefore, global constraints are not needed in our approach, the question why MLNs are still an interesting approach for this work arises. As e.g. Huang et al. (2009) noted, an important advantage of MLNs over other machine learning frameworks such as MaxEnt, kNN, Decision Trees, etc. is that weights are learned for instantiations of formulas, rather than for individual features. Similar to Conditional Random Fields, MLNs can express relations between features and weight them. Features are instantiated as predicates and be freely combined in formulas.

Furthermore, the weighting of formulas can be conditioned on any atom instantiated in the contained predicates. For example, conditioning the sentence distance between antecedents and pronouns on the pronoun type simply involves in-

[7] Relative pronouns can only have antecedents in the same sentence. Personal and possessive pronouns are allowed to have antecedents at most three sentences away. Unless a pronoun is underspecified in its morphological features, antecedent candidates must match in their morphology.

stantiating the PoS tag of the pronoun in a predicate and adding the tag to the weighting function. Such specification needs separate classifiers with specific training sets in other machine learning frameworks. Thus, MLNs provide an interesting framework, as different aspects of available information can be combined and weighted specifically.

4 Experiments

4.1 Data and evaluation metric

We use the TübaD/Z corpus (Hinrichs et al., 2005b) in its current version 9 for our experiments. The corpus contains 3444 newspaper articles annotated with coreference. We perform a 20%-20%-60% split on the data to obtain the test, development, and training sets[8]. Note that we use the gold preprocessing annotation throughout all our experiments to prevent preprocessing noise from influencing the comparison of the different approaches, but perform automated markable extraction. That is, we do not rely on the coreference annotation to identify which NPs should be considered as antecedent candidates.

As commonly used coreference metrics (MUC, BCUB, CEAF, BLANC) are not able report PoS-specific analysis of system outputs, they are not suited for pronoun resolution evaluation. Recently, Tuggener (2014) proposed the ARCS metrics, which are geared towards evaluation of coreference system outputs for higher level applications. These metrics provide PoS-based evaluation and can, therefore, be used for pronoun evaluation. Since the metrics can measure any annotated feature in corpus data, we report performance on the different pronoun types and their different lemmas[9].

The metrics use true positives (correctly resolved mentions), false negatives (unresolved mentions), and false positives (resolved markables that are not coreferential) to calculate Recall and Precision. The metrics also introduce a novel error class, called *wrong linkage*, which denotes coreferent mentions that have been resolved to wrong antecedents. Recall is calculated by $\frac{tp}{tp+wl+fn}$, and Precision by $\frac{tp}{tp+wl+fp}$. Recall thus extends over all mentions in the annotated corpus, and Precision calculation includes all coreference relations in the system output.

We choose the *ARCS inferred antecedent* metric which requires mentions to link to correct nominal antecedents within the coreference chain they are assigned to in order to be counted as true positives. The metric is strict in the sense that it does not reward simply linking pronouns to other pronouns. Only when pronouns (transitively) link to correct nominal antecedents they are regarded as true positives. We choose this metric, because we believe that pronoun resolution should at least infer correct local nominal antecedents in order to facilitate text understanding.

4.2 Extending the rule-based system

To establish a solid rule-based baseline, we add several constraints on the antecedent candidate generation mechanics in CorZu and report their impact on the development set in table 1.

PoS specific salience (+spec.sal.): The ranking of grammatical functions is performed uniformly for personal and possessive pronouns in CorZu. For relative pronouns, the most recent antecedent candidate is selected. We recalculate the salience of grammatical functions separately for personal and possessive pronouns to obtain pronoun type-specific salience rankings of the grammatical functions.

Grammatical function projection (+sal.proj.): The salience of an antecedent candidate is defined solely by the grammatical function it bears. From the discourse old entities, only the most recent mention is accessible for subsequent reference. Therefore, the grammatical function of the most recent mention of the entity determines its salience. We found that this is problematic when possessive pronouns are the most recent mentions, as they always bear the label *DET* (determiner), which is not as salient as e.g. *SUBJECT*. Therefore, if a possessive pronoun selects an antecedent within the same sentence (and is subsequently the only accessible

[8] For reproducibility, we report document ids: test set: text_0-text_689; dev set: text_690-text_1380; train set: text_1381-text_3444.

[9] We exclude the notorious (because potentially pleonastic) neutral pronoun *es (it)* from our experiments. We found that only around 10% of them are annotated as being anaphoric in the corpus. The baseline for not resolving *es* it therefore simply too high.

	Personal Pronouns						Possessive Pronouns						Relative Pronouns		
Lemma	sie			er			sein			ihr			der — die — das		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
baseline	**43.12**	**40.50**	**41.76**	**60.33**	**58.93**	**59.62**	**61.16**	**58.22**	**59.65**	**48.58**	**45.21**	**46.84**	**79.24**	**76.60**	**77.90**
+spec.sal.	47.50	44.90	46.16	64.44	62.72	63.57	65.47	62.25	63.82	51.06	47.63	49.29	79.24	76.60	77.90
+sal.proj.	47.71	45.13	46.38	66.00	64.23	65.10	67.08	63.77	65.38	51.54	48.07	49.74	79.24	76.60	77.90
+conn.	49.86	47.17	48.48	67.00	65.21	66.10	67.20	63.89	65.50	52.25	48.73	50.43	79.24	76.60	77.90
+insent	**51.29**	**48.36**	**49.78**	**65.72**	**63.97**	**64.83**	**68.43**	**65.14**	**66.75**	**53.78**	**49.95**	**51.79**	**79.23**	**76.63**	**77.91**

Table 1: Evaluation of extensions to the CorZu system on the development set.

mention of the entity), its grammatical function is overridden by that of the antecedent. Doing so, we prevent the salience of entities from being downgraded when they are referred to by a possessive pronoun in the same sentence.

Discourse connectors (+conn.): If a pronoun is preceded by a discourse connector, such as *because* or *although*, we only consider intra-sentential antecedent candidates. The intuition behind this constraint is that discourse relations such as *elaboration* or *contradiction* tend to have their arguments not too far apart in discourse. If a pronoun is an argument of such a relation, its antecedent should be nearby.

Intra-sentential candidates (+insent): A distance window of three sentences is often chosen to look for antecedents when resolving pronouns. However, pronouns tend to bind to intra-sentential antecedents quite frequently, disregarding the salience of the candidates. Therefore, we only keep candidates from within the same sentence, if available. Additionally, if there are pronouns among the intra-sentential candidates that are of the same PoS tag as the pronoun that is to be resolved, we discard all other candidates. Favoring the intra-sentential candidates is an attempt to complement the antecedent selection in CorZu, which is solely based on grammatical functions, with the similarly important factor of distance.

The results in table 1 show that all our extensions improve performance on personal and possessive pronouns. The relative pronouns do not seem to be affected, but their baseline performance is already quite strong. Calculating specific salience rankings of the grammatical functions for personal and possessive pronouns provides the highest single increase in performance. The other additions only marginally improve performance individually, but their cumulation leads to a solid upgrade of the CorZu system.

An interesting observation is the difference in performance regarding the gender of the personal and possessive pronouns. Performance on the masculine pronouns (*er, sein*) is much stronger. This may be caused by the fact that the feminine pronoun lemmas (*sie, ihr*) are ambiguous, i.e. they subsume the plural forms of the personal and possessive pronouns. These plural forms can have conjuncted NPs as antecedents, which are harder to handle.

4.3 TiMBL variant

To establish a machine learning-based baseline for the MLN system, we re-implement the TiMBL classifier approach by Klenner and Tuggener (2011). TiMBL is a kNN framework widely used in coreference and pronoun resolution (Hinrichs et al., 2005a; Hendrickx et al., 2007; Recasens and Hovy, 2009; Wunsch, 2010, inter alia). Klenner and Tuggener (2011) used individual classifiers for the different pronoun types. To stay close to their system, we implement three classifiers for each pronoun type, i.e. personal, possessive, and relative pronouns. The authors state that they used standard feature sets, but did not list them explicitly. In order to make available the same information to the TiMBL system as we will use in the MLN, we create the following feature vector for pairing an antecedent candidate i with a pronoun j:

baseline: Sentence and markable distance between i and j; grammatical function of i. **+syntax**: Grammatical function of j; whether the grammatical functions are parallel; concatenation of the grammatical functions of i and j; PoS tag of i. **+conn.**: Whether j is governed by a discourse connector. **+old/new**: Whether i is a new or old discourse entity (i.e. if the i stems from the

	Personal Pronouns						Possessive Pronouns						Relative Pronouns		
Lemma	sie			er			sein			ihr			der — die — das		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
baseline	**46.04**	**43.27**	**44.61**	**58.04**	**56.65**	**57.34**	**60.42**	**57.31**	**58.82**	**48.82**	**45.33**	**47.01**	**76.40**	**73.99**	**75.17**
+PoS.spec.	48.30	45.37	46.79	62.52	60.85	61.68	64.86	61.59	63.18	54.32	50.44	52.31	78.98	76.35	77.64
+conn.	49.27	46.28	47.73	63.53	61.83	62.67	64.98	61.71	63.30	54.56	50.66	52.54	78.98	76.35	77.64
+recency	*45.84*	*43.01*	*44.38*	*60.15*	*58.54*	*59.33*	*64.00*	*60.77*	*62.34*	*53.37*	*49.56*	*51.40*	79.29	76.65	77.95
+old/new	48.82	46.15	47.45	66.00	64.18	65.07	67.69	64.14	65.87	53.43	49.67	51.48	79.29	76.65	77.95
+syntax	50.00	46.97	48.44	68.19	66.37	67.27	71.15	67.96	69.52	50.65	46.98	48.75	79.29	76.65	77.95
+ne_type	50.00	46.97	48.44	68.19	66.37	67.27	71.15	67.96	69.52	50.65	46.98	48.75	79.29	76.65	77.95
+anim	51.67	48.56	50.07	69.38	67.65	68.50	72.38	68.82	70.55	54.67	50.71	52.62	79.29	76.65	77.95
-recency	**52.40**	**49.22**	**50.76**	**71.66**	**69.75**	**70.69**	**72.87**	**69.53**	**71.16**	**55.62**	**51.59**	**53.53**	**79.10**	**76.52**	**77.79**

Table 2: Evaluation of the TiMBL variant on the development set.

coreference partition or the buffer list). **+recency**: Whether i is the most recent candidate. **+anim.**: Animacy[10] of i. **+ne_type**: Named entity class of i.

Results of the TiMBL extension on the development set are shown in table 2. Note that we use the CorZu base system for processing, i.e. we remove the added rules from the previous experiment. For the **baseline**, we train a single classifier for all pronoun types. Next, we train separate classifiers for each pronoun type using the baseline features **(+PoS.spec.)**. We then incrementally add the additional features outlined above. To obtain the final TiMBL-based system, we remove the recency feature, as it impoverishes performance ($-$**recency**).

Evaluation shows that the TiMBL extension outperforms its rule-based counterpart especially for the masculine pronouns, and by a small difference in the female/plural pronouns. The biggest overall improvement stems from using separate, pronoun type-specific classifiers. Additionally, a relatively large performance increase can be observed for the masculine pronouns when adding the **+syntax** features, and the feminine/plural pronouns benefit from the **+anim** feature, especially. Note that the **+ne_type** feature does not have any affect on performance. We will return to this issue in section 4.5.

4.4 MLN hybrid

Next, we replace the antecedent selection step in algorithm 1 by MLNs. Table 3 shows the predicate logic formulas we experiment with. Markables in a document are enumerated from left to right following text direction. m denotes the numeric ID of a specific antecedent candidate for a specific numeric pronoun ID p. M denotes the set of available candidates for a given pronoun p. For learning, the most recent true antecedent among the candidates (i.e. the hidden predicate) is labeled based on the gold standard annotation.

We use *thebeast*[11] (Riedel, 2008) for MLN modeling. We set *thebeast* to use Integer Linear Programs for representing ground Markov networks and couple it with the *gurobi* solver[12] and learn for five epochs.

As in the TiMBL experiment, we remove the extensions to CorZu and use its vanilla instantiation as our base. We start with the baseline which uses only the formulas for sentence distance, markable distance, and grammatical function of the antecedent and incrementally append the formulas described in table 3. To enable PoS-specific weighting (**+PoS.spec.**), the predicate has_pos is added to each formula. For example, the formula for sentence distance is extended to:
$w(s2 - s1, pos) \ : \ insentence(m, s1) \wedge insentence(p, s2) \wedge has\_pos(p, pos) \rightarrow anaphoric(p, m)$
A weight is thus learned specifically for the different instantiations of the atoms in the weight function. In the sentence distance formula, the

[10] We determine animacy of named entities by a list of first names gathered from the internet. If a named entity includes a name from this list, we label it as animate. For common nouns, we query GermaNet (Hamp and Feldweg, 1997) to assess whether the noun is a hyponym of the synset *Mensch (Human)*.

[11] https://code.google.com/p/thebeast/

[12] http://www.gurobi.com/

Hidden predicate Predicate to be inferred by the MLN: $anaphoric(p, m)$
Global hard constraint formula The pronoun must have exactly one antecedent: $\lvert \forall m \in M : anaphoric(p, m) \rvert == 1$
Local soft constraint formulas **Distance-based formulas** -Sentence distance between m and p (**baseline**): $w(s2 - s1) : insentence(m, s1) \wedge insentence(p, s2) \rightarrow anaphoric(p, m)$ -Markable distance if m and p are in the same sentence (**baseline**): $w(p - m) : insentence(m, s) \wedge insentence(p, s) \rightarrow anaphoric(p, m)$ -Closest m to p (**+recency**): $w : \lvert \forall m2 \in M : m2 > m \rvert == 0 \rightarrow anaphoric(p, m)$ -Closest m to p bearing "SUBJECT" as grammatical function (**+recency**): $w : has\_gf(m, SUBJECT) \wedge \lvert \forall m2 \in M : has\_gf(m2, SUBJECT) \wedge m2 > m \rvert == 0 \rightarrow anaphoric(p, m)$
Syntax-based formulas -Grammatical function of m (**baseline**): $w(gf) : has\_gf(m, gf) \rightarrow anaphoric(p, m)$ -Parallelism of the grammatical functions of m and p (**+syntax**): $w(gf) : has\_gf(m, gf) \wedge has\_gf(p, gf) \rightarrow anaphoric(p, m)$ -Transition of grammatical functions from m to p (**+syntax**): $w(gf1, gf2) : has\_gf(m, gf1) \wedge has\_gf(p, gf2) \rightarrow anaphoric(p, m)$
Semantic formulas -Animacy of m (**+anim.**): $w(anim, gen, pos) : has\_animacy(m, anim) \wedge has\_gender(p, gen) \wedge has\_pos(p, pos) \rightarrow anaphoric(p, m)$ -Named entity type of m (**+ne_type**): $w(ne\_type) : has\_pos(m, NE) \wedge has\_ne\_type(m, ne\_type) \rightarrow anaphoric(p, m)$
Discourse-based formulas -Selecting m based on its discourse status (i.e. discourse-new vs. discourse-old) (**+old/new**): $w(ds) : has\_discourse\_status(m, ds) \rightarrow anaphoric(p, m)$ -Sentence distance if p is preceded by a discourse connector (**+conn.**): $w(s2 - s1) : insentence(m, s1) \wedge insentence(p, s2) \wedge has\_connector(p) \rightarrow anaphoric(p, m)$

Table 3: First order predicate logic formulas for MLN-based pronoun resolution in German

first value for the weight condition is the return value of a function over two atoms (the subtraction of numeric sentence IDs) and the second a PoS tag. Note that we apply this extension to all formulas in table 3.

For **+conn.**, the formula for weighting sentence distance in the presence of a discourse connective is added. **+recency** signifies the addition of the two formulas for weighting the most recent candidate and the most recent candidate bearing the grammatical label *SUBJECT*. **+old/new** adds the formula for selecting a discourse-old vs. a discourse-new candidate. **+syntax** signifies the addition of the formulas capturing the parallelism between m and p, and the transition of grammatical functions from m to p. Parallelism of grammatical functions has been used in pronoun resolution systems dating back to (Lappin and Leass, 1994). Capturing the transitions of grammatical functions from m to p is motivated by Centering theory (Grosz et al., 1995), which formulates typical transitions of grammatical functions of re-occurring entities in coherent texts. **+ne_type** weights the named entity type of m, if it is a named entity. **+anim** adds the formula for weighting the animacy of m specifically for each pronoun type and gender combination.

The results in table 4 show that the added formulas slowly but steadily increase pronoun resolution performance. A big improvement for the masculine pronouns stems from the addition of the NE type formula. For the feminine/plural pronouns, the animacy formula constitutes the single most significant improvement. Overall, the MLN hybrid outperforms the other systems by large margins. The MLN baseline using only three formulas already outperforms the CorZu extended system. Relative pronouns are the exception. Apart from learning PoS specific weights, they are not affected by the added formulas.

	Personal Pronouns						Possessive Pronouns						Relative Pronouns		
Lemma	sie			er			sein			ihr			der — die — das		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
baseline	**52.16**	**48.99**	**50.52**	**67.18**	**65.28**	**66.22**	**72.38**	**68.74**	**70.51**	**56.45**	**52.36**	**54.33**	**73.14**	**70.79**	**71.95**
+PoS.spec.	53.13	49.90	51.47	68.74	66.84	67.78	73.37	69.67	71.47	57.40	53.24	55.24	79.59	76.95	78.25
+conn.	53.13	49.90	51.47	69.29	67.38	68.32	73.37	69.67	71.47	57.40	53.24	55.24	79.59	76.95	78.25
+recency	54.32	51.01	52.61	69.20	67.41	68.29	73.49	69.87	71.63	57.75	53.63	55.61	79.84	77.18	78.49
+old/new	55.92	52.52	54.17	70.75	68.74	69.73	73.86	70.22	72.00	59.17	54.82	56.92	79.72	77.06	78.37
+syntax	55.99	52.58	54.23	71.57	69.54	70.54	73.98	70.34	72.12	58.82	54.56	56.61	79.16	76.53	77.82
+ne_type	56.48	53.04	54.70	75.41	73.27	74.32	79.28	75.29	77.24	60.59	56.20	58.31	80.09	77.42	78.73
+anim	**59.75**	**56.12**	**57.88**	**75.23**	**73.09**	**74.14**	**79.41**	**75.41**	**77.36**	**64.62**	**60.00**	**62.22**	**79.96**	**77.30**	**78.61**
TiMBL	52.40	49.22	50.76	71.66	69.75	70.69	72.87	69.53	71.16	55.62	51.59	53.53	79.10	76.52	77.79
CorZu	51.29	48.36	49.78	65.72	63.97	64.83	68.43	65.14	66.75	53.78	49.95	51.79	79.23	76.63	77.91

Table 4: Experiments with the MLN-extended system on the development set.

4.5 Comparison on the test set

Finally, we compare the systems on our test set. Table 5 reports the results. The system ranking does not change. However, we note that all systems achieve higher scores, especially for the feminine/plural pronouns.

A reason for the better performance of the MLN system compared to the TiMBL variant lies in the way they perform learning. While TiMBL calculates Gain Ratio for each of the 13 features in each of the three classifiers, amounting to 39 weights, *thebeast* learns a weight for each instantiation of the 11 formulas, which leads to 326 weights. That is, *thebeast* is able to absorb and apply the provided information in a more specific and detailed way.

Another benefit of *thebeast* manifests in the impact of adding NE types as a feature. In *thebeast*, we can require the formula to trigger only when the antecedent candidate is actually a named entity, indicated by the predicate $has\_pos(m, NE)$. The weight learning for this formula will only be triggered if this constraint is satisfied. In TiMBL, where fixed-length feature vectors are required, we need to insert a dummy value for the NE type feature if the antecedent candidate is not a NE. This dummy value will then be accounted for during feature weighting. Our evaluation on the development set showed that NE type information leads to a strong improvement in the MLN system, while it does not affect the TiMBL variant.

For error analysis, we checked the different error types that the ARCS metric measures. We found that all the systems have roughly the same number of false negatives and false positives. The false negative and false positive counts are much lower than the true positive and wrong linkage counts. For example, the MLN hybrid has the following counts for the *sie* pronoun: tp: 742, wl: 391, fn: 31, fp: 90. Therefore, it seems that it is the difference in the counts of true positives and wrong linkages that drives the difference in performance. However, we note that all our systems have much higher false positive than false negative counts, which indicates that the systems tend to resolve too many pronouns. A manual inspection of the system outputs showed that the false positives stem from cataphoric pronouns (which our systems treat as anaphors), generic uses of pronouns (which are anaphoric but not coreferent), and annotation errors (i.e. mostly missing annotations of pronouns).

5 Comparison to Related Work

Hinrichs et al. (2005a) experimented with German pronoun resolution on the TübaD/Z corpus. They first re-implemented the approach by Lappin and Leass (1994) for German and then explored TiMBL as a machine learning framework, using features based on distance and grammatical functions. The TiMBL system outperformed the rule-based system slightly, as in our experiments.

We have used similar features, but have explored two semantics-based ones, additionally. With the exception of Kouchnir (2004), who uses the semantic classes *human, physical*, or *abstract*, we are, to our knowledge, the first to use animacy and NE types as features in pronoun resolution for German. These features proved to significantly

	Personal Pronouns						Possessive Pronouns						Relative Pronouns		
Lemma	sie			er			sein			ihr			der — die — das		
	R	P	F	R	P	F	R	P	F	R	P	F	R	P	F
TübaD/Z test set															
MLN	**63.75**	**60.67**	**62.17**	**75.87**	**74.96**	**75.41**	**77.19**	**71.87**	**74.44**	**67.99**	**62.44**	**65.10**	**81.60**	**77.43**	**79.46**
TiMBL	57.23	54.96	56.07	71.24	70.69	70.96	73.09	68.60	70.77	61.62	56.93	59.18	81.25	77.10	79.12
CorZu	56.44	53.85	55.12	67.74	67.10	67.42	69.67	65.09	67.30	60.42	55.69	57.96	81.03	76.90	78.91
SemEval test set															
MLN	**52.04**	**54.69**	**53.33**	**64.79**	**65.09**	**64.94**	**72.73**	**74.61**	**73.66**	**64.55**	**65.59**	**65.07**	**79.39**	**81.43**	**80.39**
SUCRE	35.88	45.85	40.26	42.92	49.73	46.08	52.04	62.96	56.98	53.51	61.49	57.23	72.50	74.57	73.52
BART	33.83	35.00	34.40	53.30	54.85	54.07	54.82	55.96	55.38	54.79	58.86	56.75	40.60	40.71	40.65

Table 5: Comparison of systems on the test sets.

boost performance in our experiments.

Wunsch et al. (2009) explored instance sampling to reduce the large number of (negative) instances when resolving German pronouns. They used standard features and compared TiMBL to a decision tree and a maximum entropy learner. Instead of (under)sampling, we use the incremental entity-mention model to address the problem of the large number of (negative) instances.

In contrast to the approaches above, we aimed at detailed evaluation of pronoun resolution in a setting driven towards usability for higher-level applications. Therefore, we have used the ARCS metric which requires the closest nominal antecedent chosen by our systems to be correct. Our analysis showed that performance varies strongly between pronoun types and lemmas. We found that resolution of masculine pronouns is better than that of their female/plural counterparts.

As we used a more recent version of the TübaD/Z, we could not directly compare our results to previous work. However, the SemEval 2010 shared task on coreference resolution in multiple languages (Recasens et al., 2010) featured German as a language, with data drawn from the TübaD/Z[13]. We applied the ARCS scorer to the response files of the two best performing systems for German, namely SUCRE (Kobdani and Schütze, 2010) and BART (Broscheit et al., 2010), to measure their performance on pronoun resolution. We re-trained the MLN system on the shared task training data. Since we use GermaNet and gazetteers to obtain animacy information, our system falls in the category *open/gold*, like BART, while SUCRE participated in the *closed/gold* setting. The MLN system clearly outperforms the other two (cf. Table 5), although we have to consider that these systems were designed for multilingual coreference resolution and were not tuned for pronoun resolution in German.

6 Conclusion

We have investigated the integration of MLNs into a state-of-the-art rule-based entity-mention model for German pronoun resolution. An advantage of the hybrid architecture over related work using MLNs lies in the reduction of the workload for the MLNs.

We have compared the MLN extension to a rule-based antecedent selection baseline and a TiMBL variant. The MLN system clearly outperformed its competitors in our experiments.

Additionally, we have found that there are large performance differences between different pronoun types and lemmas. Our evaluation showed that pronoun resolution still leaves room for substantial improvements when we require nominal antecedents to be produced. To our knowledge, we are the first to report detailed 3rd person pronoun resolution results on the TübaD/Z 9.

Acknowledgements

We would like to thank Angela Fahrni for her inspiration for investigating MLNs and *thebeast* and for her help in doing so. Additionally, we are grateful for the excellent and helpful comments by the anonymous reviewers.

[13] The TübaD/Z version used for SemEval is significantly smaller than the current version 9. In our test set based on version 9, there are 3 to 5 times more pronouns than in the SemEval test set. We choose the newer version because it therefore is a more solid foundation for our evaluation.

Towards a syntactically motivated analysis of modifiers in German

Ines Rehbein
Universität Potsdam
German Department
SFB 632 “Information Structure”
irehbein@uni-potsdam.de

Hagen Hirschmann
Humboldt-Universität zu Berlin
Department of German Studies
and Linguistics
hirschhx@hu-berlin.de

Abstract

The Stuttgart-Tübingen Tagset (STTS) is a widely used POS annotation scheme for German which provides 54 different tags for the analysis on the part of speech level. The tagset, however, does not distinguish between adverbs and different types of particles used for expressing modality, intensity, graduation, or to mark the focus of the sentence. In the paper, we present an extension to the STTS which provides tags for a more fine-grained analysis of modification, based on a syntactic perspective on parts of speech. We argue that the new classification not only enables us to do corpus-based linguistic studies on modification, but also improves statistical parsing. We give proof of concept by training a data-driven dependency parser on data from the TiGer treebank, providing the parser a) with the original STTS tags and b) with the new tags. Results show an improved labelled accuracy for the new, syntactically motivated classification.

1 Introduction

The Stuttgart-Tübingen Tagset (STTS) (Schiller et al., 1999) is a widely used POS annotation scheme for German. It provides 54 different tags for the analysis of German, partly based on morphological and distributional properties, partly also taking semantics into account. The tagset, however, does not distinguish between adverbs and different types of particles used for expressing modality, intensity, graduation, or to mark the focus of the sentence. This is understandable, as these distinctions are often hard to make and thus might decrease the consistency of the annotations as well as make the annotation process more time-consuming.

Nonetheless, there are many tasks where one would wish for a more fine-grained analysis, especially when analysing spoken language or user-generated content from the web, but also for newspaper text where we can find a high variety of different modifiers. Consider, e.g., examples (1)-(3) below.

(1) Russland ist doch aber auch noch da.
Russia is however but also still there.
“But after all, Russia is also still there.”
[spoken language utterance]

(2) [...] , im Roman heißt sie ja ohnehin
[...] , in the novel is called she PTC anyway
zumindest fast immer nur Caro.
at least nearly always only Caro.
“[...], in the novel, she is nearly always only called Caro, anyways.” [from Twitter]

(3) [...] , jetzt vielleicht sogar noch mehr.
[...] , now maybe even still more.
“[...], but now maybe even more so.”
[newspaper text (TiGer)]

According to the STTS, the modifier sequences in (1)-(3) would be annotated as shown in (4)-(6).

(4) doch aber auch noch da
ADV ADV ADV ADV ADV

(5) ja ohnehin zumindest fast immer nur
ADV ADV ADV ADV ADV ADV

(6) jetzt vielleicht sogar noch mehr
ADV ADV ADV ADV ADV

Long sequences of adverbs and particles are particularly frequent in spoken dialogues and in conceptually spoken registers[1] but are also common in newspaper text. Thus, an analysis telling us that *ja* in (2) is a modal particle, *fast* (nearly) modifies *immer* (always) on a gradual scale, and that *nur* (only) is associated with the focus would be far more informative than the analysis given above. The question is, is such an analysis feasible with respect to annotation consistency and time, and how hard is it for automatic methods to learn these distinctions.

In the paper, we follow up on these questions and present a new classification for the analysis of modifiers in German, based on a syntactic perspective on part of speech categories (for details see Section 3).

Section 2 starts with a brief review of related work, then we describe the new tagset and motivate the linguistic basis of the distinctions between the tags (Section 3). In Section 4 we present an annotation study where we report on inter-annotator agreement and discuss the difficulties we encounter when applying the new classification to data from the TiGer treebank (Brants et al., 2002). In Section 5 we investigate the impact of the syntactically motivated annotations on the accuracy of a syntactic parser. We train a data-driven dependency parser on a subset of the TiGer treebank which we re-annotated using the new tags. Results show that the new classification improves labelled accuracy scores (LAS) especially for modifier relations. In Section 6 we discuss our results and outline future work.

2 Related Work

There is some previous work on improving natural language processing by refining POS tagsets. However, most of these studies have been conducted on English (with the exception of Kübler and Maier (2014)) and have reported negative results.

MacKinlay and Baldwin (2005) investigate the impact of different POS tagsets on automatic tagging accuracy by introducing finer distinctions between the tags. Their refined tagsets did not succeed in improving tagging accuracy. The authors attribute this to data sparseness.

Dickinson (2006) also tries to improve the results of automatic POS tagging by redefining ambiguous tags in the tagset. His approach is to add complex tags to the tagset which reflect the ambiguity of certain word forms. This approach gave slight improvements on the test set but proved to be less robust than the same tagger trained on the original tagset.

In contrast to the studies mentioned above, our main motivation for refining the STTS is not to improve tagging accuracy but to investigate whether taking a syntactically motivated perspective on POS tagset distinctions is reflected in the outcome of a syntactic parser, where (manually or automatically assigned) POS tags are crucial information to build up the syntax tree.

There is some evidence against our hypothesis. Kübler and Maier (2014) compare the influence of different POS tagsets, the German STTS, the coarse-grained universal tagset of (Petrov et al., 2012), and a fine-grained German tagset including morphological information, on constituency parsing results. They use the Berkeley parser (Petrov et al., 2006), a PCFG-LA parser, and show that in some settings, the coarse-grained universal tags are more useful to the parser than the more fine-grained STTS tags, while the morphologically enriched tags seem to be too sparse for the parser to benefit from the information. However, it is hard to draw conclusions from this, as the Berkeley parser does not take the tags as they are but, during training, refines the annotations by applying merging and splitting operations to the nodes in the tree, and only keeps those labels which have been shown to be useful during training. By just looking at the parsing results, we do not know what the internal representation used by the parser after the training cycles looked like.

We argue that a more straight-forward way to compare the influence of different POS tagset distinctions on syntactic parsing consists in using a

[1] Here we refer to the model of Koch and Oesterreicher (1985) who describe texts from written registers which display many features of spoken language as *conceptually oral*. A case in point are texts from computer-mediated communication (CMC) such as chat, facebook comments or Twitter messages.

dependency parser where the POS tags are provided as features, thus making it easier to directly compare their impact on the parsing results. In contrast to Kübler and Maier (2014), we do not compare the STTS with a general version of the tagset where all tags have been modified. Our tagset only applies linguistically motivated changes to specific tags, namely to those dealing with modification. As these are fairly frequent, we hypothesise that data sparseness will not be a big issue and that a theoretically well-funded analysis will have a positive impact on parsing results.

Relevant to our work is also the study by Plank et al. (2014), who discuss the problems of unreliable POS annotations. They show that incorporating annotator disagreements into the loss function of the POS tagger does yield better results not only on different POS tagsets but also in an extrinsic evaluation where these POS tags are used as input to a syntactic chunker.

This study is of interest to us as it gives some evidence that providing the parser with more specific information on ambiguous word forms might improve parsing. Our approach, however, is different from the one in Plank et al. (2014) who do incorporate the ambiguity in the tagging model. Instead, we aim at reducing the ambiguity in the data by refining the tagset and thus by providing the parser with more useful information.

Dalrymple (2006) follows the question how much POS tagging can help for reducing ambiguity during parsing. She presents a thorough study assessing the impact of POS tagging on parse disambiguation, applied to the output of a large-scale English LFG parser. Her findings show that presenting the parser with perfect tags would resolve ambiguitiy for around 50% of the parse trees, but that for 30% of the sentences in the test corpus even perfect POS tags would not help to disambiguate the parser output. In contrast to our work, Dalrymple does not investigate in how far modifications to the tagset might help.

3 The annotation scheme

In the standard part of speech tagset for German, the STTS, about 54 tags were defined which can be categorised into eleven major classes on a less fine-grained level (Schiller et al. 1999, pp. 4f). 48 of the tags represent word classes as such, six tags refer to punctuation marks, special characters, truncated word parts, and non-German words. The classification is based on very heterogeneous criteria – some definitions refer to the word's inflectional status (as for subclasses of verbs there are distinct categories for finite and infinite verb forms, past participles, and imperatives), to its syntactic status (as for predicative/adverbial vs. prenominal adjectives or attributive vs. substitutive pronouns), to semantic classes (e.g. different kinds of pronouns like demonstrative, indefinite or possessive pronouns), or to pure lexical classes (the word class PTKNEG (negated adverb) is represented by exactly one lexical form *nicht* (not); the same is true for all subclasses of the major class "particle" apart from the morphological class PTKVZ (verb particle)).

While all the major parts of speech contain at least two subclasses, the open word class ADV (adverb) is the only one which has not been subdivided any further. The STTS, in fact, does provide a part of speech tag PAV (pronominal adverb). This class is a purely morphologically or lexically defined class, which contains words with a prepositional and a pronominal component (words like *darauf* (literally: on that)). These words, however, are, similarly to prepositional phrases, syntactically extremely heterogeneous: they can occur as prepositional objects (*Ich warte* **darauf** (I am waiting for that)) or as adverbials (**Darauf** *solltest du nicht treten* (You should not step on that)). From a syntactic or functional perspective, only in the second case they can be regarded as adverbs. For that reason we, like most grammars, treat pronominal adverbs strictly as a morphological class which hierarchically stands above all syntactically motivated word classes and should not be mixed up with them.

According to the STTS, adverbs are defined as modifiers of verbs, adjectives, adverbs, or clauses, which are not derived from adjectives (p. 56). Since there are other parts of speech that can also modify each of these heads (e.g. modal particles, regular particles, pronominal adverbs, and ordinals), this definition is not sufficient. As a matter of fact, the category ADV in the STTS tagset can be described as a residual category. This situation is unsatisfactory for the annotation of corpora which are intended for the study of adverbs, particles, or one of the other parts of speech mentioned above. Therefore, we would like to propose a more fine-grained subcategorisation of the residual class ADV in the STTS tagset.

With regard to the fact that the part of speech category ADV in the STTS contains different word classes, we have divided the class ADV into "real" adverbs (ADV), modal particles (MODP), and other particles (PTK). The PTK category is further subdivided into focus particles (PTKFO), intensifiers (PTKINT), and lexical particles (PTKLEX). These classes are defined from a purely *functional syntactic* perspective, which does not incude semantic classes like temporal or manner adverbs which are specific semantic subcategories of the class ADV. Furthermore, we redefine the dissociation of adverbs (ADV) and adjectives (ADJD) in favour of a syntactically motivated notion of lexical modifiers. In the following section, we will first describe the newly defined classes which are already present in Schiller et al. (1999). Then we will discuss the new part of speech categories.

3.1 ADV vs ADJD

The distinction between the STTS categories ADV and ADJD is motivated inflectionally: Words that cannot be inflected and modify heads of any kind are, according to Schiller et al. (1999), p. 56, classified as adverbs (ADV). Words that can be inflected but are used as adverbials or predicatives are categorised as adjectives (ADJD) (see Schiller et al. 1999, p. 23). We argue, however, that this distinction is syntactically irrelevant and also hard to operationalise. Consider the following examples (7-12).

(7) Sie hat **behände**/ADV (?) den Baum
She has skilfully the tree
beklettert.
climbed.
"She has skilfully climbed the tree."

(8) Sie hat **elegant**/ADJD den Baum beklettert.
She has elegantly the tree climbed.
"She has elegantly climbed the tree."

(9) Sie hat **oft**/ADV den Baum beklettert.
She has often the tree climbed.
"She has often climbed the tree."

(10) Sie hat **häufig**/ADJD (?) den Baum
She has frequently the tree
beklettert.
climbed.
"She has frequently climbed the tree."

(11) Sie hat **wahrscheinlich**/ADJD (?) den
She has probably the
Baum beklettert.
tree climbed.
"She has probably climbed the tree."

(12) Sie hat **vielleicht**/ADV den Baum beklettert.
She has perhaps the tree climbed.
"Perhaps she has climbed the tree."

According to the STTS, the words in bold are assigned the tags shown above (examples (7)-(12)). However, from a syntactic perspective it is hard to justify that the different modifiers in (7)-(12) belong to fundamentally different categories; they have the same inflectional status, their distribution is exactly the same, and they have similar syntactic functions insofar as they are all modifying the main verb or are attached at a higher level in the respective sentence.[2] Since we assume that part of speech categories are often the basis for further syntactic analysis, this is our main argument against an inflectional morphological approach for distinguishing adverbs and adjectives. Furthermore, there are conceptional problems for the operationalisation offered in Schiller et al. (1999) and in many German grammars.

The different tags shown in (7)-(12) result from one particular feature of the modifier in question, namely from its *inflectibility* (+*infl.*→ADJD, -*infl.*→ADV). This means that if a given modifier can be used adverbially and at the same time prenominally, it has to be classified as ADJD. Since the feature *inflectibility* cannot be tested properly (there is, for instance, no general agreement on the question whether *hoffentlich* (hopefully) is inflectible or not), another syntactic test is given in the guidelines (Schiller et al. 1999, p. 57): If the word in question can be used as a predicative adjective, it has to be annotated as ADJD

[2] The different semantic classes have a different scope which has provable distinct syntactic effects. This is why different kinds of adverbials are not only discussed from a semantic, but also from a syntactic point of view. Here we subsume all different kinds of adverbs (like adverbial versus adsentential adverbs) under one category 'adverb' (ADV).

(*sie ist elegant*/ADJD (she is elegant); **das ist oft* (this is often) →oft/ADV).

Inflectibility and the ability to function as a predicate, however, are independent features; words can be uninflectible but, at the same time, be used as a predicate (*er ist pleite* (he is broke) – *ein *pleiter Mensch* (a broke guy)), and there also are inflectible forms which cannot be used as predicates (*der eigentliche Termin* (the actual date) – *der Termin ist *eigentlich* (the date is actual)).

Not only can the tests for distinguishing adjectives from adverbs provide contradictory outcomes, in many cases they simply fail. For instance, acceptability judgments by German native speakers do not give a clear picture on whether examples (13)-(15) are grammatical or not.

(13) Der Sprung war behände.
The jump was agile

(14) Der Vorfall war häufig.
The incident was frequent.

(15) eine wahrscheinliche Baumbesteigung
a probable tree climb

To get rid of the inflectibility criterion, we propose that all adverbial or adsentential modifiers (like the ones in 7-12) are analysed as adverbs, whereas uninflected adjectives have to be used as a syntactic predicate in order to be tagged as ADJD. This means that only complements of copula verbs are tagged as predicative adjectives.[3]

3.2 Particles

Since the residual category ADV in the STTS guidelines (Schiller et al., 1999) includes different kinds of particles (a fact not discussed in the guidelines themselves), we move these to the main class PTK of the STTS which, so far, includes the tags PTKA (particle with adjective or adverb), PTKANT (answer particle), PTKZU (*zu* (to) with infinitive), and PTKVZ (separated verb particle). Particles are modifiers which can not, on their own, stand in the German pre-field (Vorfeld) and which, in general, can not be moved around freely in the sentence but which are restricted to appearing adjacent to a specific lexical head. This can be tested easily by human annotators with the help of permutation tests – if a given modifier cannot be placed (alone) in the pre-field position, it will be analysed as a particle. We distinguish between three different types of particles.

3.2.1 Focus particles – PTKFO

Focus particles are associated with a given focus element and modify the set of alternatives which is connected with the focus itself. Consider examples (16) and (17) below.

(16) Petra ist **nur** zum KLETTERN
Petra is only for rock climbing
gekommen.
went.
"Petra only came for rock climbing"

In (16), the focus is on *klettern* (rock climbing). The particle *nur* (only) is associated with the focus and opens up a set of alternatives (any other activity). However, the modifier *nur* tells us that none of the other activities besides rock climbing should be considered in this context.

(17) Petra hat **sogar** UNTER dem Tisch
Petra has even under the table
nachgeschaut.
looked.
"Petra has even looked under the table."

In (17), the focus is *unter* (under), the set of alternatives includes any other positions in relation to the table, and the focus particle *sogar* (even) tells us that all the other possible alternatives are valid options as well (on the table, next to the table, ...).

3.2.2 Intensifiers – PTKINT

Intensifiers are expressions of graduation, intensification, or quantification. In most cases, they are modifying (gradable) adjectives or adverbs. In (18), *sehr* (very) is intensifying the adverb *kurz* (shortly) while in (19), *überaus* (extremely) strengthens the adjective *groß* (great).

[3] Please refer to Hirschmann (2014) for more detailed information on the distinction between adverbs and adjectives: `https://www.linguistik.hu-berlin.de/institut/professuren/korpuslinguistik/mitarbeiter-innen/hagen/hirschmann-adv-stts.pdf`.

(18) Petra ist **sehr** kurz zum
Petra is very shortly to the
Klettern gegangen.
rock climbing went.
“Petra went rock climbing for a very short time.”

(19) Petra hat **überaus** großen Hunger.
Petra has extremely great hunger.
”Petra is extremely hungry.“

3.2.3 Lexical particles – PTKLEX

Lexical particles are associated with a lexical head element with which they form a complex lexeme. In (20), for example, the complex lexeme *nicht mehr* (not any more) is composed of the head *nicht* and the lexical particle *mehr*, while in (21), we have a complex lexeme *immer noch* (still) with *noch* as the head. The meaning of the complex lexeme can not be derived by a compositional analysis of its individual components.

(20) Petra gefällt das [nicht **mehr**]
Petra pleases this not more
”Petra doesn’t like that any more“

(21) Petra gefällt das [**immer** noch]
Petra pleases this always still
”Petra still likes that“

3.2.4 Modal particles – MODP

Modal particles (like particles in general) are also not *vorfeldfähig*, meaning they can not on their own fill the pre-field position in a Standard German sentence. They can, however, be placed relatively freely within the German middle field (Mittelfeld), a crucial feature which does not apply to any other type of particle. Because of this – and also for other semantic-syntactical reasons (modal particles modify the sentential level of a given clause) – we consider modal particles as a distinct major class. Modal particles can be treated as a closed word class. Please refer to the tagging guidelines by Hirschmann (2014) for a comprehensive list of candidates.

4 Annotation experiment

To test the new classification, we applied it to 1000 sentences randomly selected from the TiGer treebank and reassigned labels to all tokens where the original tag was one of either ADJD (adverbially used or predicative adjective), ADV (adverb), or a past participle[4] (VAPP, VVPP). In the beginning, the annotators were presented with the original POS tags. As we had the impression that this influenced the annotators’ decision, we replaced all instances of the modifier tags with the same dummy tag.

POS	**# orig**	**# new**	**# agr.**	**Fleiss’ κ**
ADJD	191	74	63	0.891
ADV	445	378	343	0.800
MODP	-	12	6	0.515
PTKFO	-	80	67	0.797
PTKINT	-	63	49	0.788
PTKLEX	-	33	17	0.594
VAPP	21	21	21	1.000
VVPP	173	172	172	0.989
total	**830**	**833**	**88.3%**	**0.838**

Table 1: Distribution (orig, new) and agreement (percentage agreement and Fleiss’ κ) for the different tags

We started off with annotating samples of 100 sentences, then discussed the mismatches and updated the annotation guidelines. After having finished the first 400 sentences (samples 1-4), we annotated a larger batch including the remaining 600 sentences of our goldstandard. As we still made changes to the guidelines at this stage, we report inter-annotator agreement on an additional test set of 500 sentences from Tiger (sentence 9501-10000).

Our test set includes 830 instances of modifiers which had to be re-annotated (Table 1).[5] The annotators could assign one of the tags ADV, ADJD, MODP, PTKFO, PKTINT, PKTLEX, VAPP, VVPP. We achieved an inter-annotator agreement of 0.838 (Fleiss’ κ), and an overall percentage agreement for all modifier tags of 88.3%.

Table 1 also shows that modal particles (MODP) and lexical particles (PTKLEX) are the most difficult ones to annotate, maybe partly due to their low frequency in the corpus.

[4] We included past participles in the annotation as some of them had to be reannotated as ADJD → ADV.

[5] The numbers for the original data set and the re-annotated set vary slightly, as also some other instances not labelled as ADV or ADJD in TiGer have been assigned a new label, e.g. *”um/KOUI/PKTLEX so scheinheiliger“* (so much more sanctimonious).

	ADJD	ADV	PFO	PINT	PLEX	MODP
ADJD	63	6	0	0	0	0
ADV	6	343	15	6	6	5
PFO	0	12	67	2	1	0
PINT	0	9	0	49	2	0
PLEX	0	9	0	1	17	0
MODP	0	5	0	0	1	6

Table 2: Confusion matrix for adverbs (ADV), predicative adjectives (ADJD), focus-associated particles (PFO), intensifiers (PINT), lexicalised particles (PLEX) and modal particles (MODP)

4.1 Ambiguous cases

Below we show some examples where the annotators disagreed. The confusion of ADV and ADJD mostly concerned cases like (22) where the lexeme in question was interpretated as a verb modifier (ADV) by one annotator and as a predicative adjective by the other. These cases can be handled by providing more specific instructions in the annotation guidelines, e.g. by providing a list of potential copula verbs which link the subject to the adjectival predicate.

(22) ADV vs ADJD

Wer sich weigere, werde durch Drogen
Who himself refuses, is by drugs
gefügig gemacht
compliant made

"Who refuses is made compliant by drugs"

For the distinction between adverbs (ADV) and focus particles (PTKFO), many cases were indeed ambiguous (see example 23). It is not clear how much context should be taken into account in order to resolve the ambiguity in the sentence. In our experiments, we decided to only use the sentence context in order to speed up the annotation process, and to use the combined label ADV:PTKFO for those cases which could not be resolved during adjudication. However, often the annotators were only aware of one of the possible readings, which resulted in many disagreements for these tags.

(23) ADV vs PTKFO

Hennemann hatte seinen Rückzug **bereits**
Hennemann had his withdrawal already
im September angeboten.
in September offered.

"Hennemann had already offered his withdrawal in September."

Better agreement can be achieved especially for the lexicalised particles (24), which mostly consist of frequent, co-occurring lexemes. Many disagreements concerned new instances which had not been seen before. Listing the most frequent instances in the guidelines might improve inter-annotator agreement for PTKLEX.

(24) ADV vs PTKLEX

Diese werden **immer wieder** missbraucht
These become always again abused

"Again and again, these become abused"

5 Parsing experiments

This section presents a parsing experiment where we test the learnability of our new classification using a statistical dependency parser.

5.1 Data expansion

To obtain more training data than the manually annotated 1000 sentences, we extracted patterns from the goldstandard capturing the syntactic context in which each of the new tags might occur, and applied them to the whole TiGer treebank.

Example (25) shows such a pattern. It extracts all tokens #p which have a lemma form from a predefined list (*rund* (around), *etwa* (about), *kaum* (hardly), ...), which are assigned the grammatical function MO (modifier), and which are directly followed by a cardinal number which has the same mother node as #p. We use TiGerSearch for pattern extraction, identify the terminal ids of the #p nodes and assign the new tag PTKINT (intensifier) to all #p.

(25)
```
#p:[lemma=("rund"|"etwa"|...|"kaum")] &
#p . #card:[pos="CARD"] &
#mother >MO #p &
#mother > * #card
```

Another example is shown in (26). Here we look for a token with the POS tag ADV (adverb) which is the leftmost child of an NP and which has one of the following lemma forms: *allein*

Tag	gold	expanded
ADJD	142	478
ADV	686	3,289
MODP	18	36
PTKFO	161	675
PTKINT	135	516
PKTLEX	54	201
ambiguous tags		
ADJD:ADV	1	-
ADV:MODP	1	-
ADV:PTKFO	22	-
ADV:PTKINT	2	-
ADV:PTKLEX	1	-
PTKFO:PTKINT	1	-
Total	1,224	5,195

Table 3: Distribution of the different modifier classes in the goldstandard

(only), *auch* (also), ..., *zwar* (indeed). These instances are then relabelled as PTKFO (focus particles).

(26) #cat:[cat=”NP”] >@l #p:[pos=”ADV”] &
#p:[lemma=(”allein”|”auch”|...|”zwar”)]

Overall, we defined 49 different patterns, which assigned tags to 90.9% of the modifiers in the sample. Sometimes, these patterns overgeneralise. We manually checked potential errors in the first 5000 sentences of the treebank and manually annotated the remaining 478 cases which were not captured by our pattern approach. After the manual clean-up we had an additional data set with 4922 new sentences (86,517 tokens).[6] This dataset is not as “high-quality” as the 1000 sentences of the goldstandard which have been individually annotated from scratch by the authors, and where all disagreements have been resolved in discussion. However, as we do not evaluate the accuracy of the POS tags themselves but the impact of the new classification on parsing accuracy where we only evaluate the dependency labels and relations, this is not a problem for our experimental setup. Table 3 shows the distribution of our new tags in the goldstandard and in the expanded dataset.

	Malt		MATE	
fold	**orig**	**new**	**orig**	**new**
1	84.0	84.3	85.4	86.3
2	84.2	84.7	87.1	87.6
3	89.0	89.3	91.7	91.7
4	85.3	85.9	88.5	89.1
5	89.0	88.9	91.2	91.5
6	86.0	85.5	88.0	88.4
7	86.0	86.2	88.7	89.2
8	89.1	89.2	91.6	91.9
9	89.7	89.8	92.0	92.1
10	85.0	85.9	87.4	88.1
avg.	**86.7**	**87.0**	**89.2**	**89.6**

Table 4: Parsing results (Malt and MATE parsers, LAS) for original and new tags

5.2 Setup

The parsers we use in our experiments are the Malt parser (Nivre et al., 2007) and the MATE parser (Bohnet, 2010), both language-independent systems for data-driven dependency parsing. We trained the parsers on the first 5000 sentences from the TiGer treebank and evaluated them in a 10-fold crossvalidation setting. The parsers have been trained on two different versions of the data, a) on the original treebank trees, and b) on the same trees, but replacing the original POS tags with our new POS classification.

For each version of the data, we separately optimised the parameters for the Malt parser, using MaltOptimizer (Ballesteros and Nivre, 2012), and then trained the parser with the parameter and feature settings optimised for each dataset.

5.3 Results

Table 4 shows labelled attachment scores (LAS) for the 10 folds and averaged scores for the whole dataset. For both, Malt and MATE parser, we observe a small, but highly significant difference between the two datasets.[7]

This difference becomes more substantial when only looking at the modifier (MO) dependency relation. Table 5 shows precision, recall and f-score for the 10 folds and results averaged over all folds for the combined evaluation of dependency relation and attachment for the label

[6] 78 of the 5000 sentences were already included in the goldstandard.

[7] For significance testing we used Dan Bikel’s Randomized Parsing Evaluation Comparator with $n = 10000$.

		orig			new		
fold	freq.	prec.	rec.	f1	prec	rec.	f1
1	1301	72.2	70.4	71.3	76.2	74.5	75.3
2	1261	73.9	71.7	72.8	76.5	73.8	75.2
3	916	78.4	76.3	77.3	81.1	77.5	79.2
4	1159	74.2	73.5	73.8	77.9	77.0	77.5
5	1031	76.4	75.7	76.1	79.7	79.1	79.4
6	1125	75.1	74.9	75.0	76.7	77.0	76.8
7	1151	75.2	73.6	74.4	77.8	76.7	77.3
8	978	76.9	78.2	77.6	80.0	79.6	79.8
9	867	81.8	79.2	80.5	82.2	80.5	81.3
10	1081	73.6	73.4	73.5	77.2	78.5	77.8
avg.	1087	75.8	74.7	**75.2**	78.5	77.4	**78.0**

Table 5: Precision, recall and f-score for dependency relation and attachment for MO (MATE parser)

MO.[8] Here the gap is nearly 3 percentage points (MATE parser), giving evidence that our syntactically motivated classification of modifiers supports the parser in analysing these structures.

Table 6 shows that our new tag distinctions not only help when analysing MO dependencies but also improve results for other dependencies.

6 Conclusions and future work

The results presented in the paper are interesting in many ways. First of all, we proposed an extension to the STTS which gives a more detailed, as well as linguistically well-founded analysis of modifiers in German. This is of interest especially for spoken and conceptually spoken language such as CMC data, where modifiers are extremely frequent and an analysis based on the core STTS tags is not very informative. Second, we presented an annotation study where we tested the applicability of the new classification to newspaper text. We discussed the problems arising during annotation, which are mostly based on real ambiguities in the data. The new annotations are available to the research community.[9]

Last, and most important, we gave proof of concept that a more detailed analysis of modification on the POS level which is linguistically motivated can indeed support data-driven syntactic parsing.

		orig			new		
DEP	freq.	prec.	rec.	f1	prec	rec.	f1
CJ	2497	84.5	83.1	83.8	85.0	83.4	84.2
DA	533	86.1	78.0	81.9	87.8	78.4	82.8
MNR	2618	64.9	67.5	66.2	65.3	68.6	66.9
NG	496	75.1	75.6	75.4	76.3	76.4	76.3
OP	846	57.8	33.0	42.0	57.7	33.6	42.4
PD	879	77.2	70.2	73.5	81.5	71.3	76.1
RE	272	58.5	50.7	54.3	64.0	53.7	58.4
SBP	182	71.5	78.6	74.9	76.0	80.2	78.1

Table 6: Precision, recall and f-score for other dependency relations (and attachment) where the new tags improved results (MATE parser; CJ: conjunct, DA: dative object, MNR: postnominal modifier, NG: negation, OP: prepositional object, PD: predicate, RE: repeated element, SBP: passivised subject)

So far, we have only shown that our new classification scheme does improve data-driven syntactic parsing of modification relations when providing the parser with gold (or, as for our extended dataset, with nearly gold standard¡) tags. It remains to be shown that the new tags can be learned by a POS tagger (or parser) with sufficient accuracy to be useful to the parser. Also, the parsing results are based on a small testset only and thus need to be validated on a larger dataset. Additional annotations are under way, and we plan to address both issues in future work.

Acknowledgments

This work was supported by a grant from the German Research Association (DFG) awarded to SFB 632 “Information Structure” of Universität Potsdam, Humboldt Universität Berlin and Freie Universität Berlin, Project B6: “The Kiezdeutsch Korpus (KiDKo)”.

References

Miguel Ballesteros and Joakim Nivre. 2012. Maltoptimizer: An optimization tool for maltparser. In *Proceedings of the Demonstrations at the 13th Conference of the European Chapter of the Association for Computational Linguistics*, EACL ’12.

Bernd Bohnet. 2010. Very high accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING ’10, pages 89–97.

Sabine Brants, Stefanie Dipper, Silvia Hansen, Wolfgang Lezius, and George Smith. 2002. The TIGER

[8] For the evaluation we used a slightly modified version of the CoNLL07 evaluation script provided by `http://pauillac.inria.fr/˜seddah/eval07.pl`.

[9] Download from `https://www.linguistik.hu-berlin.de/institut/professuren/korpuslinguistik/mitarbeiter-innen/hagen/tiger_adv.tgz`

treebank. In *Proceedings of the First Workshop on Treebanks and Linguistic Theories*, pages 24–42.

Mary Dalrymple. 2006. How much can part-of-speech tagging help parsing? *Natural Language Engineering*, 12(4):373–389.

Markus Dickinson. 2006. An investigation into improving part-of-speech tagging. In *Proceedings of the Third Midwest Computational Linguistics Colloquium (MCLC-06)*, Urbana-Champaign, IL.

Hagen Hirschmann. 2014. Richtlinien zur Wortartenannotation von Adverb- und Partikelklassen – eine Granularisierung des STTS im Bereich von Modifikatoren. Technical report, Humboldt-Universität zu Berlin.

Peter Koch and Wulf Oesterreicher. 1985. Sprache der nhe – sprache der distanz. mündlichkeit und schriftlichkeit im spannungsfeld von sprachtheorie und sprachgeschichte. *Romanistisches Jahrbuch*, 36:15–43.

Sandra Kübler and Wolfgang Maier. 2014. über den Einfluss von Part-of-Speech-Tags auf Parsing-Ergebnisse. *Journal for Language Technology and Computational Linguistics*, 1(28):17–44.

Andrew MacKinlay and Timothy Baldwin. 2005. Pos tagging with a more informative tagset. In *Proceedings of the Australasian Language Technology Workshop 2005*, pages 40–48, Sydney, Australia.

Joakim Nivre, Johan Hall, Jens Nilsson, Atanas Chanev, Gülsen Eryigit, Sandra Kübler, Svetoslav Marinov, and Erwin Marsi. 2007. MaltParser: A language-independent system for data-driven dependency parsing. *Natural Language Engineering*, 2(13):95–135.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *The 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics (ACL-COLING)*.

Slav Petrov, Dipanjan Das, and Ryan McDonald. 2012. A universal part-of-speech tagset. In *The 8th International Conference on Language Resources and Evaluation (LREC-12)*, May.

Barbara Plank, Dirk Hovy, and Anders Søgaard. 2014. Learning part-of-speech taggers with inter-annotator agreement loss. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics*, Gothenburg, Sweden.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textkorpora mit STTS. Technical report, Universität Stuttgart, Universität Tübingen.

Detecting Relational Constructions in German Texts Automatically

Oliver Hellwig
University of Düsseldorf, SFB 991
hellwig7@gmx.de

Wiebke Petersen
University of Düsseldorf, SFB 991
petersen@phil.uni-duesseldorf.de

Abstract

Löbner's theory of concept types and determination (CTD) claims that nouns show different distributional fingerprints depending on their lexical determination class. In order to investigate this hypothesis, corpora annotated by determination patterns are needed (definiteness and relationality). In the paper, we focus on the harder of the two annotation tasks, the automatic detection of relational constructions. We apply a standard NLP pipeline and classification algorithms. Using a combination of symbolic and statistical approaches, we achieve a precision of 94.4% and a recall rate of 88.6% for the correct structural annotation of relational NPs.

1 Motivation and previous research

The paper describes the design and the evaluation of a set of classifiers that identify relational constructions within complex German noun phrases. The classifiers are intended to detect possessive relational structures consisting of the two arguments 'possessor' (POSS) and 'possessum' (PUM) and to label these arguments correctly.

Our work is embedded in a bigger research project that investigates Löbner's theory of concept types and determination (CTD). According to Löbner, on the lexical level there are four basic noun types that differ with respect to two binary features, namely inherent uniqueness, $\pm U$, and inherent relationality, $\pm R$ (Löbner, 1985; Löbner, 2011). Inherently unique nouns describe a unique referent in a given context, e.g., 'pope', 'Mary' or 'head [of]' versus 'tree', 'stone' or 'student [of]'. Inherently relational nouns are semantically unsaturated. They describe their referents by a relation to a possessor argument that must be found in their contexts of utterance, e.g., 'trunk [of]' or 'sister [of]' versus 'tree' or 'woman'. With respect to the two binary features $\pm U$ and $\pm R$ four basic noun types can be distinguished (see table 1: Non-relational nouns are either *sortal* $[-U, -R]$ or *individual* $[+U, -R]$ while relational nouns are either *functional* $[+U, +R]$ or *proper relational* $[-U, +R]$).[1]

According to Löbner, each noun type is linked to a 'natural' mode of determination, its *congruent determination*: $+U$ is linked to definiteness and $+R$ to possessive determination. In case of congruent determination, sortal nouns occur in indefinite, absolute uses ('a stone', $[Det_{-U}, Det_{-R}]$), individual ones in singular definite, absolute uses ('the pope', $[Det_{+U}, Det_{-R}]$), proper relational ones in indefinite, relational uses ('a sister of Mary', $[Det_{-U}, Det_{+R}]$), and functional nouns in singular definite, relational uses ('the head of Mary', $[Det_{+U}, Det_{+R}]$). However, in concrete utterances nouns often appear in *incongruent determination* modes that use other determination types. Such incongruent uses trigger coercions or type shifts and increase the semantic complexity. For example, typical anaphoric uses of nouns involve a pragmatic shift from non-uniqueness to uniqueness as in 'A woman entered the room. [...] Mary knew *the* woman from school.' Other examples for type shifts are generic uses like 'Mary acts like a pope' or metaphorical shifts like 'Mary bought a Picasso'. Many languages reflect the conceptual shifts grammatically and thus provide strong evidence for Löbner's claim that the type of a noun is fixed in the lexicon.[2] The classifiers described in

[1] For an early, independent distinction of functional and proper relational nouns along the same lines see de Bruin and Scha (1988).

[2] Languages with article split systems mark shifts ex-

	non-unique reference [-U]	unique reference [+U]
non-relational [-R]	**sortal noun** tree, stone, woman	**individual noun** pope, universe, Mary
relational [+R]	**proper relational noun** sister [of], student [of], page [of]	**functional noun** mother [of], dean [of], cover [of]

Table 1: The four basic noun types according to (Löbner, 2011).

this paper are designed for determining the possessive determination of nouns, thus constituting the first building block for a large-scale evaluation of Löbner's theory.

A central hypothesis in Löbner (2011, p. 29) states that incongruent determinations are less frequent than congruent ones. Thus, the distribution of nouns within different patterns of determination depends on their lexical referential properties. Horn and Kimm (2014) investigate this hypothesis for German on the basis of a small hand-annotated corpus (363 noun tokens), in which they find a significant correlation between $[+R]$ nouns and relational determination (Det_{+R}). While only $3,9\%$ of all $[-R]$ nouns are used with Det_{+R}, 33% of all $[+R]$ nouns are used with Det_{+R}. For Det_{+U} one can compute from the reported data that 78% of all $[+U]$ nouns are used with Det_{+U}, while 41% of the $[-U]$ nouns are used with Det_{+U}. However, the reported numbers concerning the $\pm U$ distinction are problematic, because the authors decided to analyze only the first occurrence of every meaning variant of a noun (236 types). This approach probably results in too low numbers for Det_{+U}, as it does not consider definites in anaphoric uses.

Our research project aims at investigating the influence of lexical noun types on their determination types in a significantly larger German corpus by using unsupervised annotation of determination modes. If Löbner's distributional hypothesis is true, the four lexical noun types should correspond to sharp distributional fingerprints. In addition, it should be possible to predict the lexical type of a noun given its typical determination mode. Because the German article system explicitly distinguishes between definite and indefinite determination, it is comparatively easy to automatically annotate the definiteness status ($Det_{\pm U}$) of nouns in German texts. In this paper we present the classifier we developed for the automatic identification of possessive noun uses ($Det_{\pm R}$) in German. As Det_{+R} occurs in structurally very different constructions such as left and right attached genitives or attached PP's, to our knowledge, no former study has aimed at the same classification task.

Related research has been done in the field of prepositional phrase (PP) attachment disambiguation. Within the class of relational constructions, noun phrases (NPs) with PPs attached to nouns (e.g., '*the house of Mary*', '*the boy with red hair*') form a dominant subclass. Automatic PP attachment in German is error-prone, because constructions such as '*den Knochen vom Hund aufheben*' ('to pick up the bone of the dog', noun attached PP) and '*den Knochen vom Boden aufheben*' ('to pick up the bone from the ground', verb attached PP) have to be disambiguated (Rehbein and van Genabith, 2007; Volk, 2006; Kübler et al., 2007).[3] Volk (2002) uses a combination of supervised and unsupervised classification methods for PP attachment in German. The author reports an accuracy for PP noun attachments of 83.92% for a purely unsupervised constructed decision tree with a coverage of 90.13%; for a supervised con-

plicitly along the definiteness dimension (Ortmann, 2014; Lyons, 1999). Relationality in noun semantics and the status of shifts along this dimension are heavily debated (Partee and Borschev, 1998; Partee and Borschev, 2003; Jensen and Vikner, 2004; Barker, 1995; Barker, 2011; Petersen and Osswald, 2014). These shifts become explicit in the different acceptability rates of constructions like 'that team is John's' vs. (#) 'that brother is John's' (Partee and Borschev, 2003) or 'the sister of Shakespeare' vs. (#) 'the knife of Shakespeare' (Søgaard, 2005).

[3] Sentences such as '*Peter bekommt ein Buch von Maria*' ('Peter gets a book of / from Maria') are ambiguous without further context even for a human reader, because Peter may either get one of Maria's book, or he may get a book personally from Maria.

structed decision tree with a coverage of 100% using a back-off strategy, he obtains an accuracy rate of 77.19% for PP noun attachments, which increases to 83.65% by combining both methods.

As we aim at labeling a special class of semantic relations between nouns, our work is further related to the area of semantic role labelling (SRL) (Gildea and Jurafsky, 2002; Pradhan et al., 2004, SRL) and familiar tasks such as attribute identification for knowledge representation (Poesio and Almuhareb, 2005) or taxonomy learning (Cimiano et al., 2004). The complexity of our approach occupies an intermediate position between the tasks of joint dependency parsing and SRL and of SRL-only as described in (Hajič et al., 2009), because we rely on silver dependency parses produced by MATE, but do not provide any information about the target words to the labeling algorithms.

The paper is structured as follows. Section 2 gives an overview of the corpus used for building and testing the classifiers, and of the features and classification algorithms used. Section 3 reports the results we achieved in our experiments, and discusses errors made by the system. Section 4 summarizes the paper and provides an outlook into future improvements and applications.

2 Data and Features

This section describes the data from which the classifiers are built and on which they are tested. In addition, it provides an overview of the features used for classification and of the classification algorithms themselves.

2.1 Corpus

The annotated data come from two sources. (i) A seed corpus containing 300 sentences from five different fictional and newspaper texts that has been annotated with relational structures (Horn and Kimm, 2014). Unfortunately, the authors don't report the inter-annotator agreement for the seed corpus. (ii) The main part of our test and training data comes from the Leipzig Corpora collection (Quasthoff et al., 2006), from which a subset of 800 sentences was randomly drawn. Independently from each other, two annotators used MMAX (Müller and Strube, 2006) to annotate these sentences with relational structures. They marked the head of each relational construction as PUM, the subordinate part as POSS and the type of relation between POSS and PUM for each instance of a relational construction. In the chunk "der Bürgermeister von Berlin", for example, the phrase "von Berlin" is marked as possessor (POSS), while "der Bürgermeister" is "possessed" by the city of Berlin and therefore marked as PUM.

	A2.POSS	A2.PUM	A2.no-poss	total
A1.POSS	951	25	264	1240
A1.PUM	54	503	208	765
A1.no-poss	150	103	13941	14194
total	1155	631	14413	16199

Table 2: Annotator agreement ('A1.POSS' stands for 'annotator 1 has annotated a word as 'POSS')

The resulting annotations were merged by a supervisor (adjudicator) and formed the corpus used for the experiments described in section 3. The annotation results for the main corpus are given in table 2. The annotator agreement is $\kappa = 0.767$ in terms of Fleiss' kappa (Fleiss, 1971). When taking into account that the maximal value for κ is 0.936 given the marginal totals in table 2, our annotator agreement reaches 81,9% of this value.[4] In order to overcome data sparsity, we decided not to use all types of relations annotated by Horn and Kimm (2014), but only the four main classes *rgen*, *lpron*, *rvon*, and *lgen*.[5] Non-relational constructions and low-frequency relational classes were subsumed under the label *no-poss* (no relation).

[4] As we did not provide an annotation guideline for this pilot study, we expect that the agreement rates will increase in successive annotations. For example, some disagreements in table 2 result from the fact that one of the annotators initially did not mark the determiners in a PUM or POSS expression as belonging to this expression.

[5] The following abbreviations are used [frequencies in our corpus are given in brackets]: *rgen* [180]: genitive to the right (e.g. '*das Haus des Mannes*', 'the house of the man'), *lpron* [120]: possessive pronoun to the left (ex.: '*sein Haus*', 'his house'), *rvon* [13]: preposition '*von*' to the right (ex. '*das Haus von Peter*', 'the house of Peter'), *lgen* [12]: genitive to the left (ex. '*Peters Haus*', 'Peter's house'), *no-poss* [4915]: no relation.

Type	Frequency
no relation (*no-poss*)	4915
Right genitive (*rgen*)	180
Possessive pronoun (*lpron*)	120
Right '*von*' (*rvon*)	13
Left genitive (*lgen*)	12

Table 3: Absolute frequencies of selected possessive classes in the seed corpus; word-based count

2.2 Features

Following the design of SRL systems described in Pradhan et al. (2004) and Gildea and Jurafsky (2002), we combined structural, lexicographic and grammatical information into a multidimensional feature vector V_i for each word w_i in the corpus. Let x_i denote the tuple consisting of the following five atomic features: (1) the surface form of a word w_i, (2) its lemma, (3) its POS tag, (4) its case (if applicable) and (5) a binary feature indicating if w_i ends on *s*, which marks the genitive singular in German non-feminine nouns. The last atomic feature has been included because an evaluation of intermediate classification output showed that the POS tagger missed some instances of genitive constructions in which named entities were involved. Now, the feature vector V_i for word w_i at position i is built from the tuple x_i for the word w_i itself and from the respective tuples for all words occurring in an maximal absolute distance of 2 from w_i. Therefore, it has the form $V_i = \{x_{i-2}, x_{i-1}, x_i, x_{i+1}, x_{i+1}, x_{i+2}, x_{p_i^?}\}$, where $x_{p_i^?}$ denotes the tuple for the syntactic parent of w_i in the dependency tree that was constructed using the MATE parser (Bohnet and Nivre, 2012). As each tuple x consists of five atomic features and V_i can consist of five or six (if the syntactic parent is present) of these tuples, each word w_i is described by $5 \times 5 = 25$ or $5 \times 6 = 30$ atomic features. It should be noted that no manually annotated gold information was presented to the system. Instead, the features were generated from the silver output of MATE only.

As an example for how to build the full feature vector V_i, consider the sentence '*Marie wischte über das Ceranfeld des Herdes*' ('Marie wiped over the ceramic cooktops of the stove'). Figure 1 displays the simplified dependency tree created using MATE. The feature vector for the word '*Ceranfeld*' is constructed from the feature tuples x for the words '*über*', '*das*', '*Ceranfeld*', '*des*' and '*Herdes*', and from the feature tuple $x_{p_i}^?$ for the word '*wischte*', which is the syntactic parent of '*Ceranfeld*' in this tree. Therefore, V_i has the following form:

$$\left\{ \begin{pmatrix} x_{i-2} \\ \text{über} \\ \text{über} \\ \text{PREP} \\ \text{-} \\ \text{false} \end{pmatrix}, \begin{pmatrix} x_{i-1} \\ \text{das} \\ \text{der} \\ \text{ART} \\ \text{acc} \\ \text{true} \end{pmatrix}, \begin{pmatrix} x_{i} \\ \text{Ceranfeld} \\ \text{ceranfeld} \\ \text{N} \\ \text{acc} \\ \text{false} \end{pmatrix}, \begin{pmatrix} x_{i+1} \\ \text{des} \\ \text{der} \\ \text{ART} \\ \text{gen} \\ \text{true} \end{pmatrix}, \begin{pmatrix} x_{i+2} \\ \text{Herdes} \\ \text{herd} \\ \text{N} \\ \text{gen} \\ \text{true} \end{pmatrix}, \begin{pmatrix} x_{p_i}^? \\ \text{wischte} \\ \text{wischen} \\ \text{V} \\ \text{-} \\ \text{false} \end{pmatrix} \right\}$$

2.3 Classifiers

Based on the dependency trees given by MATE and the multidimensional feature vectors described above, we applied symbolic and statistical models to the problem of detecting relational structures in complete German sentences. As a symbolic model, we defined hard-coded tree automata that identify the four main types of relations in the MATE dependency trees. The following rule base was used in our experiments (the head of the arrow points to the head of the syntactic construction):[6]

- *rvon* $\equiv$ N $\leftarrow$ von $\leftarrow$ (N$\vee$ NE)
- *lpron* $\equiv$ N $\leftarrow$ PRPOSS
- *rgen* $\equiv$ N $\leftarrow$ N_{gen} $\leftarrow$ ART
- *lgen* $\equiv$ N $\leftarrow$ NE_{gen}[7]

An inspection of the initial results showed that the rule base missed numerous instances of *rgen* in which the genitive case was not labeled correctly by MATE. Therefore, we rewrote the rule for *rgen*

[6] Abbreviations: N: noun, NE: named entity, PRPOSS: possessive pronoun, ART: article

[7] The construction *lgen* $\equiv$ N $\leftarrow$ N_{gen} occurs only rarely in modern standard German. Therefore, the right side of the rule was tightened from N_{gen} to NE_{gen} to reduce the number of misclassifications.

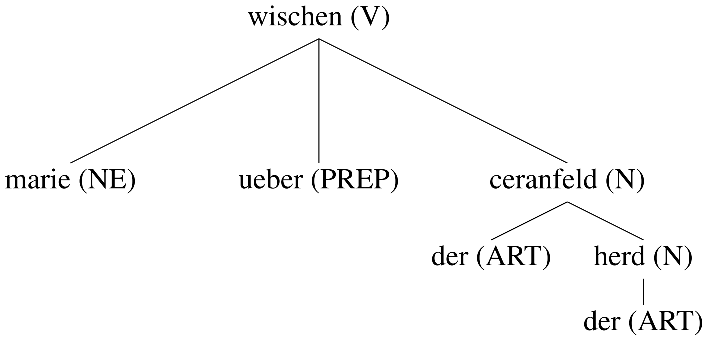


Figure 1: Simplified MATE dependency tree for the sentence '*Marie wischte über das Ceranfeld des Herdes*'.

to N ← N ← ART, as this construction should cover most instances of *rgen* in correctly parsed German, even if the morphological analysis of the construction is not correct.

The second group of models consisted of (sequential) machine learning (ML) algorithms. Using sequential ML algorithms appeared to be promising, because the local context of the relational types of relations is comparatively small in most cases (refer to the last column of table 8). Therefore, it should be possible to capture this context reliably using sequence classification algorithms, or even non-sequential classification algorithms such as Maximum Entropy, when they are trained with context features. In addition, the ML algorithms also use linear surface features such as lexical, morphological or POS information and may, therefore, be able to reduce the influence of wrong parses on the classification result. We applied Maximum Entropy (ME) (Ratnaparkhi, 1998) as a non-sequential method, and Conditional Random Fields (CRF) (Lafferty et al., 2001) and Hidden Markov Support Vector Machines (SVM$^{\mathrm{HMM}}$) (Altun et al., 2003) as sequential classification methods. All selected statistical classifiers can work with high-dimensional features spaces, which makes them a good choice for incorporating local lexical information. ME and CRF were run with the standard parameters provided by their implementations in the OpenNLP package (ME) and CRF-suite[8] (CRF). For SVM$^{\mathrm{HMM}}$, we used C classification with $C = 1000$ and $\epsilon = 0.1$. As the feature vectors V_i that were defined above consist only of features from nominal scales, they were fed into the ML algorithms without further data preprocessing.

3 Experiments and discussion

Table 4 displays the results of the first experiment. The numbers were calculated using a 30-fold cross-validation with a set of 300 annotated sentences from the seed corpus, and 800 sentences that were annotated during the project. For each run of the cross-validation, the set of 1100 sentences was split into disjoint training and test sets. Evaluation was only performed on the test sets. The values are calculated as strict word-based matches using standard measures of precision (P), recall (R) and F-score (F). Table 5 also records the evaluation of the negative no-poss class, because this label will indicate congruent determination for non-relational nouns, which is relevant for the evaluation of Löbner's theory.

The results displayed in Table 4 confirm our expectations about the classifier performance from section 2.3. The symbolic approach (*tree*) retrieves a large number of items, but is comparatively error prone. A typical example for a false positive generated by the tree is the chunk '*sei seine [Partei]$_{PUM}$ [der Auffassung]$_{POSS}$, ...*' ('his party has the opinion that ...'), where the PP '*der Auffassung*' is wrongly attached to the preceding noun instead to the verb. However, the symbolic algorithm performs well in marking long relational substructures as in '*[Vorlage]$_{PUM}$ [des von Premierminister Tony Blair zuvor als 'endgültig' angekündigten Pakets]$_{POSS}$*'. In general, the performance of this approach is largely dependent on the quality of the dependency trees

[8] http://www.chokkan.org/software/crfsuite/

	SVM^{HMM}		CRF		ME		Tree	
	P	R	P	R	P	R	P	R
no-poss	97.8	99.3	97.3	99.2	95.0	99.7	98.3	95.2
POSS	90.8	79.6	88.5	75.3	**91.9**	50.8	66.7	**82.2**
PUM	91.4	75.5	91.9	70.5	**94.8**	52.2	55.1	**76.3**

Table 4: Word-based evaluation by classifier, 30-fold cross-validation

		SVM^{HMM}		CRF		ME		Tree	
		P	R	P	R	P	R	P	R
lgen	POSS	93.15	**71.58**	**94.83**	57.89	93.48	45.26	83.12	67.37
lgen	PUM	97.5	**53.42**	97.22	47.95	**100**	28.77	84.21	43.84
lpron	POSS	96.51	**92.74**	**96.93**	88.27	99.3	78.77	92.31	87.15
lpron	PUM	**99.49**	81.07	99.47	77.37	98.18	66.67	96.76	**86.01**
rgen	POSS	99.25	**83.17**	**99.37**	78.64	99.12	56.28	96.62	82.66
rgen	PUM	96.75	**78.63**	96.54	73.61	**97.66**	55.15	89.7	78.1
rvon	POSS	**98.4**	58.57	96.9	59.52	94.74	8.57	95.6	**82.86**
rvon	PUM	94.23	63.64	**95.45**	54.55	91.67	14.29	94.55	**67.53**

Table 5: Word-based evaluation by classifier and relational type, 30-fold cross-validation

generated by MATE (refer to Bohnet and Nivre (2012, 1460-63) for a discussion of the parsing accuracy). Among the statistical methods, sequential algorithms (CRF, SVM^{HMM}) clearly outperform ME in terms of recall, because ME misses numerous long-distance structures.

Table 5 presents a more detailed view of the type labels assigned to the positive cases from the gold data. The table shows that precision and recall vary strongly among the target classes and the classifier types: The class *lgen* involves named entities (NE), which are often lexically and grammatically mislabelled. The *rvon* constructions are difficult to distinguish from '*von*' phrases attached to the verb (see examples in section 1). It should be noted that the highest differences are observed for *rvon* constructions for which the tree classifier outperforms the statistical ones in term of recall. Moreover, the surface form in which the head (POSS) of *rgen* structures appears is frequently identical with other inflected forms of the same lemma. The fact that NPs such as '*[die Wut]*$_{\text{NOM}}$ *[der Arbeiter]*$_{\text{GEN}}$' tend to be analyzed as '*[die Wut]*$_{\text{NOM}}$ *[der Arbeiter]*$_{\text{NOM}}$' complicates relation detection in complex NPs, even if the dependency tree is constructed correctly.

Type	P	R	F
POSS	91.6	80.3	85.6
PUM	93.2	74.4	82.7

Table 6: Results of merging decisions from table 5 using majority vote

	P	R	F
Full matches	93.58	87.14	90.24
Partial matches	94.38	88.64	91.42

Table 7: Structure-based evaluation, using merged results (refer to table 6)

Since the classifier types have different areas of specialization (see Table 5), we decided to merge the results of the four classifiers using majority voting. As can be observed from Table 6, this approach slightly improves the F-score of the POSS class to 85.6% when compared with the single classifier result for $\text{SVM}^{\text{HMM}}$from table 4. The low recall rates for *rvon* constructions that are reported in Table 5 are mainly due to the fact that the classifiers miss some long relational structures, as mentioned above. Therefore, we performed two further structure-based evaluations that use full and partial matches between complete relational structures in gold and silver data. Consider the gold-annotated sample phrase '*[das Haus]*$_{PUM}$

[von Peter und Maria]$_{POSS}$' ('the house of Peter and Maria'), which contains six words in relational structures. When a classifier produces the silver annotation '*[das Haus]*$_{PUM}$ *[von Peter]*$_{POSS}$ *und Maria*', 4 of 6 words are classified correctly in word-based evaluation (results in tables 5 and 6), whereas the classification has produced 1 partial resp. 0 full matches in structure-based evaluation. Results displayed in Table 7 demonstrate that recall problems observed in Table 5 are due to long phrases missed partially or completely by the ML algorithms, but not by a systematic inadequacy of these classifiers.These conclusions are supported by the numbers given in Table 8, which displays classifier performance grouped by the number of words contained in each relational structure. We counted the lengths L_R of all annotated ranges in the gold data, created two classes with $L_R \leq 4$ and $L_R > 4$, and performed a structure-based evaluation with partial matches (merged results) for both length classes. Table 8 shows that false negatives are significantly[9] more frequent among the long chunks than among the short ones, which leads to a strong decrease in recall.

	P	R	F	Total
long chunks ($LR > 4$)	97.7%	71.7%	82.7%	61
short chunks ($LR \leq 4$)	94.1%	90.4%	92.2%	597

Table 8: Performance rates for nominal chunks of different lengths; structure-based, partial matches

4 Conclusion and future research

The paper has described a system that detects and labels relational noun structures in German texts. Using a combination of symbolic and statistical classification algorithms, we were able to achieve precision of 94.4% and recall of 88.7% for structure-based evaluation with partial matches (refer to table 7). As mentioned in the introduction, the labeler is one building block in a large-scale evaluation of Löbner's theory of CTD, which is the main focus of our future work. However, given the frequency of relational structures in German texts, our results may also contribute to research on PP attachment, SRL and attribute learning in German. In addition, the evaluation has shown that the tree-based symbolic approach has strong advantages in detecting long range relational structures, but suffers in precision. Therefore, we are planning to increase the reliability of this approach by merging dependency trees constructed by MATE with trees from other engines such as the Stanford parser. At the level of classification, the present merging strategy of majority voting can certainly be replaced with more effective meta-learning strategies, which is the focus of our current research.

References

Yasemin Altun, Ioannis Tsochantaridis, and Thomas Hofmann. 2003. Hidden Markov Support Vector Machines. In *Proceedings of the Twentieth International Conference on Machine Learning*.

Chris Barker. 1995. *Possessive Descriptions*. CSLI Publications, Stanford.

Chris Barker. 2011. Possessives and relational nouns. In *Semantics: An International Handbook of Natural Language Meaning*, chapter 45, pages 1108–1129. de Gruyter.

Bernd Bohnet and Joakim Nivre. 2012. A transition-based system for joint part-of-speech tagging and labeled non-projective dependency parsing. In *EMNLP-CoNLL*, pages 1455–1465.

Philipp Cimiano, Aleksander Pivk, L Schmidt-Thieme, and S Staab. 2004. Learning taxonomic relations from heterogeneous sources. In *Proceedings of the ECAI 2004 Ontology Learning and Population Workshop*. Valencia, Spain.

Jos de Bruin and Remko Scha. 1988. The interpretation of relational nouns. In *Proceedings of the 26th annual meeting on Association for Computational Linguistics*, ACL '88, pages 25–32, Stroudsburg, PA, USA. Association for Computational Linguistics.

Joseph L. Fleiss. 1971. Measuring nominal scale agreement among many raters. *Psychological Bulletin*, 76(5):378–382.

Daniel Gildea and Daniel Jurafsky. 2002. Automatic labeling of semantic roles. *Computional Linguistics*, 28(3):245–288.

Jan Hajič, Massimiliano Ciaramita, Richard Johansson, Daisuke Kawahara, Maria Antònia Martí, Lluís Màrquez, Adam Meyers, Joakim Nivre, Sebastian Padó, Jan Štěpánek, Pavel Straňák, Mihai Surdeanu, Nianwen Xue, and Yi Zhang. 2009. The

[9] A Fisher-Yates test of the count data gives a p value of 0.0001757***.

CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of the Thirteenth Conference on Computational Natural Language Learning: Shared Task*, CoNLL '09, pages 1–18, Stroudsburg. Association for Computational Linguistics.

Christian Horn and Nicolas Kimm. 2014. Nominal concept types in German fictional texts. In Thomas Gamerschlag, Doris Gerland, Rainer Osswald, and Wiebke Petersen, editors, *Frames and Concept Types. Applications in Language and Philosophy*, volume 94 of *Studies in Linguistics and Philosophy*, pages 343–362. Springer Verlag.

Per Anker Jensen and Carl Vikner. 2004. The English pre-nominal genitive and lexical semantics. In Ji-Yung Kim, Yury Lander, and Barbara Partee, editors, *Possessives and Beyond: Semantics and Syntax*. GLSA Publications.

Sandra Kübler, Steliana Ivanova, and Eva Klett. 2007. Combining dependency parsing with PP attachment. In *Fourth Midwest Computational Linguistics Colloquium*, Purdue University.

John D. Lafferty, Andrew McCallum, and Fernando C. N. Pereira. 2001. Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning*, pages 282–289.

Sebastian Löbner. 1985. Definites. *Journal of Semantics*, 4(4):279–326.

Sebastian Löbner. 2011. Concept types and determination. *Journal of Semantics*, 28:1–55.

Christopher Lyons. 1999. *Definiteness*. CUP, Cambridge.

Christoph Müller and Michael Strube. 2006. Multi-level annotation of linguistic data with MMAX2. In Joybrato Mukherjee Sabine Braun, Kurt Kohn, editor, *Corpus Technology and Language Pedagogy. New Resources, New Tools, New Methods*, pages 197–214. Peter Lang, Frankfurt.

Albert Ortmann. 2014. Definite article asymmetries and concept types: Semantic and pragmatic uniqueness. In Thomas Gamerschlag, Doris Gerland, Rainer Osswald, and Wiebke Petersen, editors, *Frames and Concept Types*, volume 94 of *Studies in Linguistics and Philosophy*, pages 293–321. Springer International Publishing.

Barbara Partee and Vladimir Borschev. 1998. Integrating lexical and formal semantics: Genitives, relational nouns, and type-shifting. In R. Cooper and Th. Gamkrelidze, editors, *Proceedings of the Second Tbilisi Symposium on Language, Logic, and Computation*, April.

Barbara Partee and Vladimir Borschev. 2003. Genitives, relational nouns, and the argument-modifier ambiguitiy. In Ewald Lang, Claudia Maienborn, and Cathrine Fabricius-Hansen, editors, *Modifying Adjuncts*, pages 67–112. de Gruyter, Berlin.

Wiebke Petersen and Tanja Osswald. 2014. Concept composition in frames: Focusing on genitive constructions. In Thomas Gamerschlag, Doris Gerland, Rainer Osswald, and Wiebke Petersen, editors, *Frames and Concept Types*, volume 94 of *Studies in Linguistics and Philosophy*, pages 243–266. Springer International Publishing.

Massimo Poesio and Abdulrahman Almuhareb. 2005. Identifying concept attributes using a classifier. In *Proceedings of the ACL-SIGLEX Workshop on Deep Lexical Acquisition*, pages 18–27. Association for Computational Linguistics.

Sameer Pradhan, Wayne Ward, Kadri Hacioglu, James H. Martin, and Dan Jurafsky. 2004. Shallow semantic parsing using support vector machines. In *Proceedings of the HLT-NAACL*. 233-240.

Uwe Quasthoff, Matthias Richter, and Christian Biemann. 2006. Corpus portal for search in monolingual corpora. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation*, pages 1799–1802.

Adwait Ratnaparkhi. 1998. *Maximum Entropy Models for Natural Language Ambiguity Resolution*. Ph.D. thesis, University of Pennsylvania.

Ines Rehbein and Josef van Genabith. 2007. Treebank annotation schemes and parser evaluation for German. In *Proceedings of the 2007 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 630–639.

Anders Søgaard. 2005. The semantics of possession in natural language and knowledge representation. *Journal of Universal Language*, 6:85–115, September.

Martin Volk. 2002. Combining unsupervised and supervised methods for PP attachment disambiguation. In *Proceedings of COLING-2002*.

Martin Volk. 2006. How bad is the problem of PP-attachment? A comparison of English, German and Swedish. In *W06-2112, SIGSEM Workshop On Prepositions*.

Tagging Complex Non-Verbal German Chunks with Conditional Random Fields

Luzia Roth
Institute of Computational Linguistics
University of Zurich
luzia.roth@uzh.ch

Simon Clematide
Institute of Computational Linguistics
University of Zurich
simon.clematide@cl.uzh.ch

Abstract

We report on chunk tagging methods for German that recognize complex non-verbal phrases using structural chunk tags with Conditional Random Fields (CRFs). This state-of-the-art method for sequence classification achieves 93.5% accuracy on newspaper text. For the same task, a classical trigram tagger approach based on Hidden Markov Models reaches a baseline of 88.1%. CRFs allow for a clean and principled integration of linguistic knowledge such as part-of-speech tags, morphological constraints and lemmas. The structural chunk tags encode phrase structures up to a depth of 3 syntactic nodes. They include complex prenominal and postnominal modifiers that occur frequently in German noun phrases.

1 Introduction

In this paper[1], we report on comprehensive experimental results for a chunk tagging approach that recognizes complex non-verbal phrases such as nominal phrases (NP), prepositional phrases (PP), adjectival and adverbial phrases in German. We go beyond simple base chunks, that is, non-recursive and non-overlapping sequences of words. Base chunks were introduced and formalized as a sequence classification problem by Ramshaw and Marcus (1995) and popularized by a CoNLL shared task on chunking (Tjong Kim Sang and Buchholz, 2000). This problem is also known as the IOB chunk tagging problem because the chunk layer can be formulated as a sequence of tags expressing the begin (B) and continuation (I) of a chunk, or whether a token is viewed as being outside (O) of any chunk.

In contrast to the base chunk approach, we analyze the internal structure of complex phrases up to a maximal depth of 3 phrase structure nodes. As introduced by Skut and Brants (1998), structural chunk tags are needed that encode the hierarchical relation between adjacent tokens. Both, the IOB and the structural chunk tag approach can be treated as a sequence classification problem. We compare the performance of well-established sequence classifiers such as Hidden Markov Models (HMMs) with the state-of-the-art method of Conditional Random Fields (CRFs) on the TüBa-D/Z treebank (Telljohann et al., 2004), which is the largest collection of consistently annotated newspaper sentences in German.

The paper is organized as follows: In Section 2, we introduce the idea of structural chunk tags and present the data extraction and transformation from the treebank as well as the automatic linguistic enrichment of the raw data in preparation to the experiments. In addition, we describe the statistical tools and models used in our cross-validation experiments. In Section 3, we report the quantitative results of the experiments and discuss qualitative aspects of the most frequent errors.

2 Methods

Our approach is based on early work of Skut and Brants (1998). They introduced the term *chunk*

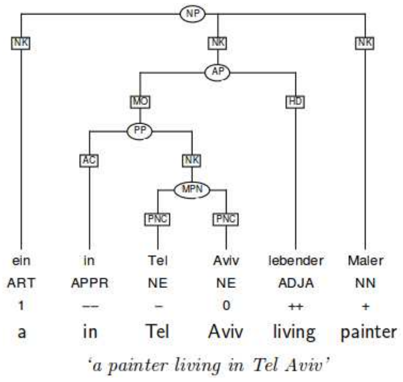


Figure 1: Complex NP annotated with structural tags as presented in Skut and Brants (1998). See Section 2.2.2 for an explanation of their chunk tags.

tagging for applying standard statistical PoS tagging techniques (i.e. HMMs) to the problem of chunking complex NPs and PPs. We extend their approach by using more data, more linguistic features, and more advanced statistical sequence classification methods to deal with this problem. Additionally, we investigate the question of how well post-nominal PPs can be identified by our improved approach.

2.1 Related Work

Skut and Brants (1998) developed a recognizer for complex chunk structures in order to create a tool for semi-automatic syntactic annotation. Their main idea was to extend chunking from a simple recognition of the boundaries of flat chunks to the calculation of nested chunk-internal syntactical structures. Given the outer boundaries of a chunk by a human annotator, their annotation system built the internal structures of chunks as complex as the one shown in Figure 1. They also evaluated their chunk tagger as a stand-alone application without human indication of chunk boundaries. This is more comparable to our experimental setting. They reached 90.9% of correctly tagged tokens using the NEGRA treebank (Skut et al., 1997) with a training corpus of 12,000 sentences. Due to the difficulties introduced by post-nominal attachment of NPs, PPs and focus adverbs, they trained and evaluated a chunk tagger without attachment of post-nominal NPs, PPs and adverbs. For this less complex task, they report a precision of 95.5%.

It is noteworthy that structural chunk tags can handle complex prenominal constructions as shown in Figure 1. IOB-style chunks typically need to disconnect the indefinite article from the nominal head of the NP (see Kübler et al. (2010) for a workaround). The NEGRA-derived German chunk tagger for flat noun, prepositional and verb chunks built on top of the TreeTagger (Schmid, 1994) shows exactly these limitations.

The recursive chunker from Kermes and Evert (2002) is based on a symbolic regular expression grammar and handles even complex prenominal constructions. It also deals with post-nominal NP attachment, but excludes post-nominal PP attachment due to the high degree of ambiguity.

Chunkers based on cascaded rules (e.g. Müller (2007)) or finite state transducer (for a more recent implementation see Barbaresi (2013)) can efficiently build shallow syntactic structure. Hinrichs (2005) contains an overview of several earlier approaches for German.

2.2 Data

For our experiments, we use the TüBa-D/Z corpus version 7.0, containing 65,524 sentences (henceforth referred to as TüBa)[2]. The corpus consists of newspaper articles with detailed morphological and syntactic annotations. This treebank is the largest for German and because of its topological and context-free grammar there are no discontiguous phrase structures as for example in the TIGER treebank (Brants et al., 2004).

2.2.1 Data Transformation and Enrichment

As can be seen in the upper tree of Figure 2, TüBa's phrase structures are deeply nested. For instance, the proper name 'Taake' is embedded at a depth of 6 phrase structure nodes. In order to be able to treat such complex PPs with our approach of limited chunk depth, we need to flatten the TüBa trees in the style of TIGER trees. The following transformations were applied:

1. The constituents of the dependent NP of a preposition are treated as immediate constituents of the PP. This approach has also been followed recently in the setting of multilingual dependency treebanks (McDonald et al., 2013).

[2] http://sfs.uni-tuebingen.de/ascl/ressourcen/corpora/tueba-dz.html

		Internal tagset (preceding / succeeding token): p/s			
Tagset from Skut and Brants		*External tagset: p*		*External tagset: s*	
`0`	if $m(w_i) = m(w_{i-1})$	`e`	if $m(w_i) = m(w_{i-1})$	`e`	if $m(w_i) = m(w_{i+1})$
`+`	if $m(w_i) = m^2(w_{i-1})$	`r`	if $m(w_i) = m^2(w_{i-1})$	`r`	if $m(w_i) = m^2(w_{i+1})$
`++`	if $m(w_i) = m^3(w_{i-1})$	`R`	if $m(w_i) = m^3(w_{i-1})$	`R`	if $m(w_i) = m^3(w_{i+1})$
`-`	if $m^2(w_i) = m(w_{i-1})$	`l`	if $m^2(w_i) = m(w_{i-1})$	`l`	if $m^2(w_i) = m(w_{i+1})$
`--`	if $m^3(w_i) = m(w_{i-1})$	`L`	if $m^3(w_i) = m(w_{i-1})$	`L`	if $m^3(w_i) = m(w_{i+1})$
`=`	if $m^2(w_i) = m^2(w_{i-1})$	`E`	if $m^2(w_i) = m^2(w_{i-1})$	`E`	if $m^2(w_i) = m^2(w_{i+1})$
`1`	else	`-`	not integrated into syntax structure	`-`	not integrated into syntax structure
		`0`	removed from syntax structure	`0`	removed from syntax structure
		`x`	chunk boundary	`x`	chunk boundary

Table 1: Comparison between Skut and Brants' tagset and our tagsets. Our data contains 50 different p/s tags out of 81 possible combinations.

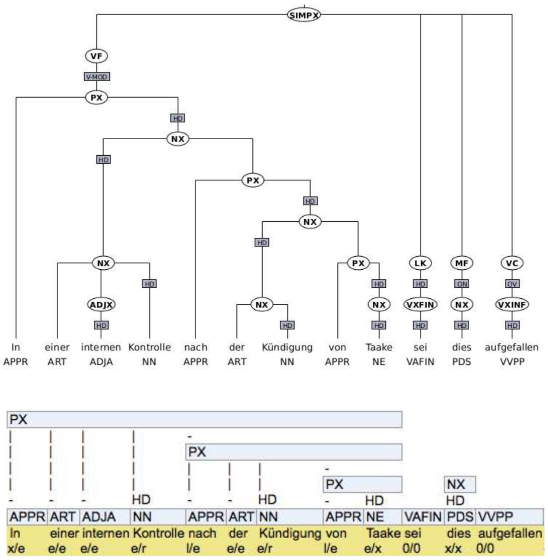


Figure 2: Example for the transformation of a deep syntactic phrase structure to the flattened chunk format. The structural chunk labels with our internal chunk tagset are on the last line. The shown sentence fragment translates as "*In an internal control after the termination of Taake this was noticed,...*".

2. The content of unary nodes which are non-heads in their mother constituent is directly attached to the mother node.

3. Coordinated unary nodes are directly attached to their mother nodes.

After the application of these transformations, all topological and verbal constituents were removed from the syntactic trees. All remaining phrase structures with a syntactic depth larger than 3 were removed. The final result of these transformations for the example sentence is shown in the lower part of Figure 2.

2.2.2 Internal and External Chunk Tagsets

For our experiments, we work with an enriched internal chunk tagset that encodes the structural relation of a token to its preceding (p) and succeeding (s) token. More fine-grained internal tagsets have proved to be profitable for statistical tagging approaches in the past (Brants, 1997). One goal of our experiments was to check whether this is also the case for chunk tags.

Table 1 compares Skut and Brants' tagset and our tagsets. An equation as $m(w_i) = m^2(w_{i-1})$ reads as 'the mother node of token w at position i is the grandmother node of the preceding token'. The depth of the hierarchical dominance relation m is given by its superscript. i specifies the linear position of a word in a sentence. Punctuation is never integrated in the syntactic structure (marked as '-'). Tokens connected to nodes (e.g. verbal) that were removed from the syntax structure are marked as '0'. Chunk tag 'x' indicates chunk boundaries. Figure 2 shows an example of the chunk encoding.

In our bidirectional internal tagset, an error often affects two tokens. This deteriorates the evaluation results because a single error will be counted twice. In a sentence like 'Aber weil der Koffer in einem unterirdischen See gelandet ist, [...]' ('*But because the suitcase has landed in an underground lake, [...]*') our system attaches 'in einem unterirdischen See' erroneously as a postnominal PP, resulting in two errors in the internal tagset as shown in Table 2. However, reducing the internal tagset to one of the external tagsets does not lead to a loss of information for the chunk structure. Therefore, we can train and label on the bidirectional internal tagset and map to an external before evaluation.

Tokens	*Gold*	*System*	*p/s*	*p/*	*/s*
der	x/e	x/e			
Koffer	e/x	e/r	X		X
in	x/e	l/e	X	X	
einem	e/e	e/e			
unterirdischen	e/e	e/e			
See	e/x	e/x			

Table 2: Error propagation in the internal tagset

2.2.3 Input Data Enrichment

The task of a chunk tagger is to compute the sequence of chunk tags (=outcome) for a given sequence of tokens (=evidence). However, directly using the raw text as the only evidence for predicting the outcome misses useful linguistic generalizations that are beneficial for this task. Therefore, we automatically enrich the raw text by PoS tags, normalized lemmas and morphological constraints.

First, we apply the TreeTagger 3.2 (Schmid, 1994) to compute PoS tags and lemmas from the raw text input. For unknown words, we use the tokens as the lemma. In order to reduce the sparse data problem, all lemmas are further normalized by reducing hyphenated compounds to their last segment, for instance '0:2-Niederlage' (*0-2 defeat*) is normalized to 'Niederlage'.

Second, morphological constraints for each PoS-tagged token are built from the output of GERTWOL, a commercial morphological analyzer (Koskeniemmi and Haapalainen, 1996). Morphological information is restricted to case and number and filtered according to the PoS tag computed by the TreeTagger. Because GERTWOL and the TreeTagger have slightly different categorizations of parts of speech, some tag mapping was necessary.

In German, word forms exhibit a lot of syncretism, especially between accusative and nominative case. In our current approach, we do not attempt to guess the correct analysis out of all admissible analyses, but we strive for a compact representation of the admissible as well as the excluded morphological categories. An 8 character string is used to encode these constraints in a systematic way where upper-case letters denote the admission of a category and lower-case letters denote the exclusion. Table 3 shows the actual encoding conventions. The morphological constraints of 'Häuser' (*houses*) are 'KNAdGUsP'. Word forms not known by GERTWOL are encoded by '????????' and punctuation tokens by '$$$$$$$$'.

Code	*Description*
K/k	With case/Without case
N/n	Nominative admissible/excluded
A/a	Accusative admissible/excluded
D/d	Dative admissible/excluded
G/g	Genitive admissible/excluded
U/u	With number/Without number
S/s	Singular admissible/excluded
P/p	Plural admissible/excluded

Table 3: Encoding of morphological constraints

Token	*PoS*	*Lemma*	*Morphology*	*Chk*
In	APPR	in	KnADgusp	x/e
einer	ART	ein	KnaDGUSp	e/e
internen	ADJA	intern	KNADGUSP	e/e
Kontrolle	NN	Kontrolle	KNADGUSp	e/r
nach	APPR	nach	KnaDgusp	l/e
der	ART	d	KNaDGUSP	e/e
Kündigung	NN	Kündigung	KNADGUSp	e/r
von	APPR	von	KnaDgusp	l/e
Taake	NN	Taake	????????	e/x
sei	VAFIN	sein	knadgUSp	0/0
dies	PDS	dies	KNAdgUSp	x/x
aufgefallen	VVPP	auffallen	knadgusp	0/0
,	$,	,	$$$$$$$$	-/-

Table 4: Representation of linguistic evidence and outcome (= column 'Chk')

Table 4 shows the result of the data enrichment process for our example sentence. In our experiments, we are interested to estimate the performance increase in chunk tagging that results from the morphological information, the PoS layer and the lemmas.

2.3 Tagging Structural Chunk Tags

As mentioned before, complex chunk structures in a sentence can be expressed by chunk tag sequences that correspond to the token sequence. Therefore, any sequence classification method can be applied to this problem. In our experiments, we focus on baseline methods based on HMM techniques and on state-of-the-art methods based on CRFs.

2.3.1 Chunk Tagging with Trigram Taggers

As a baseline, we use the HMM-based trigram tagger hunpos (Halácsy et al., 2007). This tool is an open-source reimplementation of the TnT tagger (Brants, 2000) that Skut and Brants (1998) developed and used for their work (see Section 2.1). A standard PoS tagger as hunpos has a predefined and limited model how the evidence for the classification of the outcome is used. In a typical trigram setting, this is the current token (lexical emission probability) and the preceding two outcome labels (transition probability predicted from the limited history of Markov models). These restrictions guarantee a very efficient training and labeling. Additionally, there is no need for a development set for training, which enables the user to split the available tagged material into a large training (90%) and a test set (10%). As an extension to the classical trigram tagging model, the hunpos tagger allows for condition the emission probability of a word w_i on the preceding and the current tag ($P(w_i|t_{i-1}t_i)$). This second order emission probability produced consistently better results in our chunk tagging experiments than a simple first order emission probability.

A disadvantage of HMM taggers is their restriction to a single layer of evidence. For instance, if we want to predict the chunk tags from the layer of PoS and morphology, we need to integrate that information in one combined evidence token. For example, in order to chunk the third token 'internen' from our example sentence based on the evidence of PoS and morphology we would encode the evidence layer as 'NN_KNADGUSP', i.e. the concatenation of PoS and morphology. CRFs are a lot more general in that respect, as they allow to have as many separate evidence layers as needed and to combine them freely into features.

2.3.2 Chunk Tagging with sequential CRFs

Sequential Conditional Random Fields (Sutton and McCallum, 2012) are state-of-the-art sequence classification models for typical NLP problems and have been shown to deliver excellent performance on the IOB-style chunking tasks (Sha and Pereira, 2003).

In our experiments, we use the freely available and efficient CRF tool wapiti (Lavergne et al., 2010). Unlike HMM tools, wapiti needs handcrafted feature templates that specify which information from which evidence layer is selected and combined in order to predict the outcome, i.e. the most probable sequence of structural tags for a sentence. Feature templates are a practical abstraction layer that allow the user to specify the model in a concise way without actually forcing the user to precompute the instantiated features for each position in the sequence. The CRF tool automatically instantiates the templates with the training material. During training, it learns the optimal weights for the instantiated features, and by using appropriate regularization, it is able to filter out irrelevant features. In all experiments reported in the evaluation section, we used the default settings of wapiti: L-BGFS for the optimization of the feature weights and elastic-net for regularization. wapiti requires a development set for training, therefore, the data was split into a training (72%), development (18%) and a test (10%) set.

Relative Position	*PoS Layer Example*
Current	NN
Preceding	ADJA
Succeeding	APPR
Preceding and current	ADJA/NN
Current and succeeding	NN/APPR
Preceding, current and succeeding	ADJA/NN/APPR
Two positions back and current	ART/NN
Current and two positions forth	NN/ART

Table 5: Local context of our best CRF feature template model. The second column illustrates the template with the instantiation on the PoS layer on position 4 (token 'Kontrolle') in our example sentence from Table 4.

Our best feature model. All evidence columns shown in Table 4 can be used to define feature templates. For a given position in the sequence, evidence from the current, preceding or succeeding positions can be combined. The amount and source of evidence packed in a feature is unbound in principle, however, for performance reasons evidence from the local context is most useful. In typical sequential CRF modeling tools, the evidence features can be automatically conditioned on outcome bigrams (preceding and current token, similar to the emission order of two of HMMs) or outcome unigrams (current token only). Bigram features can easily lead to feature explosion, long training times, and decreased performance (sparse data problem). We performed extensive tests for building an optimal set of feature templates. To our own surprise, a uniform and elegant set of unigram feature templates proved to be the best. The evidence layer of tokens could be ignored totally. For the layer of

PoS, lemmas and morphological constraints, we have exactly the same feature templates[3]. Table 5 shows the local context involved in our features and illustrates them by examples taken from the PoS layer. Only one bigram feature was used, namely the bigram output distribution of the chunk tags.

Alternative or more complex additional feature templates could not improve the performance. We tested for specific morphological cases (e.g. genitive), pattern matching for function words (e.g. articles), or combinations of evidence from PoS/morphology and lemma/morphology.

Our 25 feature templates instantiate about 118 million features (standard deviation (SD) 331,577) out of which the final model contains on average 690,540 active ones (SD 134,916). The rather high SD is due to the lemma features.

3 Results and Discussion

We present selected comparative evaluation results derived from 10-fold cross-validation experiments.[4] We give the mean tagging accuracy, standard deviation and confidence intervals (CI 95%) derived from a t-test applied to the means of the 10 test folds. The CI expresses that there is a 95% chance that the true accuracy in all representative texts is contained within the computed CI.

3.1 Quantitative Evaluation

Table 6 shows the results of our evaluation. The best system with 93.54% accuracy is our wapiti model using PoS tags, morphological constraints and lemmas evaluated on the external tagset s. We outperform the hunpos baseline based on PoS evidence (87.15%) by 7.3%. Compared to the best hunpos system (88.13%) using PoS and morphology, we get an improvement of 6.1%. As expected, HMM-based tagging cannot make use of complex input tokens that combine lemma, PoS and morphology. However, the CRF model can make use of the lemma evidence resulting in a relative improvement of 1.63% compared to PoS and morphology.

Internal and external tagsets. As mentioned in Section 2.2.2, we expect the internal tagset to have a lower accuracy than the external due to error duplication. Tagset s is consistently slightly better than tagset p (the one more related to Skut and Brants') with the one exception for wapiti using PoS evidence only. The use of an enriched internal tagset proves to be beneficial. For the best system, performance is about 0.5% higher using the internal tagset. The difference is not overwhelming but appears to be very stable across all system combinations.

Upper bound by gold PoS tags and morphology. The lower part of Table 6 shows the effect of providing the correct (gold) PoS tags and morphological information (case and number) from the TüBa as evidence for the statistical tools. Using these results we can estimate the upper bound of the performance if we improve the PoS tagging and provide a better morphological disambiguation. For wapiti and our best feature templates, this is 95.15%, resulting in a maximal relative improvement of 1.72%. For hunpos, the gold information improves by maximally 1.44% for the best evidence (P,M). These rather small numbers show that there is not much room for improvement by optimizing the linguistic enrichment because there will always remain wrong PoS tags and morphological analyses.

Learning curve of internal tagset. In order to check whether more training data could lead to better results, we performed an additional experiment on the first fold using the best wapiti system. Starting with only 10,000 sentences of the TüBa, we obtain 87.08% correctly tagged tokens. Going up to 60,000 sentences, we reach 89.05%. As shown in Figure 3, the learning curve does not yet level off and more data will probably help.

3.2 Qualitative Error Analysis

In order to better understand the error types of the best system, we randomly sampled 10 errors for each of the 7 most frequent error types (see Table 7) from the test set of the first fold. In Table 8, we give a breakdown of the linguistic properties

[3] The actual wapiti code for the feature templates can be downloaded from `http://kitt.cl.uzh.ch/kitt/chunktag/wapiti.txt`.

[4] See Vanwinckelen and Blockeel (2012) for arguments why repeated cross-validation does not lead to better model estimates than simple cross-validation.

Tagger	*Evidence*	*Internal Tagset p/s*				*External Tagset p*				*External Tagset s*			
		Acc.	SD	CI_l	CI_u	Acc.	SD	CI_l	CI_u	Acc.	SD	CI_l	CI_u
wapiti	P,M,L	89.08	0.41	88.79	89.37	93.47	0.32	93.24	93.70	**93.54**	0.33	93.31	93.78
						92.67	0.28	92.46	92.87	93.02	0.31	92.80	93.24
	P,M	86.60	0.39	86.32	86.88	91.92	0.30	91.70	92.13	92.04	0.29	91.84	92.25
						90.95	0.27	90.76	91.14	91.40	0.30	91.18	91.61
	P	84.62	0.45	84.30	84.94	**90.89**	0.30	90.68	91.10	90.74	0.33	90.51	90.98
						89.76	0.35	89.51	90.01	90.43	0.34	90.19	90.67
hunpos	P,M,L	79.15	0.45	78.83	79.47	86.91	0.36	86.65	87.18	87.17	0.37	86.90	87.43
						84.24	0.34	84.00	84.49	87.34	0.39	87.07	87.62
	P,M	80.40	0.50	80.05	80.75	87.89	0.37	87.63	88.16	**88.13**	0.38	87.86	88.40
						85.04	0.39	84.76	85.32	87.63	0.40	87.34	87.91
	P	78.73	0.54	78.34	79.11	86.93	0.39	86.65	87.21	87.15	0.40	86.87	87.43
						83.80	0.37	83.54	84.07	86.61	0.42	86.31	86.91
Using gold PoS (GP) and gold morphology (GM)													
wapiti	GP,GM,L	91.46	0.30	91.24	91.67	95.12	0.24	94.95	95.30	**95.15**	0.24	94.98	95.33
						94.37	0.26	94.19	94.56	94.55	0.25	94.37	94.73
	GP,GM	89.21	0.35	88.96	89.46	93.83	0.26	93.64	94.01	93.87	0.25	93.70	94.05
						92.90	0.25	92.72	93.08	93.07	0.29	92.87	93.28
hunpos	GP,GM,L	80.38	0.40	80.09	80.67	88.10	0.31	87.88	88.33	88.25	0.30	88.04	88.46
						85.73	0.34	85.49	85.98	88.30	0.33	88.06	88.53
	GP,GM	81.94	0.47	81.61	82.28	89.24	0.34	89.00	89.48	**89.40**	0.35	89.15	89.65
						86.19	0.36	85.93	86.45	88.83	0.38	88.56	89.11

Table 6: Evaluation results of 10-fold cross validation experiments. Mean accuracy, standard deviation (SD) and confidence interval 95% (CI_l, CI_u) are reported. The evidence column specifies the type of evidence used for training and testing: P=PoS, M=morphological constraints, L=lemmas. Rows without numbers for the internal tagset indicate experiments where we trained directly on the external tagsets.

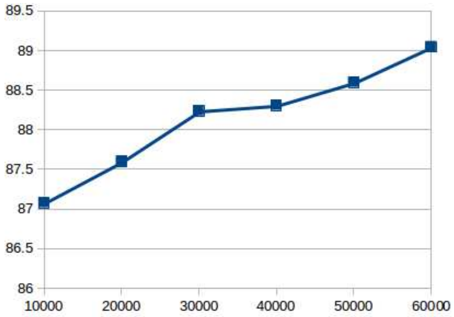


Figure 3: Learning curve of internal tagset

for the two main sources of mistakes, namely attachment errors (47 of 70) and errors in the attachment level (19 of 70). The 4 remaining cases are due to inconsistent tag sequences.

Attachment errors. In 27 cases an attachment is missing, 20 cases have wrong attachments. This error type is mostly related to PPs, followed by NPs, and adjectival phrases (APs). Furthermore, our system often has difficulties with attachment in combination with conjuncts, appositions and comparisons (see Table 8).

Wrong	*Count*	*Error Type*	*Correct*
l/e	730	Attachment	x/e
e/r	648	Attachment	e/x
x/e	626	No attachment	l/e
e/e	551	Attachment one level lower	l/e
e/x	514	No attachment	e/E
e/x	432	No attachment	e/r
x/e	406	Attachment one level lower	x/r
...	...	...	...
All	12,998		

Table 7: Most frequent error types of the internal tagset (from about 500 error types)

	Count	*PP*	*NP*	*AP*
Attachment	**20**	19	1	
thereof ambiguous	4	4		
No attachment	**27**	18	7	2
thereof with conjuncts	8	3	5	
thereof comparisons	2	1	1	
thereof with appositions	1		1	

Table 8: Attachment errors

Errors related to the level of attachment. In these cases, our system attaches a level lower than the gold standard. 12 of 19 cases are shallowly embedded prepositions, most of the time combined with conjuncts and appositions.

Figure 4 shows a case where the material of

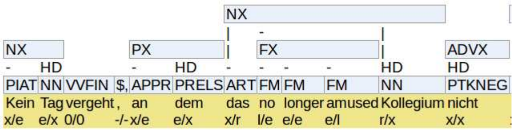

NX
NX PX FX ADVX
- HD - HD - - - - HD HD
PIAT NN VVFIN $, APPR PRELS ART FM FM FM NN PTKNEG
Kein Tag vergeht , an dem das no longer amused Kollegium nicht
x/e e/x 0/0 -/- x/e e/x x/r l/e e/e e/l r/x x/x

Figure 4: Sentence 330 with gold chunk tags: “*Not a day goes by that the no longer amused college does not...*”

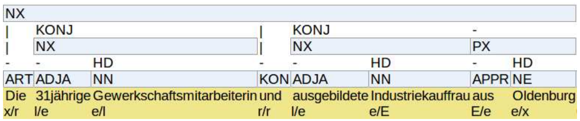

NX
KONJ KONJ -
NX NX PX
- - HD - - HD - HD
ART ADJA NN KON ADJA NN APPR NE
Die 31jährige Gewerkschaftsmitarbeiterin und ausgebildete Industriekauffrau aus Oldenburg
x/r l/e e/l r/r l/e e/E E/e e/x

Figure 5: Sentence 78: “*The 31 year-old union employee and industrial merchant from Oldenburg . . .* ”

the gold standard phrase ’FX’ (‘foreign language material’) was directly embedded in the nominal chunk ’NX’ by our system. It assigned to following chunk tags: *das*_x/e *no*_e/e *longer*_e/e *amused*_e/e *Kollegium*_e/x.

Another possibility is that appositions, conjuncts or APs are not recognized as such and the respective tokens are embedded on the same level as the rest of the chunk.

Another source of errors are conjuncts where some tokens are assigned to another conjunct than annotated in the TüBa. Figure 5 shows a case where the determiner ’Die’ (’the’) is integrated in the first conjunct by our system: *Die*_x/e *31jährige*_e/e *Gewerkschaftsmitarbeiterin*_e/l . . . .

Inconsistent tag sequences. As the bidirectional chunk tags encode the relation to the preceding and the succeeding token, the forward-looking tag part of one token defines the backward-looking part of the following token. Our system assigns inconsistent tag sequences in 4 cases. In all cases this involves punctuation marks inside a chunk. Figure 6 shows the gold standard where our system predicted the following inconsistent (in bold) chunk tags: *der*_x/e *Frankfurter*_e/e *”*_-/- *Guru*_e/**x** *”*_-/- *Berthold*_**E**/e *Kilian*_e/E.

4 Conclusion

With our experiments we have shown that a CRF-based state-of-the-art statistical sequence tagger as wapiti using our hand-crafted feature templates

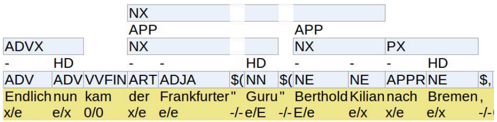

NX
APP APP
ADVX NX NX PX
- HD - - HD - - - HD
ADV ADV VVFIN ART ADJA $(NN $(NE NE APPR NE $,
Endlich nun kam der Frankfurter " Guru " Berthold Kilian nach Bremen ,
x/e e/x 0/0 x/e e/e -/- e/E -/- E/e e/x x/e e/x -/-

Figure 6: Sentence 211: “*Finally the ”Guru” Berthold Kilian from Frankfurt came to Bremen, to...*”

can solve the structural chunk tagging problem for German with an accuracy of 93.5%. For the same task, a classical HMM-based trigram tagger reaches only 88.1% accuracy and is therefore substantially outperformed. Standard HMM tools cannot easily profit from additional evidence such as lemmas. Our results for HMMs are not directly comparable with the reported accuracy of 90.9% of Skut and Brants (1998). Their HMM system additionally includes carefully selected morphological and syntactic information.

Our final feature templates for chunk tagging turned out to be concise and uniformly structured across the evidence layers of PoS, morphological constraints and lemmas. Features and feature combinations from a local context of maximally two tokens to the left and right of the current token turned out to be optimal. Although we tried our best, there might be some unexplored optimizations. However, we would not expect substantial improvements using the same sources of evidence that we experimented with.

The learning curve for the best system suggests that more training material can improve the results even further. More training material could be provided by transforming the TIGER treebank and/or the NEGRA treebank into chunk structures comparable to the ones derived from the TüBa.

The evaluation of the internal tagset with its 50 different tags showed 504 different tag confusions for the test set of our first fold. However, the majority of the occurring errors are attachment errors and most of them are rather unsurprisingly PP attachment errors. Although more training data will probably result in some improvement, a more principled approach for the PP attachment problem seems necessary (see Van Asch and Daelemans (2009)). Within the framework of CRFs an especially crafted evidence column for verb/preposition preferences could be feasible. However, given the progress in efficient dependency parsing we should carefully consider the combination of local evidence – which is typically exploited by approaches as ours – and non-local evidence which is needed for full parsing (see Swift et al. (2004)).

Our experiments with perfect morphology and PoS tags from the TüBa show that better morphological evidence can slightly improve chunk tagging. However, our morphological constraints on case and number for each token realized a lot of the theoretically achievable performance gain. A practical approach of testing the effective gain using currently available resources could be the application of the German rftagger that assigns PoS and morphological tags (Schmid and Laws, 2008).

References

Adrien Barbaresi. 2013. A one-pass valency-oriented chunker for German. In *Human Language Technologies as a Challenge for Computer Science and Linguistics, Proceedings of the 6th Language & Technology Conference, Poznan, Poland*, pages 157–161.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. Tiger: Linguistic interpretation of a german corpus. *Research on Language and Computation*, 2(4):597–620.

Thorsten Brants. 1997. Internal and external tagsets in part-of-speech tagging. In *Proceedings of Eurospeech*, pages 2787–2790.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Applied Natural Language Processing Conference ANLP-2000*, pages 224–231.

Péter Halácsy, András Kornai, and Csaba Oravecz. 2007. Hunpos: an open source trigram tagger. In *Proceedings of the 45th Annual Meeting of the ACL on Interactive Poster and Demonstration Sessions*, ACL '07, pages 209–212. Association for Computational Linguistics.

Erhard Hinrichs. 2005. Finite-state parsing of German. In *Inquiries into Words, Constraints, and Contexts*, pages 35–44. CSLI Publications.

Hannah Kermes and Stefan Evert. 2002. Yac - a recursive chunker for unrestricted german text. In *LREC*.

Kimmo Koskeniemmi and Mariikka Haapalainen, 1996. *GERTWOL - Lingsoft Oy*, pages 121–140. Number 34 in Sprache und Information. Niemeyer Max Verlag GmbH.

Sandra Kübler, Kathrin Beck, Erhard Hinrichs, and Heike Telljohann. 2010. Chunking german: an unsolved problem. In *Proceedings of the Fourth Linguistic Annotation Workshop*, LAW IV '10, pages 147–151, Stroudsburg, PA, USA. Association for Computational Linguistics.

Thomas Lavergne, Olivier Cappé, and François Yvon. 2010. Practical very large scale CRFs. In *Proceedings the 48th Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 504–513. Association for Computational Linguistics, July.

Ryan McDonald, J Nivre, Y Quirmbach-Brundage, Y Goldberg, D Das, K Ganchev, K Hall, S Petrov, H Zhang, O Täckström, C Bedini, N Bertomeu Castelló, and J Lee. 2013. Universal dependency annotation for multilingual parsing. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 92–97, Sofia, Bulgaria, August. Association for Computational Linguistics.

Frank Henrik Müller. 2007. *A finite-state approach to shallow parsing and grammatical functions annotation of German*. Ph.D. thesis.

L. A. Ramshaw and M. P. Marcus. 1995. Text chunking using transformation-based learning. In *Proceedings of the Third Annual Workshop on Very Large Corpora*, pages 82–94. ACL.

Helmut Schmid and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (Coling 2008)*, pages 777–784, Manchester, UK, August.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.

Fei Sha and Fernando Pereira. 2003. Shallow parsing with conditional random fields. In *NAACL '03: Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, pages 134–141.

Wojciech Skut and Thorsten Brants. 1998. Chunk tagger – statistical recognition of noun phrases. In *Proceedings of the ESSLLI Workshop on Automated Acquisition of Syntax and Parsing*, Saarbrücken, Germany.

Wojech Skut, Brigitte Krenn, Thorsten Brants, and Hans Uszkoreit. 1997. An annotation scheme for free word order languages. In *Proceedings of the Fifth Conference on Applied Natural Language Processing (ANLP)*, pages 88–95, Washington, D.C.

Charles A. Sutton and Andrew McCallum. 2012. An introduction to conditional random fields. *Foundations and Trends in Machine Learning*, 4(4):267–373.

Mary Swift, James Allen, and Daniel Gildea. 2004. Skeletons in the parser : Using a shallow parser to improve deep parsing. In *COLING'04*.

H. Telljohann, E. Hinrichs, S. Kübler, et al. 2004. The TüBa-D/Z treebank: Annotating German with a context-free backbone. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC 2004)*, pages 2229–2235.

Erik F. Tjong Kim Sang and Sabine Buchholz. 2000. Introduction to the CoNLL-2000 shared task: Chunking. In *Proceedings of CoNLL-2000 and LLL-2000*, pages 127–132. Lisbon, Portugal.

Vincent Van Asch and Walter Daelemans. 2009. Prepositional phrase attachment in shallow parsing. In *Proceedings of the International Conference RANLP-2009*, pages 12–17, Borovets, Bulgaria, September.

Gitte Vanwinckelen and Hendrik Blockeel. 2012. On estimating model accuracy with repeated cross-validation. In *BeneLearn 2012: Proceedings of the 21st Belgian-Dutch Conference on Machine Learning, Ghent, 24-25 May 2012*, pages 39–44.

Enforcing Consistent Translation of German Compound Coreferences

Laura Mascarell, Mark Fishel, Natalia Korchagina and **Martin Volk**
Institute of Computational Linguistics
University of Zurich
Switzerland
{mascarell,fishel,korchagina,volk}@cl.uzh.ch

Abstract

Coreferences to a German compound (e.g. *Nordwand*) can be made using its last constituent (e.g. *Wand*). Intuitively, both coreferences and the last constituent of the compound should share the same translation. However, since Statistical Machine Translation (SMT) systems translate at sentence level, they both may be translated inconsistently across the document. Several studies focus on document level consistency, but mostly in general terms. This paper presents a method to enforce consistency in this particular case. Using two in-domain phrase-based SMT systems, we analyse the effects of compound coreference translation consistency on translation quality and readability of documents. Experimental results show that our method improves correctness and consistency of those coreferences as well as document readability.[1]

1 Introduction

Statistical Machine Translation (SMT) systems translate sentences as isolated units, ignoring document-level information (Koehn, 2009). Since this document unawareness negatively impacts the translation quality, many approaches have been proposed to introduce discourse level features in SMT.

Specifically, the issue of consistent lexical choice is our focus of attention. The one-sense-per-discourse hypothesis (Gale et al., 1992) and later the one-translation-per-discourse applied to machine translation (Carpuat, 2009) show that consistency in discourse is desirable. Some methods were then proposed to enforce consistency by applying caching (Tiedemann, 2010; Gong et al., 2011). Later, Carpuat and Simard (2012) showed that SMT systems already translate consistently, but consistency is not a good indicator of translation quality. Translation systems trained on large text collections deal with more translation choices and, therefore, they translate more inconsistently. The same study also proved that inconsistencies signal translation errors more often than consistences do. Repetition as a consequence of strict consistency enforcement is also discussed, since it is difficult to determine whether repetition is desirable or not (Carpuat and Simard, 2012). On the one hand, human translators tend to use repetition across the document. On the other hand, it may negatively affect fluency (Guillou, 2013).

Whilst all these analyses focus on a general application of consistency, in this paper, we address consistency on coreferences to a compound at document-level. We tackle a specific case in which the compound is coreferenced using its last part, proposing a method to enforce the consistent translation of those coreferences. We focus on German on the source side, since it is a language rich in compounds. For instance, considering the German-French language pair, the compound *Ostwand* (“east face” or “east wall”) in the mountaineering domain is coreferenced as *die Wand* (“the wall”). While the best French translation candidate of *Ostwand* is *face nord*, *Wand* as an isolated word is more likely to be translated as *paroi*. Here we assume that the last part of a compound and its coreferences should share the translation, which would help to identify connectedness between sentences. In our experiments, we assess consistency and its correlation with translation quality in this particular case. Although it is not clear that repetition is always desirable, does it improve readability?

In the following, section 2 gives an overview of the work related to consistency in SMT. After describing our method to enforce consistency in section 3, we detail the carried out experiments in section 4 and discuss the results in section 5.

2 Related Work

The well-known one-sense-per-discourse hypothesis (Gale et al., 1992) was later applied to machine translation as one-translation-per-discourse (Carpuat, 2009), proving that more than one translation per discourse is often due to wrong lexical choices. Based on this constraint, some studies focused on analysing consistency in SMT. Carpuat and Simard (2012) analysed how consistent is the output of SMT systems compared to human translations. They experimented with several phrase-based SMT systems trained on different conditions such as data size, domain or language pair. The work showed that SMT systems translate nearly as consistently as human translators. However, inconsistency often points to translation errors and therefore cannot be ignored. Guillou (2013) studied a different approach analysing *when* (i.e. genre) and *where* (i.e. part-of-speech) lexical consistency is desirable.

Several approaches focused on enforcing consistent lexical choice. Tiedemann (2010) proposed a cache-based model that propagates the translation of phrases across the document. However, the caching approach is sensitive to error propagation. Gong et al. (2011) extended the approach applying a dynamic, static and topic cache, where the latest keeps the error propagation problem controlled. Xiao et al. (2011) described a three steps procedure that enforces consistent translation of ambiguous words and Ture et al. (2012) introduced cross-sentence features to the translation model, achieving improvements on the Arabic-English language pair.

3 Enforcing Consistent Translation

A compound can be coreferenced by its last constituent. For example, the compound *Nordwand* ("north face"), formed by *Nord* (*X*) and *Wand* (*Y*) can be coreferenced by *Wand* alone (*Y*)[2]. The main aim of our method is to detect such cases and to enforce that last constituent *Y* to have the same translation in both *XY* and *Y*.

To consistently translate *Y*, we cache its translation and enforce it when a coreference is detected. This greedy approach is sensitive to error propagation in general; however, our method is restricted to compounds, which provide more context for a correct translation than single roots, yielding less translation variants.

In detail our method works as follows. We translate each sentence individually, caching the translation of the last part of compounds and enforcing a translation for a coreference when required. To identify compounds, first we analyse each noun with the German morphology system Gertwol (Koskeniemmi and Haapalainen, 1994), which marks the boundaries between independent morphemes (e.g. the analysis of *Ostwand* is *Ost#wand*). Next, we obtain the translation of a compound from the word alignment given by the SMT decoder. We then check at the phrase table which one of its content words is the translation of *Y*, and we cache it. For instance, considering the German compound *Bundesamt* ("federal office"), which is aligned to the French *le office fédéral* in the target side, and its coreference *Amt* ("office"), we cache the pair *Amt* and *office*. If there are several compounds sharing the last morpheme, *Y* usually will corefer to the closer one, but not necessarily, which intoduces an ambiguity problem. For instance, the noun *Wand* ("wall") in *Ostwand* ("east face") and *Felswand* ("rockface") is translated into French as *face* and *paroi*, respectively. An analysis of local context would provide better precision, but for the sake of simplicity, we assume that *Y* corefers to the last compound translated, always caching its last occurrence.

To identify *Y* as a coreference, we apply the pattern "determiner + (adjective) + *Y* lemma",

[2] Compounds can consist of more than two roots and thus also *X* and *Y*. For instance, considering the compound *Eigernordwand* ("Eiger north face"), *Y* can be either *Wand* or the compound *Nordwand*.

where the *adjective* is optional and the *determiner* is tagged as one of the following parts-of-speech: PDS (substituting demonstrative pronoun), PDAT (attributive demonstrative pronoun), PPOSS (substituting possessive pronoun), PPOSAT (attributive possessive pronoun) or ART (restricted to definite articles). Thus, *die prächtige Fahrt* and *diesen Grat* are examples matching the pattern. We use the lemma of *Y* to also match examples where German cases (e.g. genitive and dative) change the form of *Y* (e.g. *Grates* is the genitive form of *Grat*). We then check that *Y* is cached and there is a compound *XY* in the four preceding sentences[3]. In case of PDAT (e.g. *diese*), we consider the whole document. PDAT is a strong coreference indicator, and we found examples having more than four sentences between the compound and its coreference. The cached translation of *Y* is then plugged into the decoder.

4 Experiments

We first conduct an analysis of compounds and coreferences automatically detected, which is mostly manually addressed by two different annotators. We then carry out all the experiments on the German-French language pair, testing different approaches to plug the translation into the decoder and to increase the coverage of our method.

The data comes from the Text+Berg corpus (Bubenhofer et al., 2013), a collection of documents from the Alpine domain, which was built as a result of digitising and processing the Swiss Alpine Club yearbooks from 1864 to 2009 (Volk et al., 2010). The sentence alignment was carried out with Bleualign (Sennrich and Volk, 2010). The test set in both manual analysis and translation task is a collection of 318 examples, that is, groups of sentences containing a compound noun and its coreferences, randomly sampled from Text+Berg data (see section 4.1).

4.1 Analysing the detected compounds and their coreferences

To evaluate how often a German compound is coreferenced using its last constituent, we automatically detect them in a German corpus consisting of roughly 1.1 million sentences from Text+Berg. Our method is the same described in section 3 to identify compounds and their coreference. We found 24,317 cases where this occurs, and to assess the effectiveness of our method at detecting a compound and its coreferences, we carried out a manual analysis on a random sample containing 318 compound-coreference pairs automatically detected. This analysis shows that 107 of these pairs are false positives, that is, the coreference is incorrectly detected, often due to the lexicalization of the compound or a number disagreement between compound and coreference. A lexicalized compound cannot be coreferenced by its last part, since its translation does not correspond to the translation of its constituents. For example, the German compounds *Zusammenarbeit* ("cooperation") and *Augenblick* ("moment") are lexicalized and thus, they cannot be corefereced by *die Arbeit* ("the work") or *der Blick* ("the view"), respectively. The disagreement in number is due to our method match the lemma of *Y* to also detect examples when the German cases change the word forms. In other less frequent cases, the detected coreference has nothing to do with the compound. For instance, in the example shown in Table 1, the pattern matches correctly the coreference *Gipfel* ("summit"), but the method fails at detecting *Schneegipfel* ("snowy summit") as the compound coreferenced. Indeed, *Gipfel* ("summit") corefers to the mountain *Königsspitze*.

The manual analysis also focuses on the correct detections, distinguishing the following most common patterns:

- The coreference is preceded by a definite article + adjective or by the demonstrative adjectives *dieser* ("this") and *jener* ("that") in all their grammatical forms.

- The compound is in genitive case and its coreference in nominative or dative case. For example, *das Tal* ("the valley") corefers to *Haupttals* ("main valley") in *Sohle des Haupttals* ("bottom of the main valley").

4.2 Enforcing translation

The translation of compounds is the first step to proceed with our method. However, compounds are often Out-Of-Vocabulary (OOV) (i.e. they do

[3] We carried out several experiments with different number of sentences. We decided to use a four sentence window, since more than four introduces too noise.

Er sah von ihr wirklich auf den obern Trafoierferner links hinunter und erblickte über mehrere ***Schneegipfel*** hinweg sein Ziel, die im Hintergrunde sich erhebende Königsspitze .
Auf deren ***Gipfel*** grub er sich dann halbliegend in den zusammengewehten Schnee ein .
“He looked from her to the upper Trafoierferner down to the left and saw several *snow peaks* across his goal, which is in the background of *Königsspitze*.”
“On its *summit* he dug himself in a half-lying in the snow along a wind-blown.”

Table 1: Example where the coreference to a compound was incorrectly detected.

not appear in the training corpus) and the system cannot translate them. These compounds are usually composed of frequent words in the training corpus, so we can obtain the translation of an unseen compound by splitting it into its known parts and translating them (Koehn and Knight, 2003).

We want to assess the performance of our method in both approaches (i.e. splitting compounds and not splitting them), so we build two phrase-based SMT systems *SMT-1* and *SMT-split*, where *SMT-split* performs compound splitting. Both systems are built using the standard settings (Koehn et al., 2003), 5-gram language model KenLM (Heafield, 2011) and GIZA++ (Och and Ney, 2003). The language model is trained on a total of 624,160 sentences (13 million target tokens) and the training set consists of 219,187 sentences and roughly 4.1/4.7 million words in German and French, respectively. The SMT systems are tuned with Minimum Error Rate Training (Och, 2003) on a development set, also from Text+Berg, consisting of 1,424 sentence pairs and approximately 31,000 tokens for each language. We expect to enforce a consistent translation in a higher number of cases with the *SMT-split* system. Furthermore, the splitting method allows us to have a one-to-one alignment between the compound constituents and their translation. Thus, we can identify the translation of the last part of the compound and cache it directly.

Once a compound *XY* is translated, and in order to enforce the correct translation of *Y*, we explore two approaches. The idea is to find out which is the best at selecting the translation candidate. The first one lets the decoder decide which is the best translation of *Y*. We first cache the translation of a compound *XY* as a translation of *Y*, and when a coreference *Y* is detected, we plug all the content words cached into the decoder, not assigning any probability to them, so by default they are 1. Then, the decoder chooses the best candidate based on translation and language model scores. Interestingly, this first approach fails in our experiments. Most of the time, the decoder takes the translation of the first constituent of the compound instead of the last one. For instance, *Wand* as a coreference of *Nordwand* (French translation: *nord face*) is enforced to be translated into *nord*. We think that if the first constituent of a compound appears more frequently in the language model, the score computed is then higher and it is then picked as the translation candidate.

In the second approach, for each content word of a compound translation, we check that it appears as a translation candidate of *Y* in the phrase table. We then cache only the one that has the highest direct phrase translation probability. We observe that by applying this method, some examples where the compound was aligned to one word in the target side due to a misalignment or lexicalization of the compound are improved. In the first approach, we consider that these examples enforce an incorrect translation to the coreference, so they are detected as false positives and discarded. However, in this second approach, the translation is enforced, since it appears as a translation candidate of *Y* in the phrase table, resulting in a better translation of the term according to the context. The second example in the Table 2 shows that the term *Fahrt* is translated into *ascension*, which is also the translation of the compound coreferenced (*Bergfahrten*).

The results in section 5 are obtained with the second approach. Moreover, we use the automatic generator from the Apertium[4] MT toolbox to generate the correct form of those cases where compound and coreference do not agree in number.

[4] `www.apertium.org`

Source	Die Originalauswertung wurde in den Zwischenmassstab 1:20000 reduziert, worauf das ***Bundesamt*** (trans: *office fédéral*) für Landestopographie in Aktion trat. Nur dieses ***Amt*** war in der Lage, [...]
English translation by the authors	“The original evaluation was reduced in the intermediate scale 1:20000, followed by the *Federal Office* of Topography went into action. Only this *office* was able to [...] ”
SMT-1, SMT-split	que ce ***poste*** tait dans la situation, [...]
SMT-1 enf., SMT-split enf.	que de cet ***office*** tait en mesure [...]
Source	Unter den Neuen ***Bergfahrten*** (trans: *ascension*) in den Schweizeralpen ist im IV. Band der Alpen 1928 eine erste Begehung des ganzen Südostgrates von der Gemsenlücke [...] über die prächtige ***Fahrt*** geblieben.
English translation by the authors	“Among the new *hill climbing* in the Swiss Alps is mentioned in the fourth volume of the Alps 1928 a first ascent of the whole South East ridge of the Gemsenlück [...] remained about the magnificent *journey*.”
SMT-1, SMT-split	par cette magnifique ***course***.
SMT-1 enf., SMT-split enf.	par cette magnifique ***ascension***.
Source	Einen Teil ihrer bergsteigerischen und wissenschaftlichen Erfolge finden unsere Mitglieder in diesem ***Quartalsheft*** (trans: *présent numéro trimestriel*) verzeichnet. Das vorliegende ***Heft*** möge daher [...]
English translation by the authors	“Our members find part reported of their mountaineering and scientific achievements in this *quarterly bulletin*. This *bulletin* may therefore [...]
SMT-1, SMT-split, SMT-1 enf.	le ***cahier*** möge donc [...]
SMT-split enf.	le présent ***numéro*** möge donc [...]
Source	Dass dies gemacht wird, zeigt das Routenbuch Clean-Begehungen, das im ***Klettergebiet*** (trans: *site d’escalade*) liegt. Wir diskutieren über die schönsten Routen im ***Gebiet***.
English translation by the authors	“That this is done, the route book shows Clean inspections, which is located in the *climbing area*. We discuss about the best tours in the *area*.”
SMT-1, SMT-split	nous discutons sur les plus belles voies dans la ***région***.
SMT-1 enf., SMT-split enf.	nous discutons sur les plus belles voies du ***site***.

Table 2: Examples where our enforcing method improves the translation of the coreference. The first example shows that the enforcing method improves the translation of *Amt*. In the second example, the compound is aligned to only one word in the target side, but its coreference translation is correctly enforced and improved. In the third example, SMT-1 misaligned *Quartalsheft* to only *trimestrel*, thus the coreference is not enforced. Due to the compound splitting technique, there is one-to-one correspondence between *sheft* and *numéro*, then the coreference translation is successfully enforced. In the last example, both translations of *Gebiet* are correct, but *site* is consistent with the translation of the compound coreferenced.

5 Results

We present results on both correctness and consistency. The analysed systems are *SMT-1* and *SMT-split* with and without applying our enforcing method. The experiments are performed on the test set consisting of 211 compound-coreference pairs correctly detected. To get those results, two annotators conducted a manual analysis and annotation of the results. The agreement between them at the task of deciding "is/is not a coreference" and "is correct/wrong coreference translation" is 73.4% and 86.8%, respectively.

Automatic detection of coreferences to a compound: The precision of our method correctly detecting a coreference to a compound is 66.4% (i.e. 211 out of 318 coreferences). We only analyse the sentences detected by the method, so recall is not computed. However, our detector's approach is broad-coverage-oriented, that is, it tends to detect more false positives examples while practically avoiding false negatives.

Coverage of the method: We compute statistics on the examples where a translation is enforced in both correct and incorrect detection of compound and coreference. When we do not perfom splitting, 42.2% (i.e. 89 out of 211) of the positive examples and 27.1% (i.e. 29 out of 107) of the incorrectly detected are enforced. The remaining 57.8% of the positive examples (i.e. where no enforcing is applied) is due to OOV compounds and misalignments. Splitting significantly increases the coverage of enforced translations from 42.2% to 56.4% (i.e. 119 out of 211). The incorrectly identified coreferences have again a lower impact ratio (34.6%; 37 out of 107).

Consistency and correctness of *SMT-1*: The *SMT-1* system without enforcing translates correctly with 80.1% accuracy and 27.5% consistency (see Table 3). The German noun *Wand* is the most common example of inconsistent but correct translation in our test set. The most likely translation for this noun is *paroi* in the Text+Berg corpora. However, when *Wand* is part of a compound, it is usually translated into *face*.

Our method applied to *SMT-1* enforces a consistent translation in 89 of the cases improving the translation of six of them and 15 cases stay correct, but become consistent. For instance, at the last example in Table 2, the noun *Gebiet* ("area") is translated into *site* instead of *région* when a consistent translation is enforced, yet both translations are correct. Furthermore, a coreference to a compound stays incorrect but become consistent, increasing the value of incorrect and consistent by one (see Table 4). The remaining 67 stay unmodified, that is, *SMT-1* chooses the consistent translation for the coreference without enforcing. Thus, while the correctness is slightly raised from 80.1% to 82.9%, the consistency improves from 27.5 to 37.9%.

	Consistent:	
	yes	no
Correct	52	117
Incorrect	6	36

Table 3: Consistency and correctness results of the *SMT-1* system without enforcing consistency.

	Consistent:	
	yes	no
Correct	73	102
Incorrect	7	29

Table 4: Consistency and correctness results of the *SMT-1* system when our enforcing method is applied.

Consistency and correctness of *SMT-split*: When we perform splitting, three cases become worse, but most of the cases that are not enforced with the *SMT-1* system due to a misalignment or OOV compounds, are now enforced and improved. For instance, the third example in Table 2 shows that the translation of the German noun *Heft* is only well enforced with the splitting approach, since without splitting, the compound *Quartalsheft* is misaligned to only *trimestrel*. The *SMT-split* system without enforcing translates correctly with 82.0% accuracy and 35.1% consistency (see Table 5).

	Consistent:	
	yes	no
Correct	68	105
Incorrect	6	32

Table 5: Consistency and correctness results of the *SMT-split* system without enforcing consistency.

When we apply enforcing to the *SMT-split* system, the coverage is increased and more improvement is shown (Table 6). Indeed, it applies enforcing to 109 cases improving 10 of them. Although there are six consistent and incorrect cases in both Table 3 and Table 6, some of them are different. Specifically, *SMT-split* improves two of them and makes consistent another two, although both stay incorrect. The correctness rises from 82.0% to 86.7% and consistency from 35.1% to 52.1%.

	Consistent:	
	yes	no
Correct	103	80
Incorrect	6	22

Table 6: Consistency and correctness results of the *SMT-split* system when our method is applied.

	Correctness	Consistency
SMT-1	80.1%	27.5%
SMT-split	82.0%	35.1%
SMT-1 enf.	82.9%	37.9%
SMT-split enf.	86.7%	52.1%

Table 7: Overall percentages of consistency and correctness results of *SMT-1* and *SMT-split* systems, with and without applying our enforcing method.

Overall, the final effect is positive (see Table 7). Correctness rises from 80.1% to 86.7%, improving 17 examples, that is, one third of errors are fixed, and consistency from 27.5% to 52.1%.

6 Conclusions

We present a method to enforce consistent translation of coreferences to a compound, when the coreference matches with the last constituent of the compound coreferenced. We assess correctness and consistency with two systems *SMT-1* and *SMT-split*, where the latest performs compound splitting. We then evaluate how well our method performs when applied in both systems.

We also conduct a manual analysis on the source side. We detect that the demonstrative adjectives *dieser* ("this") and *jener* ("that") are strong indicators of coreference. Furthermore, compounds are often in genitive case and their coreferences in either nominative or dative. The incorrect detection of a coreference to a compound are often due to a lexicalization of the compound and number disagreement between compound and coreference. Note that we match lemmas to abstract away from morphological changes due to the German cases (e.g. genitive or dative).

Experimental results show that the Statistical Machine Translation (SMT) systems often translate correctly and consistently coreferences to a compound. However, when our method is applied, some cases are improved and there are only few cases where the translation become worse. When the translation is successfully improved, it usually enforces a more specific term in the context. Since the splitting method allows the *SMT-split* system to translate out-of-vocabulary compounds, *SMT-split* increases the number of the enforced examples, improving the translation in a higher number of cases. Finally, we point out the importance of consistency in this study. At the examples where the coreference is correct, but inconsistent, our method also enforces consistency, which helps the reader to identify connectedness between coreference and compound, improving the readability of the document.

7 Future Work

We want to extend the study testing our method with an out-of-domain system. We expect that compounds will be correctly translated, but not their coreferences. Then, our method would enforce a correct translation, improving the output of the machine translation system.

Another case of study is when the compound is coreferenced using its first constituent rather than its last. The following made-up example shows that *triples* in the phrase *the identified triples* is a coreference of *triple structures*. Note that *triple* has been nominalized in the coreference.

> *[...] to identify* ***triple structures*** *[...]*
> ***The identified triples*** *[...]*

Since the first constituent of a compound is often used to describe the rest of it, we want to analyse whether the compound could be coreferenced by the nominalization of its first part.

We detected also cases where the coreference is not the last part of the compound coreferenced, but a synonym instead. For example, *Nordwand* ("north face" or "north wall") can be also corefered by *dieser Mauer* ("this wall"). We want to assess how consistency impacts on these examples, where the source is already inconsistent.

Acknowledgments

The authors would like to thank Bonnie Webber for pointing out coreferenced noun compounds as a problem for machine translation. This research was supported by the Swiss National Science Foundation under grant CRSII2_147653/1 through the project "MODERN: Modelling discourse entities and relations for coherent machine translation".

References

Noah Bubenhofer, Martin Volk, David Klaper, Manuela Weibel, and Daniel West. 2013. Text+Berg-korpus (release 147_v03). Digitale Edition des Jahrbuch des SAC 1864-1923, Echo des Alpes 1872-1924 und Die Alpen 1925-2011.

Marine Carpuat and Michel Simard. 2012. The trouble with SMT consistency. In *Proceedings of the Seventh Workshop on Statistical Machine Translation*, pages 442–449, Montréal, Canada.

Marine Carpuat. 2009. One translation per discourse. In *Proceedings of the Workshop on Semantic Evaluations: Recent Achievements and Future Directions*, pages 19–27, Boulder, Colorado.

William A Gale, Kenneth W Church, and David Yarowsky. 1992. One sense per discourse. In *Proceedings of the Workshop on Speech and Natural Language*, pages 233–237, Harriman, New York.

Zhengxian Gong, Min Zhang, and Guodong Zhou. 2011. Cache-based document-level Statistical Machine Translation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 909–919, Edinburgh, UK.

Liane Guillou. 2013. Analysing lexical consistency in translation. In *Proceedings of the Workshop on Discourse in Machine Translation*, pages 10–18, Sofia, Bulgaria.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, pages 187–197, Edinburgh, Scotland.

Philipp Koehn and Kevin Knight. 2003. Empirical methods for compound splitting. In *Proceedings of the tenth conference on European chapter of the Association for Computational Linguistics*, volume 1, pages 187–193, Budapest, Hungary.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1, pages 48–54, Edmonton, Canada.

Philipp Koehn. 2009. *Statistical Machine Translation*. Cambridge University Press, New York, USA.

Kimmo Koskeniemmi and Mariikka Haapalainen. 1994. Gertwol–lingsoft oy. *Linguistische Verifikation: Dokumentation zur Ersten Morpholympics*, pages 121–140.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics*, 29(1):19–51.

Franz Josef Och. 2003. Minimum error rate training in Statistical Machine Translation. In *Proceedings of the 41st Annual Meeting on Association for Computational Linguistics*, volume 1, pages 160–167, Sapporo, Japan.

Rico Sennrich and Martin Volk. 2010. MT-based sentence alignment for OCR-generated parallel texts. In *The Ninth Conference of the Association for Machine Translation in the Americas*, Denver, Colorado.

Jörg Tiedemann. 2010. Context adaptation in Statistical Machine Translation using models with exponentially decaying cache. In *Proceedings of the 2010 Workshop on Domain Adaptation for Natural Language Processing*, pages 8–15, Uppsala, Sweden.

Ferhan Ture, Douglas W. Oard, and Philip Resnik. 2012. Encouraging consistent translation choices. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics. Human Language Technologies*, pages 417–426, Montréal, Canada.

Martin Volk, Noah Bubenhofer, Adrian Althaus, Maya Bangerter, Lenz Furrer, and Beni Ruef. 2010. Challenges in building a multilingual alpine heritage corpus. In *Proceedings of the Seventh International Conference on Language Resources and Evaluation*, Valletta, Malta, may.

Tong Xiao, Jingbo Zhu, Shujie Yao, and Hao Zhang. 2011. Document-level consistency verification in Machine Translation. In *Proceedings of the 13th Machine Translation Summit*, volume 13, pages 131–138, Xiamen, China.

Knowledge Discovery in Scientific Literature [*]

Jinseok Nam [1,2], Christian Kirschner [1,2], Zheng Ma [1,2], Nicolai Erbs [1], Susanne Neumann [1,2]
Daniela Oelke [1,2], Steffen Remus [1,2], Chris Biemann [1], Judith Eckle-Kohler [1,2]
Johannes Fürnkranz [1], Iryna Gurevych [1,2], Marc Rittberger [2], Karsten Weihe [1]
[1] Department of Computer Science, Technische Universität Darmstadt, Germany
[2] German Institute for Educational Research, Germany
`http://www.kdsl.tu-darmstadt.de`

Abstract

Digital libraries allow us to organize a vast amount of publications in a structured way and to extract information of user's interest. In order to support customized use of digital libraries, we develop novel methods and techniques in the Knowledge Discovery in Scientific Literature (KDSL) research program of our graduate school. It comprises several sub-projects to handle specific problems in their own fields. The sub-projects are tightly connected by sharing expertise to arrive at an integrated system. To make consistent progress towards enriching digital libraries to aid users by automatic search and analysis engines, all methods developed in the program are applied to the same set of freely available scientific articles.

1 Introduction

Digital libraries in educational research play a role in providing scientific articles available in digital formats. This allows us to organize a vast amount of publications, and the information contained therein, in a structured way and to extract interesting information from them. Thus, they support a community of practices of researchers, practitioners, and policy-makers. In order to support diverse activities, digital libraries are required to provide effective search, analysis, and exploration systems with respect to specific subjects as well as additional information in the form of metadata.

Our analysis is mainly focused on the educational research domain. The intrinsic challenge of knowledge discovery in educational literature is determined by the nature of social science, where the information is mainly conveyed in textual, i.e., unstructured form. The heterogeneity of data and lack of metadata in a database make building digital libraries even harder in practice. Moreover, the type of knowledge to be discovered that is valuable as well as obtainable is also hard to define. As this type of work requires considerable human effort, we aim to support human by building automated processing systems that can provide different aspects of information, which are extracted from unstructured texts .

The rest of this paper is organized as follows. In Section 2, we introduce the Knowledge Discovery in Scientific Literature (KDSL) program which emphasizes developing methods to support customized use of digital libraries in educational research contexts. Section 3 describes the sub-projects and their first results in the KDSL program. Together, the sub-projects constitute an integrated system that opens up new perspectives for digital libraries. Section 4 finally concludes this paper.

2 Knowledge Discovery in Scientific Literature

In the age of information overload, even research professionals have difficulties in efficiently acquiring information, not to mention the public.

An accessible, understandable information supply of educational research will benefit not only the academic community but also the teachers, policy makers and general public.

There are several related research projects. The CORE (Knoth and Zdrahal, 2012) project aims to develop a system capable of seamless linking of existing repositories of open access scientific papers. The CODE project developed a platform which facilitates exploration and analysis in research areas using open linked data.[1]

In contrast to general-purpose systems for managing scientific literature, we aim at building a system in specific domains including, but not limited to, the educational research where, for instance, users are allowed to navigate visually a map of research trends or are provided with related works which use the same datasets.

2.1 Structure of KDSL

The KDSL program is conducted under close collaboration of the Information Center for Education (IZB) of the German Institute for International Educational Research (DIPF) and the Computer Science Department of TU Darmstadt. IZB provides modern information infrastructures for educational research. It coordinates the German Education Server and the German Education Index (*FIS Bildung Literaturdatenbank*).[2]

Consisting of several related sub-projects, the KDSL program focuses on text mining, semantic analysis, and research monitoring, using methods from statistical semantics, data mining, information retrieval, and information extraction.

2.2 Data

All of our projects build up on the same type of data which consists of scientific publications from the educational domain. However, the publications differ from each other in their research approach (e.g., empirical/theoretical and qualitative/quantitative), in their topics and in their target audience / format (e.g., dissertations, short/long papers, journal articles, reviews). This leads to a vast heterogeneity of content which also follows from the broad range of disciplines involved in the educational research (for example psychology, sociology and philosophy).

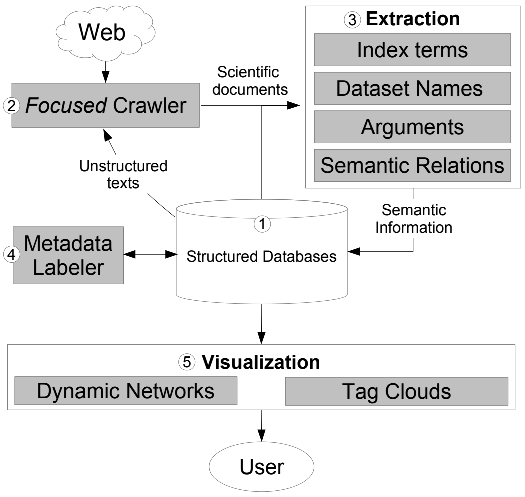


Figure 1: Links between sub-projects in KDSL for educational research

At DIPF, there are mainly two databases containing relevant publications for our projects: *pedocs* and *FIS Bildung*. *FIS Bildung* (Carstens et al., 2011) provides references to scientific articles collected from more than 30 institutions in all areas of education. Specifically, the database consists of over 800,000 entries and more than a half of them are journal articles in German. One-third of the references to articles published recently has full-text in a pdf format.[3] *pedocs* (Bambey and Gebert, 2010), a subset of *FIS Bildung*, maintains a collection of open-access publications and makes them freely accessible to the public as a long-term storage of documents. As of today, the total number of documents in *pedocs* is about 6,000.[4] Each entry in both databases is described by metadata such as title, author(s), keywords and abstract.

2.3 Vision and Challenges

The overall target of KDSL is to structure publications automatically by assigning metadata (e.g., index terms), extracting dataset names, identifying argumentative structures and so on. Therefore, our program works towards providing new

[1] http://code-research.eu

[2] http://www.fachportal-paedagogik.de

[3] Detailed statistics can be found at http://dipf.de/de/forschung/abteilungen/pdf/diagramme-zur-fis-bildung-literaturdatenbank

[4] April 2014

methods to identify and present the information searched by a user with reduced effort, and to structure the information regarding the specific needs of the users in searching the mentioned databases.

Figure 1 shows how the sub-projects interact with each other to achieve our goal. Each subproject in KDSL acts as a building block of the targeted system, i.e., an automated processing system to help educational researchers. Getting more data, even unlabeled (or unannotated), is one of the key factors which lead to more accurate machine learning models. The focused crawler collects documents from websites in educational contexts (block ② in Fig. 1). Other sub-projects can benefit from a large corpus of the crawled documents that might provide more stable statistics in making predictions on unseen data. By using structured databases and the crawled documents, we perform several extraction tasks (block ③), such as identifying index terms (Sec. 3.2, 3.5), dataset names (Sec. 3.3), argumentative structures (Sec. 3.4), and semantic relations between entities (Sec. 3.1). Towards the enrichment of databases, we investigate methods to assign the extracted information in structured formats, i.e., metadata (block ④). In turn, we also aim at providing novel ways to visualize the search results and thus to improve the users' search experience (block ⑤), for instance through displaying dynamics of index terms over time (Sec. 3.6) and tag clouds (Sec. 3.7).

3 Projects

In the following sections, we describe subprojects in KDSL with regards to their problems, approaches, and the first results.

3.1 Crawling and Semantic Structuring

A vital component of the semantic structuring part of this project is the process of reliably identifying relations between arbitrary nouns and noun phrases in text. In order to achieve high-quality results, a large in-domain corpus is required.

Task The corpus necessary for unsupervised relation extraction is created by enlarging the existing *pedocs* corpus (cf. Sec. 2.2) with documents from the web that are of the same kind. The project's contribution is thus twofold: *a*) focused crawling, and *b*) unsupervised relation extraction.

Dataset Plain texts extracted from *pedocs* pdfs define the domain of the initial language model for a focused crawler (Remus, 2014).

Approaches The *Distributional Hypothesis* (Harris, 1954), which states that similar words tend to occur in similar contexts, is the foundation of many tasks including relation extraction (Lin and Pantel, 2001). Davidov et al. (2007) performed unsupervised relation extraction by mining the web and showed major improvements in the detection of new facts from only few initial seeds. They used a popular web search engine as a major component of their system. Our focused crawling strategy builds upon the idea of utilizing a *language model* to discriminate between relevant and irrelevant web documents. The key idea of this methodology is that web pages coming from a certain domain — which implies the use of a particular vocabulary (Biber, 1995) — link to other documents of the same domain. The assumption is that the crawler will most likely stay in the same topical domain as the initial language model was generated from.

Using the enlarged corpus, we compute distributional similarities for entity pairs and dependency paths, and investigate both directions: a) grouping entity pairs, and b) grouping dependency paths in order to find generalized relations. Initial results and further details of this work can be found in (Remus, 2014).

Next Steps Remus (2014) indicates promising directions, but a full evaluation is still missing and still has to be carried out. Further, we plan to apply methods for supervised relation classification using unsupervised features by applying similar ideas and methodologies as explained above.

3.2 Index Term Identification

In this section, we present our analysis of approaches for index term identification on the *pedocs* document collection. Index terms support users by facilitating search (Song et al., 2006) and providing a short summary of the topic (Tucker and Whittaker, 2009). We evaluate two approaches to solve this task: (1) index term extraction and (ii) index term assignment. The first one extracts index terms directly from the text based on lexical characteristics, and the latter one assigns index terms from a list of frequently used index terms.

Task Approaches for index term identification in documents from a given document collection find important terms that reflect the content of a document. Document collection knowledge is important because a good index term highlights a specific subtopic of a coarse collection-wide topic. Document knowledge is important because a good index term is a summary of the document's text. Thesauri which are available for English are not available in every language and less training data may be available if index terms are to be extracted for languages other than English.

Dataset We use manually assigned index terms, which were assigned by trained annotators, as a gold standard for evaluation. We evaluate our approaches with a subset of 3,424 documents.[5] Annotators for index terms in *pedocs* were asked to add as many index terms as possible, thus leading to a high average number of index terms of 11.6 per document. The average token length of an index term is 1.2. Hence, most index terms in *pedocs* consist of only one token but they are rather long with on average more than 13 characters. This is due to many domain-specific compounds.

Approaches We apply index term extraction approaches based on tf-idf (Salton and Buckley, 1988) using the *Keyphrases* module (Erbs et al., 2014) of *DKPro*, a framework for text processing,[6] and an index term assignment approach using the *Text Classification* module, abbreviated as *DKPro TC* (Daxenberger et al., 2014). The index term extraction approach weights all nouns and adjectives in the document with their frequency normalized with their inverse document frequency. With this approach, only index terms mentioned in the text can be identified. The index term assignment approach uses decision trees (J48) with BRkNN (Spyromitros et al., 2008) as a meta algorithm for multi-label classification (Quinlan, 1992). Additionally, we evaluate a hybrid approach, which combines the extraction and assignment approach by taking the highest ranked index terms of both approaches.

Type	Precision	Recall	R-prec.
Extraction	11.6%	15.5%	10.2%
Assignment	**33.0%**	6.1%	6.6%
Hybrid	20.0%	**17.9%**	**14.4%**

Table 1: Results for index term indentification approaches

Table 1 shows results for all three approaches in terms of precision, recall, and R-precision. The extraction approach yields good results for recall and R-precision, while the assignment approach yields a high precision but a lower recall and R-precision. Assignment determines few index terms with high confidence that increases precision but lowers recall and R-precision, while extraction allows for identifying many index terms with lower confidence. The hybrid approach (Erbs et al., 2013), in which index term extraction and assignment are combined, results in better performance in terms of recall and R-precision.

Next Steps We believe that using semantic resources will further improve index term identification by grouping similar index terms. Additionally, we plan to conduct a user study to verify our conclusion that automatic index term identification helps the users in finding documents.

3.3 Identification and Exploration of Dataset Names in Scientific Literature

Datasets are the foundation of any kind of empirical research. For a researcher, it is of utmost importance to know about relevant datasets and their state of publications, including a dataset's characteristics, discussions, and research questions addressed.

Task The project consists of two parts. First, references to datasets, e.g. "PISA 2012" or "National Educational Panel Study (NEPS)", must be extracted from scientific literature. This step can be defined as a Named Entity Recognition (NER) task with specialized named entities.[7]

Secondly, we want to investigate functional contexts, which can be seen as the purpose of mentioning a certain dataset, i.e., introducing,

[5] We divided the entire dataset in a development, training, and test set.

[6] https://code.google.com/p/dkpro-core-asl/

[7] We extract the NEs from more than 300k German abstracts of the *FIS Bildung* dataset.

discussing, side-mentioning, criticizing, or using a dataset for secondary analysis.

Approaches First of all, the term *dataset* must be defined for our purposes. Although there is a common sense about what a dataset is, no formal definition exists. As a starting point, we use a list of basic descriptive features from Renear et al. (2010), which are *grouping, content, relatedness,* and *purpose*. As those features are not precise enough for our case, we need to further refine unclear aspects, like how to treat nested datasets,[8] or general names like PISA, which are not datasets in the strict sense, as they denote projects comprised of multiple datasets. Another question being discussed with domain experts is, if only primary datasets or also aggregated datasets, e.g., statistical data from the Zensus (German censuses), are relevant or if they should be treated differently.

There is a large number of approaches for NER (Nadeau and Sekine, 2007). Due to the lack of labeled training data and the high annotation costs, we have to resort to three un- and semi-supervised methods; *a*) an information engineering approach, where we manually crafted rules, *b*) a baseline classifier using active learning (Settles, 2011), and *c*) a bootstrapping approach for iterative pattern induction (Riloff and Jones, 1999), which has been used successfully by Boland et al. (2012) on a similar task.[9]

Challenges Apart from general NER challenges like ambiguity, variants, multi-word names or boundary determination (Cohen and Hersh, 2005), extracting dataset names comes with additional challenges. First, not even a partially complete list of names is available, and second, there is no labelled training data. A user study showed, that manual labelling is very costly. Furthermore, dataset names are sparse in our dataset and most names only occur once.

Next Steps After evaluating the different approaches, named entity resolution must be conducted on the results to map each name variant to a specific project or dataset entity. To finally explore the functional contexts, we will use clustering methods to determine clusters of contexts. After verifying and refining them with domain experts, multi-label classification can be applied to assign functional contexts to dataset mentions.

3.4 Identification of Argumentation Structures in Scientific Publications

One of the main goals of any scientific publication is to present new research results to an expert audience. In order to emphasize the novelty and importance of the research findings, scientists usually build up an argumentation structure that provides numerous arguments in favor of their results.

Task The goal of this project is to automatically identify argumentation structures on a fine-grained level in scientific publications in the educational domain and thereby to improve both reading comprehension and information access. A potential use case could be a user interface which allows to search for arguments in multiple documents and then to combine them (for example arguments in favor or against private schools). See Stab et al. (2014) for an overview of the topic Argumentation Mining and a more detailed description of this project as well as some challenges.

Dataset As described in section 2.2, the *pedocs* and *FIS Bildung* datasets are very heterogeneous. In addition, it is difficult to extract the structural information from the PDF files (e.g. headings or footnotes). For this reason, we decided to create a new dataset consisting of publications taken from PsyCONTENT which all have a similar structure (about 10 pages of A4, empirical studies, same section types) and are available as HTML files.[10]

Approaches Previous works have considered the automatic identification of arguments in specific domains, for example in legal documents (Mochales and Moens, 2011) or in online debates (Cabrio et al., 2013). For scientific publications, more coarse-grained approaches have been developed, also known as Argumentative Zoning (Teufel et al., 2009; Liakata et al., 2012; Yepes et al., 2013). To the best of our knowledge, there is

[8] E. g. the PISA project contains several datasets from multiple studies, like PISA 2000, PISA 2003, PISA-International-Plus, or even research specific sub-datasets could be considered.

[9] However, their dataset was completely different, so that it is unclear at this point if bootstrapping performs well on our task.

[10] http://www.psycontent.com/

no prior work on identifying argumentation structures on a fine-grained level in scientific fulltexts yet.

We define an argument as consisting of several argument components which are related: an argument component can either support or attack another argument component; the argument component being supported or attacked is also called claim. We set the span of an argument component to be a sentence. In the following (fictitious) example, each sentence (A, B, C, D) can be seen as an argument component connected by support and attack relations as visualized in figure 2.

A *Girls are better in school.* **B** *In the XY study, girls performed better on average.* **C** *One reason for this is that girls invest more time in their homework.* **D** *However, there are also other studies where no differences between girls and boys could be found.*

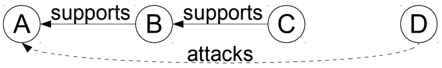


Figure 2: Visualization of an argumentation structure: The nodes represent the four sentences (A, B, C, D), continuous lines represent support relations, dotted lines represent attack relations

Next Steps Due to the lack of evaluation datasets, we are performing an annotation study with two domain experts and two annotators who developed the annotation guidelines. Next, we plan to develop weakly supervised machine learning methods to automatically annotate scientific publications with argument components and the relations between them. The first step will be to distinguish non-argumentative parts from argumentative parts. The second step will be to identify support and attack relations between the argument components. In particular, we will explore lexical features, such as discourse markers (words which indicate a discourse relation, for example "hence", "so", "however") and semantic features, such as text similarity.

3.5 Scalable Multi-label Classification for Educational Research

This project aims at developing and applying novel machine learning algorithms which can be useful for providing methods to automate the processing of scientific literature. Scientific publications often need to be organized in a way of providing high-level and structured information, i.e., metadata. A typical example of a metadata management system is assigning index terms to a document.

Task The problem of assigning multiple terms to a document can be addressed by multi-label classification algorithms. More precisely, our task is to assign multiple index terms in *FIS Bildung*, to a given instance if we have a predefined list of the terms. There are two problems for multi-label classification in the text domain; 1) What kinds of features or which document representations are useful for our task of interest? 2) How do we exploit the underlying structure in the label space?

Dataset and Challenges In *FIS Bildung* database, tens of thousands of index terms are defined, because it is a collection of links to documents coming from diverse institutions each of which deals with different subjects, thereby requiring expertise of index terms maintenance. The difficulty of predicting index terms for a given document is divided largely into two parts. First, only abstracts are available which contain a small number of words compared to fulltexts. Secondly, given a large number of distinct labels, it is prohibitively expensive to use sophisticated multi-label learning algorithms. To be more specific, we have about 50,000 index terms in *FIS Bildung* which most of current multi-label algorithms cannot handle efficiently without a systematic hierarchy of labels. Hence, as a simplified approach, we have focused on 1,000 most frequent index terms as target labels that we want to predict because the rest of them occur less than 20 times out of 300K documents.

Approaches Multi-label classifiers often try to make use of intrinsic structures in a label space by generating subproblems (Fürnkranz et al., 2008) or exploiting predictions of successive binary classifiers for the subsequent classifiers (Read et al., 2011).

Neural networks are a good way for capturing the label structure of multi-label problems, as has been shown in BP-MLL (Zhang and Zhou, 2006). Recent work (Dembczyński et al., 2012; Gao and Zhou, 2013) find inconsistency of natural (convex) rank loss functions in multi-label learning.

Based on these results, Nam et al. (2014) showed that the classification performance can be further increased with methods that have been recently developed in this area, such as Dropout (Srivastava et al., 2014), Adagrad (Duchi et al., 2011), and ReLUs (Nair and Hinton, 2010), on the *FIS Bildung* dataset as well as several text benchmark datasets. Specifically, for multi-label text classification task, the cross-entropy loss function, widely used for classification tasks, has shown to be superior to a loss function used for BP-MLL which try to minimize errors resulting from incorrect ranked labels. Even though the former does not consider label ranking explicitly, it converges faster and perform better in terms of ranking measures. More details can be found in (Nam et al., 2014).

Next Steps Even though our proposed approach has shown interesting results, the original problem remains unsolved. How do we assign multiple labels to an instance where tens or even hundreds of thousands of labels are in our list? To answer this, we are going to transform both instances and labels into lower dimensional spaces while preserving original information or deriving even more useful information (Socher et al., 2013; Frome et al., 2013) which enables us to make predictions for unseen target labels at the time of training.

3.6 Temporally Dynamic Networks of Topics and Authors in Scientific Publications

In this part of the KDSL program, we build a probabilistic network for various aspects of scientific publication. The important entities are authors, ideas and papers. From authors, writing style and communities can be modelled. From papers, index terms, citations and arguments can be extracted. In reality, all these factors affect each other and when they are considered in one probabilistic model, the precision of each model should be improved, as a result of enhanced context.

Task and Data At first, we took the *pedocs* dataset and performed temporal analysis as the first dimension of the probabilistic network. By tracking the occurrence of index terms in the last 33 years, we monitor the development of topics in the corpus. The first assumption is that trendy topics lever-up the frequency of their representing keyword in the corpus at each period of time. The second assumption is that the significant co-occurrence of keywords indicates the emergence of new research topics.

Approach Co-occurrence has been used in trend detection (Lent et al., 1997). To capture more interesting dynamic behaviors of the index terms, we experimented with different measures to find index term pairs of interest. Covariance, co-occurrence, Deviation-from-Random, Deviation-from-Lower-Envelop are some of the measures we used to detect the co-developing terms. The covariance, co-occurrence are the standard statistical measures in temporal relation analysis (Kontostathis et al., 2004). The other measures are developed in our work, which exhibit the capability to gain more insights from the data.

Interestingly, some of the measures can reveal strong semantic relatedness between the index terms, e.g., Internationalisierung - Globalisierung (Internationalization - Globalization). This phenomenon indicates a potential unsupervised semantic-relatedness measure. And generally, our methodology can find interesting pairs of index terms that help the domain researcher to gain more insight into the data, please see (Ma and Weihe, 2014) for detailed examples of the findings.

For the manually selected index terms (about 300), we collaborated with domain experts from DIPF to assign categories (Field, Topic, Method, etc.) to them. With the category, we can look for the term pairs of our interest. For example, we can focus on the method change of topics, by limiting the categories of a term pair to Topic and Method.

Next Steps One critical problem to these analyses is data sparsity. Some experiments can only output less than 10 instances, which may be insufficient for statistically significant results. We adapt the methods to larger datasets like *FIS Bildung*. Besides optimization, we will work on other new measures and evaluate the results with the help of domain experts.

3.7 Structured Tag Clouds

Tag clouds are popular visualizations on web pages. They visually depict a set of words in a spatial arrangement with font size being mapped to an approximation of term importance such as term frequency. It is supposed that by organizing the words according to some (semantic) term relation, the usefulness of tag clouds can be further improved (see e.g., (Hearst and Rosner, 2008; Rivadeneira et al., 2007)). The goal of this project is to investigate if this assumption holds true and to research the optimal design and automatic generation of such structured tag clouds (Figure 3).

Task To approach our research goal, three main tasks can be distinguished: First, we examine how humans structure tags when being told that the resulting tag cloud should provide a quick overview of a document collection. Second, based on the determined criteria that the participants of our study aimed at, when layouting the clouds, we develop methods for automatically generating structured tag clouds. Finally, the performance of users employing structured tag clouds is compared to unstructured ones for specific tasks.

Dataset As the name suggests, tag clouds are often employed to visualize a set of (user-generated) tags. In our research, we use user-generated tags from social bookmarking systems such as BibSonomy[11] or Edutags[12]. We expect that the results can be generalized to similar data such as index terms assigned to scientific publications or these extracted from a document (collection).

Challenges There are many ways to (semantically) structure tags (e.g., based on co-occurrences or lexical-semantic relations). However, our goal must be not to generate an arbitrary tag structure but to organize tags in a way that is conclusive for human users and thus easy to read. A key challenge here is that no ground-truth exists saying how a specific tag set is arranged best.

Approaches We conducted a user study in which the participants were asked to manually arrange user-generated tags of webpages that were retrieved by a tag search in the social bookmarking system BibSonomy. Being aware that no single ground-truth exists, we investigated the criteria underlying the layout in detailed post-task interviews. Those criteria are now the basis for researching automatic algorithms and visual representations that can best approximate the user-generated layouts. Finally, unstructured and structured tag clouds will be compared in a study in which the performance of users in specific tasks is measured.

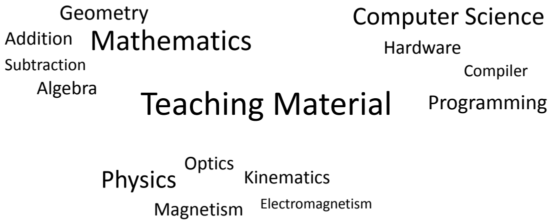


Figure 3: Example for a structured tag cloud.

Results & Next Steps In (Oelke and Gurevych, 2014) we presented the results of our user study. While previous work mainly relies on co-occurrence relations when building structured tag clouds, our study revealed that semantic associations are the main criterion for human layouters to build their overall structure on. Co-occurrence relations (i.e., two tags that are at least once assigned to the same bookmarked webpage) were only rarely taken into account, although we provided access to this information.

While some participants included all tags in their final layout, others consequently sorted out terms that they deemed redundant. Lexical-semantic relations (e.g., synonyms or hypernyms) turned out to be the basis for determining such redundant terms. Furthermore, small clusters were preferred over large ones and large clusters were further structured internally (e.g., arranged according to semantic closeness, as a hierarchy, or split into subclusters).

Next, we will work on the algorithmic design and finally evaluate the performance of structured tag clouds.

4 Conclusion

This paper describes 'Knowledge Discovery in Scientific Literature', a unique graduate program with the goal to make the knowledge concealed in various kinds of educational research literature more easily accessible. Educational researchers will benefit from automatically processed information on both local and global scopes. Local information consists of index terms (Sec. 3.2, 3.7, 3.5), relations (Sec. 3.1), dataset mentions and

[11] http://www.bibsonomy.org/

[12] http://www.edutags.de/

functional contexts (Sec. 3.3), and argumentation structures (Sec. 3.4). On the level of the entire corpus, temporal evolution of index terms and authors can be provided (Sec. 3.6).

Each sub-project aims at new innovations in the particular field. The close connection between computer science researchers and educational researchers helps us with immediate evaluation by end users.

Acknowledgments

This work has been supported by the German Institute for Educational Research (DIPF) under the Knowledge Discovery in Scientific Literature (KDSL) program. We thank Alexander Botte, Marius Gerecht, Almut Kiersch, Julia Kreusch, Renate Martini, Alexander Schuster from the Information Center for Education at DIPF for providing valuable feedback.

References

Doris Bambey and Agathe Gebert. 2010. Open-Access-Kooperationen mit Verlagen – Zwischenbilanz eines Experiments im Bereich der Erziehungswissenschaft. *BIT Online*, 13(4):386.

Douglas Biber. 1995. *Dimensions of Register Variation: A Cross-Linguistic Comparison*. Cambridge University Press, Cambridge.

Katarina Boland, Dominique Ritze, Kai Eckert, and Brigitte Mathiak. 2012. Identifying References to Datasets in Publications. In *Theory and Practice of Digital Libraries*, pages 150–161, Paphos, Cyprus.

Elena Cabrio, Serena Villata, and Fabien Gandon. 2013. A Support Framework for Argumentative Discussions Management in the Web. In *The Semantic Web: Semantics and Big Data*, pages 412–426. Montpellier, France.

Carola Carstens, Marc Rittberger, and Verena Wissel. 2011. Information search behaviour in the German Education Index. *World Digital Libraries*, 4(1):69–80.

Aaron M. Cohen and William R. Hersh. 2005. A survey of current work in biomedical text mining. *Briefings in bioinformatics*, 6(1):57–71.

Dmitry Davidov, Ari Rappoport, and Moshe Koppel. 2007. Fully Unsupervised Discovery of Concept-Specific Relationships by Web Mining. In *Proceedings of the 45th Annual Meeting of the Association of Computational Linguistics*, pages 232–239, Prague, Czech Republic.

Johannes Daxenberger, Oliver Ferschke, Iryna Gurevych, and Torsten Zesch. 2014. DKPro TC: A Java-based Framework for Supervised Learning Experiments on Textual Data. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. System Demonstrations*, pages 187–192, Baltimore, MD, USA.

Krzysztof Dembczyński, Wojciech Kotłowski, and Eyke Hüllermeier. 2012. Consistent Multilabel Ranking through Univariate Losses. In *Proceedings of the 29th International Conference on Machine Learning*, pages 1319–1326, Edinburgh, UK.

John Duchi, Elad Hazan, and Yoram Singer. 2011. Adaptive Subgradient Methods for Online Learning and Stochastic Optimization. *Journal of Machine Learning Research*, 12:2121–2159.

Nicolai Erbs, Iryna Gurevych, and Marc Rittberger. 2013. Bringing Order to Digital Libraries: From Keyphrase Extraction to Index Term Assignment. *D-Lib Magazine*, 19(9/10):1–16.

Nicolai Erbs, Pedro Bispo Santos, Iryna Gurevych, and Torsten Zesch. 2014. DKPro Keyphrases: Flexible and Reusable Keyphrase Extraction Experiments. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. System Demonstrations*, pages 31–36, Baltimore, MD, USA.

Andrea Frome, Greg S Corrado, Jon Shlens, Samy Bengio, Jeff Dean, Tomas Mikolov, et al. 2013. DeViSE: A Deep Visual-Semantic Embedding Model. In *Advances in Neural Information Processing Systems 26*.

Johannes Fürnkranz, Eyke Hüllermeier, Eneldo Loza Mencía, and Klaus Brinker. 2008. Multilabel Classification via Calibrated Label Ranking. *Machine Learning*, 73(2):133–153.

Wei Gao and Zhi-Hua Zhou. 2013. On the Consistency of Multi-Label Learning. *Artificial Intelligence*, 199–200:22–44.

Zellig S Harris. 1954. Distributional structure. *Word*, 10:146–162.

Marti A. Hearst and Daniela Rosner. 2008. Tag Clouds: Data Analysis Tool or Social Signaller? In *Proceedings of the 41st Annual Hawaii International Conference on System Sciences*, pages 160–160.

Petr Knoth and Zdenek Zdrahal. 2012. CORE: Three Access Levels to Underpin Open Access. *D-Lib Magazine*, 18(11/12).

April Kontostathis, Leon M Galitsky, William M Pottenger, Soma Roy, and Daniel J Phelps. 2004. A Survey of Emerging Trend Detection in Textual Data Mining. In *Survey of Text Mining*, pages 185–224. Springer.

Brian Lent, Rakesh Agrawal, and Ramakrishnan Srikant. 1997. Discovering Trends in Text Databases. In *Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining*, volume 97, pages 227–230, Newport Beach, CA, USA.

Maria Liakata, Shyamasree Saha, Simon Dobnik, Colin Batchelor, and Dietrich Rebholz-Schuhmann. 2012. Automatic recognition of conceptualization zones in scientific articles and two life science applications. *Bioinformatics*, 28(7):991–1000.

Dekang Lin and Patrick Pantel. 2001. DIRT - Discovery of Inference Rules from Text. In *Proceedings of the 7th International Conference on Knowledge Discovery and Data Mining*, pages 323–328, San Francisco, CA, USA.

Zheng Ma and Karsten Weihe. 2014. Temporal analysis on pairs of classified index terms of literature databases. In *Proceedings of the 10th International Conference on Webometrics, Informetrics, and Scientometrics (WIS) and the 15th COLLNET Meeting 2014*, Ilmenau, Germany.

Raquel Mochales and Marie-Francine Moens. 2011. Argumentation mining. *Artificial Intelligence and Law*, 19(1):1–22.

David Nadeau and Satoshi Sekine. 2007. A survey of named entity recognition and classification. *Lingvisticae Investigationes*, 30(1):3–26.

Vinod Nair and Geoffrey E. Hinton. 2010. Rectified Linear Units Improve Restricted Boltzmann Machines. In *Proceedings of the 27th International Conference on Machine Learning*, pages 807–814, Haifa, Israel.

Jinseok Nam, Jungi Kim, Eneldo Loza Mencía, Iryna Gurevych, and Johannes Fürnkranz. 2014. Large-scale Multi-label Text Classification – Revisiting Neural Networks. In *Proceedings of the European Conference on Machine Learning and Knowledge Discovery in Databases*, pages 437–452, Nancy, France.

Daniela Oelke and Iryna Gurevych. 2014. A Study on the Layout of Human-Generated Structured Tag Clouds. In *Proceedings of the International Working Conference on Advanced Visual Interfaces*, Como, Italy.

John Ross Quinlan. 1992. *C4.5: Programs for Machine Learning*. Morgan Kaufmann Publishers.

Jesse Read, Bernhard Pfahringer, Geoff Holmes, and Eibe Frank. 2011. Classifier Chains for Multi-label Classification. *Machine Learning*, 85(3):333–359.

Steffen Remus. 2014. Unsupervised Relation Extraction of In-Domain Data from Focused Crawls. In *Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 11–20, Gothenburg, Sweden.

Allen H. Renear, Simone Sacchi, and Karen M. Wickett. 2010. Definitions of Dataset in the Scientific and Technical Literature. *Proceedings of the American Society for Information Science and Technology*, 47(1):1–4.

Ellen Riloff and Rosie Jones. 1999. Learning Dictionaries for Information Extraction by Multi-Level Bootstrapping. In *Proceedings of the National Conference on Artificial Intelligence and the Innovative Applications of Artificial Intelligence Conference Innovative Applications of Artificial Intelligence*, pages 474–479, Orlando, FL, USA.

Anna W. Rivadeneira, Daniel M. Gruen, Michael J. Muller, and David R. Millen. 2007. Getting Our Head in the Clouds: Toward Evaluation Studies of Tagclouds. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 995–998, San Jose, CA, USA.

Gerard Salton and Christopher Buckley. 1988. Term-weighting Approaches in Automatic Text Retrieval. *Information Processing & Management*, 24(5):513–523.

Burr Settles. 2011. Closing the Loop: Fast, Interactive Semi-supervised Annotation with Queries on Features and Instances. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 1467–1478, Edinburgh, UK.

Richard Socher, Milind Ganjoo, Christopher D Manning, and Andrew Ng. 2013. Zero-Shot Learning Through Cross-Modal Transfer. In *Advances in Neural Information Processing Systems 26*.

Min Song, Il Yeol Song, Robert B. Allen, and Zoran Obradovic. 2006. Keyphrase Extraction-based Query Expansion in Digital Libraries. In *Proceedings of the 6th ACM/IEEE-CS Joint Conference on Digital Libraries*, pages 202–209, Chapel Hill, NC, USA.

Eleftherios Spyromitros, Grigorios Tsoumakas, and Ioannis P. Vlahavas. 2008. An Empirical Study of Lazy Multilabel Classification Algorithms. In *Artificial Intelligence: Theories, Models and Applications*, pages 401–406.

Nitish Srivastava, Geoffrey E. Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan R. Salakhutdinov. 2014. Dropout: A Simple Way to Prevent Neural Networks from Overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Christian Stab, Christian Kirschner, Judith Eckle-Kohler, and Iryna Gurevych. 2014. Argumentation mining in persuasive essays and scientific articles from the discourse structure perspective. In *Frontiers and Connections between Argumentation The-*

ory and Natural Language Processing, Bertinoro, Italy.

Simone Teufel, Advaith Siddharthan, and Colin Batchelor. 2009. Towards Discipline-Independent Argumentative Zoning: Evidence from Chemistry and Computational Linguistics. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 1493–1502, Edinburgh, UK.

Simon Tucker and Steve Whittaker. 2009. Have A Say Over What You See: Evaluating Interactive Compression Techniques. In *Proceedings of the 2009 International Conference on Intelligent User Interfaces*, pages 37–46, Sanibel Island, FL, USA.

Antonio Jimeno Yepes, James G. Mork, and Alan R. Aronson. 2013. Using the Argumentative Structure of Scientific Literature to Improve Information Access. In *Proceedings of the 2013 Workshop on Biomedical Natural Language Processing*, pages 102–110, Sofia,Bulgaria.

Min-Ling Zhang and Zhi-Hua Zhou. 2006. Multi-Label Neural Networks with Applications to Functional Genomics and Text Categorization. *IEEE Transactions on Knowledge and Data Engineering*, 18(10):1338–1351.

Testing user interaction with LSP e-lexicographic tools: A case study on active translation of environmental terms

Laura Giacomini
Department of Translation and Interpreting
University of Heidelberg
Plöck 57a, D-69117 Heidelberg
laura.giacomini@iued.uni-heidelberg.de

Abstract

This paper summarizes the results of a test carried out by MA students in translation and aimed at assessing the usability of three LSP online resources dealing with environment terminology during an active L1-L2 translation task. Data access has also been introduced as a subset of usability features with immediate relevance to the specific target user group. The ultimate purpose is to make observations on the translator's lexicographic needs and the benefits of usability tests for the creation of new tools.

1 Introduction

In the modern, competitive era of e-lexicography, usability tests potentially play a crucial role in identifying structural and contentual properties of digital lexicographic tools, the primary function(s) of which need to be tailored to well-defined users groups and usage situations (Bergenholtz 2012, Bothma 2012). This study relies on the definition of function proposed in the framework of the Function Theory of lexicography: "a lexicographical function is the satisfaction of the specific types of lexicographically relevant need that may arise in a specific type of potential user in a specific type of extra-lexicographical situation" (Tarp 2008: 81). From this perspective, the paper summarizes the results of a test carried out in the academic environment by MA translation students and aimed at assessing the usability of three LSP online resources during an active translation task. Section 2 briefly discusses the topic of usability in relation to LSP dictionaries and to translation-oriented terminological representation. Section 3 is dedicated to the test procedure, performance and results. Finally, section 4 draws some key conclusions on relevant lexicographic properties related to a L1-L2 text-traductive function and on the necessity of systematic usability tests. Today, the question of the status of specialised lexicography, both in relation to linguistics and information science, is being discussed simultaneously with the continuous expansion of the market (Fuertes-Olivera/Tarp 2014, Fuertes-Olivera 2013, Tarp 2012). A new challenge also consists of tracing a typology of lexicographic resources which takes into account the emergence of hybrid and multifunctional forms, such as the ones selected for this test. In line with the view on lexicographically designed information tools expressed by Leroyer (2011), in what follows they will be designated comprehensively as *e-lexicographic tools*, in order to adequately account for their heterogeneous properties.

2 Usability of LSP e-lexicographic tools for translation purposes

The concept of usability will be employed hereafter as a set of properties that make a piece of software successful in terms of achievement of the user's goals. According to the international standard ISO 9241-11:1998, "Guidance on usability", these properties are effectiveness (degree of task completion), efficiency (amount of time required to complete a task) and user satisfaction. They can, of course, also be tested in lexicographic tools as far as a specified context of use is available, with a specific user group performing a task in a given environment (Heid 2012). As maintained by the Function Theory, among user situations and correspondent communicative functions is the L1-L2 text-traductive function (Tarp 2008: 86, 158 ff.), a threefold process connected to function-related and usage-related information needs which will be described in detail in section 3.2. In the practice of specialised translation, the effectiveness of an e-lexicographic item as a usability criterion rests primarily upon the availability of helpful (meta)linguistic and encyclopaedic data. At the same time, extensive access to sample data in the form of corpus instances, for example, is perceived by a translator as a highly valuable feature, since it can successfully replace extra-lexicographic search for corpus concordances and parallel texts (Hvelplund et al. 2013). Data access, for this reason, has been considered as a subset of usability features, and has been accounted for in the performed test.

3 Usability of selected tools: test procedure, performance and results

3.1 The selected tools

The observations here described were made during two workshops on specialised lexicography which took place at the Institute for Translation and Interpreting, Heidelberg University, in 2013 and 2014. The 18 participants were students of the MA in Translation Studies programme, most of whom had firsthand experience as professional specialised translators. They already had quite extensive knowledge of Terminology Management features of widespread CAT-tools (e.g. SDL Trados, Across, DéjàVu), which had also been discussed during a separate course. For these reasons, at the time of the workshop the participants held clear expectations of the degree of user-friendliness and data completeness e-lexicographic tools should potentially offer translators to effectively support their work. Each workshop focused on usability of digital resources such as glossaries, dictionaries and term bases covering the specialised fields of economy/finance, law and environment. This paper reports the main results of a comparison between three lexicographic tools dealing with environment terminology:

EcoLexicon (University of Granada) http://ecolexicon.ugr.es

EPA Terminology Services (US Environmental Protection Agency) http://ofmpub.epa.gov and

DiCoEnviro (Observatoire de linguistique Sens-Texte OLST, Montreal) http://olst.ling.umontreal.ca.

The basic requirement for a comparison was fulfilment of three criteria: terminology in English, digital-only form, and outward representation of terms and concepts relations. In the following outline, the selected tools will be briefly described and compared in their main functional, structural and contentual features according to predefined, differently distributed metalexicographic parameters (cf. also Fata 2009, Wang 2001).

- **Object**:

  EcoLexicon: linguistic/encyclopaedic

  EPA: encyclopaedic

  DiCoEnviro: linguistic

- **Languages**:

  EcoLexicon/DiCoEnviro: multilingual

  EPA: monolingual

- **Target user**:

  EcoLexicon: translators and other language professionals

  EPA: internal use in support of EPA's environmental protection mandate

  DiCoEnviro: language professionals, but also interested non-experts

- **Method/theory**:

  EcoLexicon: Frame-Based Terminology

  EPA: user-oriented and well-formed vocabularies

  DiCoEnviro: Sens-Texte Theory

- **Knowledge base**:

  EcoLexicon: terms (words and MWEs), concepts, relations, categories

EPA: terms (words and MWEs), vocabularies (keyword lists, glossaries, taxonomies, thesauri, ontologies)

DiCoEnviro: terms and lexical relations

3.2 Testing user interaction with the selected tools

In order to gain realistic insights into the interaction of translators with these lexicographic tools, a set of tasks was designed and carried out in the environment that was closest to the professional translators' actual work environment (Nielsen/Fuertes-Olivera 2012). The participants were provided with monolingual and bilingual general language print dictionaries, as well as access to online documentation (e.g. parallel texts, online dictionaries, online encyclopaedias). In the translation situation devised for the study, a source text in the translator's native language or a language which was mastered at nearly native level had to be translated into a foreign language. The reference source languages (SL) were German/French and the target language (TL) was English. EcoLexicon includes both SL languages, DiCoEnviro only French, and the EPA tool neither.

The participants were not asked to translate the entire source text, rather to concentrate on specific passages which included terminology in the form of single lexical items or multiword expressions. Usability of the three resources was assessed by testing the degree of effectiveness, efficiency and user satisfaction in retrieving information on terms/concepts such as *climate change*, which will be exemplified here. In what follows, each step involved in the usability testing is paired with one of the stages in the translation process and specific tasks (Nord 2005, Gerzymisch-Arbogast 2005). The final stage in the active translation process involves text (re)production in the foreign language with a communicative purpose. However, before producing the target text/passages, a cognitive situation also occurs in the first, or reception, phase and in the second, or transfer, phase of the process. The translator needs to retrieve extralinguistic information about a particular concept and how it relates to its domain in both linguistic systems. Data access, as interpreted in section 2, was especially tested in task 3a.

Reception stage:

1a) linguistic data: find equivalent term(s) of *Klimawechsel/changement climatique* in the TL,

1b) encyclopaedic data: find information about the concept CLIMATE CHANGE and its position inside the domain in the SL

Transfer stage:

2a) linguistic data: not required at this stage,

2b) encyclopaedic data: compare the concept/domain in the SL and the TL

Reproduction stage:

3a) linguistic data: find the context of usage (concordances) and collocates of the term *climate change*,

3b) encyclopaedic data: not required at this stage

No statistical evaluation was conducted of the test-related data, and final observations were based on the translators' feedback reports. The test was performed individually and in the presence of a moderator, who explained the tasks and provided the participants with a paper feedback form containing the correspondent questions (cf. tasks 1a-3b) and related instructions. Feedback had to be submitted for every task performed on each tool by answering the following questions:

A) How was the task accomplished and which steps were involved? Please specify if other resources were necessary to complete the task.

B) How much time did you spend on the task?

C) Please rate your overall satisfaction with the tool in performing the task: not satisfied, somewhat satisfied, satisfied, highly satisfied.

D) Please leave your final comments.

Only for task 3a was a question added to group A, as follows:

What kind of access does the tool grant to its lexical database?

The design of the feedback form enabled a targeted collection of users' impressions on effectiveness (as well as data access, where relevant), efficiency and user satisfaction, and allowed for a final group discussion about the participants' feedback. Expert evaluation methods were rejected in favour of a user testing process, with feedback provided after each task. This method was chosen as the most adequate to observe the interaction of potential users with a system while performing a translation task during a testing session (a detailed overview of usability evaluation methods is offered, among others, by Novick/Hollingsed 2007 and JISC 2004). Both the e-lexicographic tools and the text to be translated were new to the participants; however, they could rely on their prior knowledge of other tools, as well as their language skills and translational strategies. As a matter of fact, expert evaluation methods such as a cognitive walkthrough or heuristic evaluation, in which experts step through a series of tasks either simulating the users' skills and goals or assessing usability of a tool against a pre-defined set of heuristics, could not have been applied in the context of regular coursework. Of course, they would be quite helpful to integrate the results of user testing with the feedback received by specialists with extensive usability knowledge, as well as linguistic and domain experience. Other methods, such as questionnaires or field interviews, were also rejected since they do not envisage implementation inside a testing session with users.

3.3 Presentation of the test results on *climate change*

A professional translator needs to have fairly deep knowledge of the specialised field in which he/she is usually working, both from a terminological and a conceptual point of view. Even when producing a text in a foreign language, the translator may already have potential solutions in mind and just need suitable resources to test his/her hypothesis and identify possible contextual variants. This is often the case with borrowings, both lexical and semantic: terms in two different languages share formal and contentual traits which may be easily inferred by a language expert, irrespective of whether the direction of assimilation (i.e. the etymology of the borrowing) is known. *Klimawechsel* and *changement climatique* , which can be classified as synonymic calques from American English (cf. the classification of borrowings in Giacomini 2012 and Scarpa 2008: 61-63), are a good example of that. Most participants involved in the test expected an English equivalent to be *climate change* but were not sure as to whether further equivalents might be available and if climate change would fit exactly into the given context.

EcoLexicon allows for a term/concept search query and, in this way, for quick identification of the equivalent *climate change* with its variant *climatic change* (task 1a). Both *Klimawechsel* and *changement climatique* can be found in the knowledge base. The concept CLIMATE CHANGE itself is represented in the central interactive map (task 1b), which displays the multilingual lexicalisations of the concept, as well as direct and indirect relations among all domain-related concepts (task 2b): generic-specific relations such as (CLIMATE CHANGE is a type of PROCESS), part-whole relations such as (EXOSPHERE is part of the ATMOSPHERE) and non-hierarchical relations (CLIMATIC CHANGE is a result of ATMOSPHERIC POLLUTION). The microstructure comprises a brief definition of the term which virtually summarises the graphical conceptual representation. The term/concept is also ostensively explicated by linked hypertexts and images. A major drawback of this system is that there is no evidence that possible culture-specific differences in conceptualisation are actually taken into consideration, and the user may be led into thinking that the terminological definition or the relations between concepts, for instance, are valid without exception for all displayed languages.

In the final stage of the process, the translator can retrieve linguistic information on the term *climate change* from a collection of concordances (task 3a). A list of phraseologisms should also be generated for each term but, unfortunately, this feature did not seem to work properly during our assessment.

In DiCoEnviro, a search can produce a word, a term, a lexical relation or an expression. The SL term *changement climatique* is best identified through lexical relations of the adjective *climatique* (lexical relation 'type of'), whereas a comparable search on the base *changement* does not directly lead to this result. English equivalents are only shown for single lexical items, so it is necessary to carry out further access acts in order to retrieve the combination *climate change*, which is recorded under the lexical relations of the entry *climate* (task 1a).

The variants *climatic change* and *change in climate* are not intuitively cross-referenced to *climate change*: the former can only be identified through a targeted search for *climatic*, the latter via *change*. Despite the labelling of subject fields in the lexicographic entries and the identification of CLIMATE CHANGE as one of these fields, DiCoEnviro is not designed to provide a conceptual structure, and tasks 1b and 2b cannot be handled with the help of this tool. Linguistic information in the form of concordances and lexical relations is restricted to single lexemes and cannot be directly obtained for the MWE *climate change* (task 3a). The visual representation of the lexical relations (shown in the separate DiCoEnviro Visuel) fails to substantially improve the already produced results.

The EPA tool is a monolingual tool, so that the SL reception (stage 1) and SL-TL transfer (stage 2) need to be tackled by consulting other resources. As far as the monolingual SL perspective is concerned, a translator can find at least sufficient information in general/LSP bilingual dictionaries, encyclopaedic resources and parallel texts. Cognitive transfer, however, implies mapping a conceptual system onto another (task 2b) and, for this reason, requires the support of special domain-oriented resources which are still largely missing in the present LSP e-lexicographic landscape. During the final stage, a term search generated a list of matches extracted from different vocabularies (e.g. glossaries, taxonomies, etc.). For each result, an acronym, a source-based definition, the source vocabulary, the vocabulary type and a preferred term are indicated (task 3a). The list of results may also comprise collocations of the search term, such as *global climate change* or *abrupt climate change*, as far as they constitute autonomous environmental concepts. However, they are not to be understood as linguistic, but rather as purely conceptual data.

3.4 Summarising and interpreting the test results

The following considerations aim to summarise the findings presented in the previous section. They rely primarily on the participants' evaluation of their overall satisfaction with the tool in performing each task (question C), as well as on their final comments (question D). Of course, answers to questions C and D are closely related to the degree of effectiveness (question A) and efficiency (question B) that a specific tool offers a user who wants to accomplish a specific task. Furthermore, in the translator's work environment, efficiency, being the amount of time spent to obtain the required information, often plays a crucial role and the test participants unanimously rated this aspect on the same level of relevance as effectiveness.

Of the three resources analysed, EcoLexicon has been the most discussed in publications written both by the authors (cf., for instance, Faber 2012) and interested linguists and lexicographers. In particular, Fuertes-Olivera/Tarp (2014: 185-189) highlights the main features of this online dictionary, such as definitions which cover the conceptual, lexical and pragmatical properties of each term, the 3D visual thesaurus, access to frames and frame relations, as well as selection patterns, collocational and grammatical tendencies of terminological units. This publication also points out the main drawbacks of the dictionary from a functional perspective: experts do not contribute to the data on a regular basis and the search system is time-consuming and can lead to confusing results, especially in the case of complex concepts or terms simultaneousy belonging to diffent parts of speech. However, the experiences of the participants in the Heidelberg's workshop do not seem to completely match the findings of Fuertes-Olivera/Tarp with respect to a translation situation. Even though EcoLexicon does not target specific user situations, which would be, of course, a desirable feature, during the test it actively supported all stages in the translation process. This may well depend on the type of lexical data the translator needs to deal with.

Generally speaking, however, it turned out to be a remarkably efficient tool, requiring from the user little effort to perform the given tasks in the given order. Effectiveness is clearly imbalanced in favour of encyclopaedic information. On the conceptual level, this resource contains an in-depth description of the contextual domains related to the topic of the environment and visualisation of the knowledge base is highly customizable. However, linguistic data such as frequency or other statistical values of collocates, or pragmatic labels, which are indispensable in the case of translation into a foreign language, are missing.

In DiCoEnviro, linguistic and, in particular, equivalency information is restricted to single lexical items and cannot fully serve a specific translation purpose. From the point of view of effectiveness, this is a good resource. Useful linguistic data such as syntactic functions, roles labels, related meanings, derivatives etc., are available in the term base but, due to a lack of intuitive cross-referencing, they are quite difficult to retrieve in a limited number of steps. For this reason, efficiency was rated as unsatisfactory. Moreover, the complexity of linguistic description requires from the potential user a high degree of familiarity with the metalexicographic apparatus and the theoretical frame of reference, which is comprehensively described in outer texts and several publications by the authors (cf. L'Homme/Robichaud/Rüggeberg 2014).

The EPA tool basically aggregates definite terminological information from multiple online lexicographic and non-lexicographic resources. It was evaluated as being a tool with low effectiveness for active translation, particularly as far as the entire translation process is concerned. Insufficient effectiveness is mainly due to the non-linguistic orientation of this resource. However, it may be useful for gaining a deeper knowledge of the specialised field and, of course, for text reception in English. Efficiency is also a weak point of the EPA tool, due to the considerable effort needed in terms of time to navigate to single external resources.

None of the tools entirely meet the needs of the target user group in terms of performing an active translation task, however user satisfaction was clearly in favour of EcoLexicon because of its intuitive GUI, more rapid access to the required data, concept visualisation by means of interactive maps and coherent cross-referencing of the multilingual layer. Immediate data access, which was analysed during task 3a, is best granted by EcoLexicon, which, in contrast to DiCoEnviro, provides a relatively substantial number of concordances for each search term, although potentially helpful information such as text sources and statistical facts are not supplied.

4 Conclusions

The test performed on translators' interaction with the environmental terminology resources uncovered the essential properties of a LSP e-lexicographic tool with a primary text-traductive function and, more specifically, designed to support active translation (cf. Tarp 2013). Besides an intuitive GUI, minimum requirements may be summarized as follows:

- **Object**:

  - linguistic (L2)

  - encyclopaedic (L1, L2, contrastive)

  Support of active translation requires detailed data representation for the L2 (TL), but also the availability of encyclopaedic data concerning both languages, with special focus on culture-specific differences in conceptual encoding.

- **Function**:

  - translation (monofunctional)

  Monofunctionality needs to be stressed as one of the most relevant characteristics of a good lexicographic resource (s. on this topic Bergenholtz/Bergenholtz 2012), Unfortunately, the analysed tools lack a clear statement by their authors concerning this aspect. As pointed out by Fuertes-Olivera/Tarp (2014) with respect to EcoLexicon and other surveyed online resources, a dictionary should address specific users in specific situations, which also implies that a translation situation should be further distinguished in active and passive translation and that, of course, the translator?s linguistic and encyclopaedic skills in the SL and TL should be carefully considered (Wang 2001).

- **Languages**:

  - L1, L2,

  - coherent cross-linguistic mediostructure

- **Knowledge base**:

  - terms (single words and MWEs)

  - concepts

  - relations among concepts

  An onomasiological structure, assigning terms to concepts along language-independent criteria, has proven to be a valuable approach in modern terminology management tools since it produces coherent cross-referencing and enables subsequent manipulation and addition of linguistic data without compromising the quality and consistency of the underlying conceptual design. Moreover, terminological entries should be both single terms and multiword terms, as long as the latter can be identified with phraseological units such as collocations or idioms. This contributes to the conceptual consistency of the knowledge base and greatly improves the tool's user friendliness.

- **Search options**:

  - all database elements

  - additional filters (e.g. subdomain, kind of conceptual relation, etc.)

  All items in the knowledge base should be separately or jointly searchable to allow the user to perform targeted queries in a reasonably short time. Metadata on the terminological and conceptual level, such as pragmatic markers or subdomain tags, would be of even greater advantage if they could be systematically employed as filters to further narrow down search results.

- **Presentation modes**:

  - text flow

  - conceptual maps

  - further ostensive items

  Presentation of search results and, in general, of sections of the knowledge base, should be made possible using different methods, including textual modes (for instance, paper-dictionary-like or relational-database-like), 2D or 3D conceptual maps for the rapid visualisation of concepts and relations, and possibly ostensive data for direct exemplification.

- **Microstructure**:

  - terminological definition

  - variants and near synonyms

  - pragmatic labels

  - semantic/domain disambiguators

  - equivalents

  - phraseology

  Closer attention should certainly be paid to terminological definitions in order to ensure coherency of the conceptual representation.The definition of climate change in EcoLexicon, for instance, is not completely satisfactory: ”long-term changes in temperature, precipitation, wind, and all other aspects of the Earth's climate in response to physical feedbacks, chemical feedbacks, and changes in terrestrial and aquatic systems caused by humans and nature”. The presence of the lexical elements of the definiendum inside the definition produces a confusing circularity; the extensional method used in listing the aspects of the climate ends with a formulation, ”all other aspects”, which is far too vague for a terminological resource, and the correlation between the feedback and response is ambiguous. The EPA tool provides several definitions of climate change from various sources, which seem to suffer the same drawbacks mentioned for EcoLexicon and in some cases are even contradictory.

  The analysed resources would benefit from a systematic approach to definition (Magris et al. 2001: 83-95, Scarpa 2008: 53-54), in which intensional and extensional modes play clearly distinct roles and ostensive methods (i.e. defining by demonstration) help represent concrete entities through prototypical images, videos etc..

- **Resources**:

  - access to corpus data (concordances, collocations, etc.) and related statistical information

The study has also revealed the need for further, systematic usability testing of LSP resources as the predominant consultation works for a target translators group. Despite the growing attention paid to LSP e-lexicography and the obvious potential of the digital medium for lexicography and terminography in general (Costa 2013), currently available resources are still unbalanced for what concerns user-orientation and monofunctionality. In order to improve the quality of software design and, as a consequence, user satisfaction with LSP products (Lew 2013, Rundell 2012), further tests on ease of use and ease of access to data should be carried out both in the professional environment and in the laboratory, with focus on domain-based considerations rather than on software typology.

References

Henning Bergenholtz. 2012. Access to and Presentation of Needs-Adapted Data in Monofunctional Internet Dictionaries. *e-Lexicography.The Internet, Digital Initiatives and Lexicography*. Continuum, London: 30-53.

Henning Bergenholtz, Inger Bergenholtz. 2012. A Dictionary Is a Tool, a Good Dictionary Is a Monofunctional Tool. *e-Lexicography.The Internet, Digital Initiatives and Lexicography*. Continuum, London: 187-207.

Theo JD. Bothma. 2012. Filtering and Adapting Data and Information in an Online Environment in Response to User Needs. *e-Lexicography.The Internet, Digital Initiatives and Lexicography*. Continuum, London: 71-102.

Rute Costa. 2013. Terminology and Specialised Lexicography: two complementary domains. *Lexicographica*, 29/1: 29-42.

Pamela Faber. 2012. *A Cognitive Linguistics View of Terminology and Specialized Language*. De Gruyter/Mouton, Berlin/Boston.

Idilko Fata. 2009. *Das zweisprachige Translationswoerterbuch fuer Fachsprachen in der wissenschaftlichen Theorie und Praxis*. Tinta Könyvkiadó, Budapest.

Pedro A. Fuertes-Olivera. 2013. The Theory and Practice of Specialised Online Dictionaries for Translation. *Lexicographica*, 29/1: 69-91.

Pedro A. Fuertes-Olivera, Sven Tarp (eds.). 2014. *Theory and Practice of Specialised Online Dictionaries*. Lexicographica Series Maior. De Gruyter/Mouton, Berlin.

Heidrun Gerzymisch-Arbogast. 2005. Introducing Multidimensional Translation. MuTra 2005. Challenges of Multidimensional Translation: Conference Proceedings: 1-15.

Laura Giacomini. 2012. Lexical borrowings in German and Italian IT terminology: At the crossroads between language interference and translation procedures. *Proceedings of the BDÜ Conference "Übersetzen in die Zukunft 2012"*. Berlin 28-30.09.12.

Ulrich Heid. 2012. Electronic Dictionaries as Tools: Toward an Assessment of Usability. *e-Lexicography.The Internet, Digital Initiatives and Lexicography*. Continuum, London: 287-304.

Holger Hvelplund, Adam Kilgarriff, Vincent Lannoy, and Patrick White. 2013. Augmenting online dictionary entries with corpus data for Search Engine Optimisation. *Electronic lexicography in the 21st century: thinking outside the paper. Proceedings of the eLex 2013 conference, 17-19 October 2013, Tallinn*. Estonia. Ljubljana/Tallinn: Trojina, Institute for Applied Slovene Studies/Eesti Keele Instituut.: 66-75.

JISC The Joint Information Systems Committee. 2004. Usability Studies. JISC Services and Information Environment. Version 2.0. *http://www.jisc.ac.uk*. City University London, London.

Patrick Leroyer. 2012. Change of Paradigm: From Linguistics to Information Science and from Dictionaries to Lexicographic Information Tools. *e-Lexicography.The Internet, Digital Initiatives and Lexicography*. Continuum, London: 121-140.

Marie Claude L'Homme, Benoit Robichaud, and Carlos Subirats Rüggeberg. 2014. Discovering Frames in Specialized Domains. *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*. Reykjavik, Iceland.: 1364-1371.

Robert Lew. 2013. Online dictionary skills. *Electronic lexicography in the 21st century: thinking outside the paper. Proceedings of the eLex 2013 conference, 17-19 October 2013, Tallinn*. Estonia. Ljubljana/Tallinn: Trojina, Institute for Applied Slovene Studies/Eesti Keele Instituut.: 16-31.

Marella Magris, Maria Teresa Musacchio, Lorenza Rega, and Federica Scarpa. 2001. *Manuale di terminologia. Aspetti teorici, metodologici e applicativi*. Ulrico Hoepli, Milano.

Sandro Nielsen, Pedro A. Fuertes-Olivera. 2012. Online dictionaries for assisting translators of LSP texts: the Accounting Dictionaries. *International Journal of Lexicography*, 25:2: 191-215.

Christiane Nord. 2005. *Text Analysis in Translation. Theory, Methodology, and Didactic Application of a Model for Translation-Oriented Text Analysis*. Rodopi, Amsterdam/New York.

David G. Novick, Tasha Hollingsed. 2007. Usability inspection methods after 15 years of research and practice. *Departmental Papers (CS). Paper 16. Proceedings of the 25th annual ACM international conference on Design of communication (SIGDOC '07)*. ACM, New York: 249-255.

Michael Rundell. 2012. The road to automated lexicography: An editor's viewpoint. *Electronic Lexicography*. Oxford University Press, Oxford, UK: 15-30.

Federica Scarpa. 2008. *La traduzione specializzata*. Ulrico Hoepli, Milano.

Sven Tarp. 2008. *Lexicography in the Borderland between Knowdledge and Non-Knowledge*. Max Niemeyer Verlag, Tübingen.

Sven Tarp. 2012. Theoretical challenges in the transition from lexicographical p-works to e-tools. *Electronic Lexicography*. Oxford University Press, Oxford, UK: 107-118.

Sven Tarp. 2013. What should we demand from an online dictionary for specialised translation? *Lexicographica*, 29/1: 146-162.

Weiwei Wang. 2001. *Zweisprachige Fachlexikographie: Benutzungsforschung, Typologie und mikrostrukturelle Konzeption*. Peter Lang, Frankfurt am Main.

User-generated Exploratory Search Routes: ENCONTRAR UN TÉRMINO in the Accounting Dictionaries

Pedro A. Fuertes-Olivera
University of Valladolid (Spain)
International Centre for Lexicography
Plaza del Campus, 1, 47011 Valladolid
pedro@emp.uva.es

Patrick Leroyer
University of Aarhus (Denmark)
Centre for Lexicography
School of Business and Social Sciences
Jens Chr. Skous Vej 4, 8000 Aarhus C
pl@asb.dk

Abstract

This paper presents the theories and methods that have been used to develop a specific user-generated search route in the *Spanish Accounting Dictionary*, which consists in offering users the search button ENCONTRAR UN TÉRMINO. This allows users who are uncertain of the exact form of the term to be searched for, or who want to explore the data of a particular term field, to generate their own searches and search strategies by using Boolean operators. With these, users can then retrieve various data categories from the lexicographic database, e.g. they can retrieve lemmas, equivalents, parts of collocations or examples, etc. Clicking on the retrieved data from the list they can then access data that can suit their needs in different situations, typically in cognitive or communicative situations .

1 Introduction

Several researchers defend that lexicography, especially e-lexicography, is an integrated part of the social and information science paradigm. Within this paradigm, lexicography is concerned with the study, design and development of information tools whose distinctive feature must be the interrelationship of three key elements: users, data, and access routes (Fuertes-Olivera and Bergenholtz, 2011; Verlinde, Leroyer and Binon, 2010). The above-mentioned interrelationship has resulted in several lines of work, being the description of a particular access structure the point of discussion in this paper. By access structure we understand the search route that the dictionary user follows during a data consultation procedure (Gouws, 2001: 102). In particular, we will focus on the workings of ENCONTRAR UN TÉRMINO, one of the search routes used in the Accounting Dictionaries that allow users to carry out searches for retrieving hints, i.e. suggestions about lemmas or concepts that may suit users needs in several usage situations. The interest for access has gained momentum recently, e.g. Bergenholtz and Gouws's (2010) seminal publication on accessology, Granger and Paquot's (2010) description of two access modes in their Louvain EAP dictionary, Bergenholtz and Bergenholtz's (2011) analysis of usage based dictionaries, Bosman's (2012) study on the easiness of access of the Afrikaans-Nederlands/Nederlands-Afrikaans diction-ary, and Fuertes-Olivera and Tarp's (2014) presentation of several access routes in the Accounting Dictionaries. This paper follows suit and focuses on searching options with combined search strings (Bergenholtz and Gouws, 2010: 109), i.e. an intelligent search mode with Boolean operators used in user-oriented access.

2 Search routes and situation driven access modes

Lexicographers normally agree on the fact that fast and unimpeded access to the data is of utmost importance in the design of new lexicographical works, particularly in the case of electronic lexicography. This also holds true for already existing e-lexicographical works, in which continuous efforts are being made to transform access and make it easier to the user (LHomme, Robichaud and Leroyer, 2012; Verlinde and Peeter, 2012). In this light, it will come as no surprise that the great majority of De Schryver's lexicographic dreams ten years ago (2003) were directly related to access. As Granger (2012, 3-4) puts it, when speaking of efficiency of access, and quoting De Schryver's dreams: [n]o matter how outstanding the contents of a dictionary, if the contents cannot be accessed in a quick and straightforward way, the dictionary *de facto* fails to be a good dictionary. In spite of this acknowledgement, and in spite of efforts to optimize access, much remains to be done, and improved accessibility remains a key challenge (Granger, 2012: 4). A theoretical explanation to this state of affairs can be found in the fact that the lexicographic consensus on the importance of access does not imply a consensus on the nature of access, as access is treated and prioritised in two opposite directions that profoundly affect the design of lexicographical works:

2.1 From data to access

Data structuring is treated as top priority by the lexicographer and data structuring yields adapted or coinciding access modes: This is the structural perspective, in which access is regarded and treated as a static structure (*Zugriffsstruktur*) coinciding with the structuring of the data, including data description and data presentation. The task of the lexicographer then is to plan and provide adequate access modes to the multiple data structures via structural indicators and matching search options, and hereby make the data accessible from both the outside and the inside of the lexicographic work. This line of thought is much in line with Wiegand"s later work (2008) who speaks of inner and outer accessibility of access structures at different formal levels of the dictionary (*Zugriffsstruktur*). The major danger of this perspective is that intricate data structuring can lead to intricate access and jeopardise usability. As a case in point, the online dictionary of French TLFi (2013) makes use of an advanced search interface in which search options correspond exactly to the intricate structuring of the data into series of linguistic data categories. Another example of the formal coincidence of data structuring and access options can be found in the DiCoube (2013), in which the formulation of queries and the choice or combination of access routes is entirely ruled by an advanced data structuring governed by Meaning-Text Theory and Lexical Functions. Gouws (2013: 352) formulates an intermediary position, and associates access and data structuring to the needs and reference skills of the users: The data distribution structure, the article structure, microstructure, various article-internal structures as well as the access and addressing structures are central to the discussion. It is shown how the development and application of new structures, responding to the needs and reference skills of the intended users of a dictionary, can contribute to better consultation procedures.

2.2 From access to data

Data access is treated as top priority by the lexicographer and data access yields adapted data structuring. This is the functional perspective, which regards access as a dynamic, user-profile and user-situation oriented process. This is the view defended by Bergenholtz and Gouws (2010) who introduce a new terminology to describe the different steps of the process from its start (recognition of the problem and choice of source of information) to its completion (reaching the destination in the information source and concluding the consultation process as successful or not). Of particular interest for this article are the two following types of access: user-profile oriented access (access to different types of data presentation according to the level of expertise of the user: lay person, semi expert, or expert, which in e-dictionaries can be achieved via function buttons), and user-situation oriented access (monofunctional access matching the usage-situation, which is also achieved via function buttons). User-situation oriented access includes communicative situations (text production, reception or translation), cognitive situations (knowledge acquisition of some kind), operative situations (instructions to perform operations as in user manuals and text books), and interpretive (reception of nonverbal signs). In each of these situations, data selection, data structuring and data presentation are specifically adapted to the relevant situation. A third type of user-profile and user-situation driven access should be added, in which access is gained by user-generated search strategies, as in the Spanish Accounting Dictionary (Fuertes-Olivera and al. 2013).

3 Single-targeted search routes versus multi-targeted explorative search routes

According to the tenets of the modern theory of lexicographic functions (Fuertes-Olivera and Tarp 2914; Tarp 2008) and in conformity with the functional access principles that have been stated above, dictionaries are to be seen as lexicographic information tools solely built to cater for the lexicographically relevant information needs of their users in carefully identified use situations, meaning providing easy and unimpeded search modes and search routes, and hereby user-friendly access to the data. The construction of the Accounting Dictionaries was accordingly determined by:

- a preliminary, thorough analysis of user-needs in the field of accounting in different kinds of foreseen user-situations. This analysis was carried on in close cooperation with experts of the profession, who financed the development of the dictionaries in the first place.

- a lexicographic concept matching the user-needs and the profiles of the intended users, including expert users (professional accountants, auditors), semi-experts (professional translators and/or professional translator students), and interested lay-people, as a growing number of international users are genuinely interested in financial communication, particularly the publication of annual results from corporations, and therefore need dictionary assistance to read and understand all the terms and expressions used in such specialised text genres and discourses.

- a dedicated data-base and user-interface design in order to facilitate and adapt data access in the foreseen user situations.

In the Spanish Accounting Dictionary, which is one of the monolingual versions of the series of Accounting Dictionaries, users can make use of 4 functional buttons in order to adapt their search and the related data presentation to the needs of their specific use-situation:

- Reception: clicking on the *reception* (Recepción) button allows users to access meaning data associated to the search lemma, i.e. mainly definitions, as no other data are needed when users experience a reception problem that has to be solved quickly in order to ensure effective comprehension and reading.

- Production: clicking on the *production* (Producción) button allows users to access collocations and examples data associated to the lemma, which are most useful in production situations where explicit syntagmatic information is needed (syntactic constraints, conventional combinations of bases and collocates, etc.) in order to ensure effective written, specialised communication.

- Knowledge: clicking on the *knowledge* (Conocimiento) button allows users to access all data associated to the search lemma, including grammatical information, meaning and usage comments, lexical remarks, knowledge of international accounting standards or of national differences in national accounting systems etc. Common to situations 1, 2, and 3 described above, is the fact that the user is firmly guided in his search by the exact form of the search lemma, be it a single term unit or a multiword term unit. Users are then directed to a single data set consisting of different data categories, but no data exploration is needed as information needed is normally solely associated to the search lemma alone.

- Find a term is quite different. Clicking on the *find a term* (ENCONTRAR UN TÉRMINO) button will allow users to filter their search and access larger series of data sets (in the form of lists) and related data associated to terms and expressions closely connected to the search word. This option thus offers greatly extended search possibilities. The *find a term* button is particularly useful in what we call exploratory situations. These are situations in which users may be uncertain of the exact form of the lemma to be searched for, or who want to explore the data of a particular lemmatic field. They are thus offered the possibility to generate their own searches and search strategies by using Boolean operators. Users can then retrieve various data categories from the lexicographic database, e.g. they can retrieve lemmas, parts of collocations or examples, etc. Clicking on the retrieved data from the lists, they can then be redirected and access data that can suit their needs in different situations, typically in cognitive or communicative situations.

In other terms, buttons 1, 2 and 3 offer single-targeted search routes in which data presentation is customised according to the users specific data presentation needs, whereas button 4, *find a term*, offers multi-targeted, explorative search routes in which search routes are defined and customised by the user according to the users data exploration needs. In the following section, we will explain the working of the button and provide some examples of search situations and search options, and associated search results.

4 User-generated exploratory access mode and route: ENCONTRAR UN TÉRMINO in the Spanish Accounting Dictionary

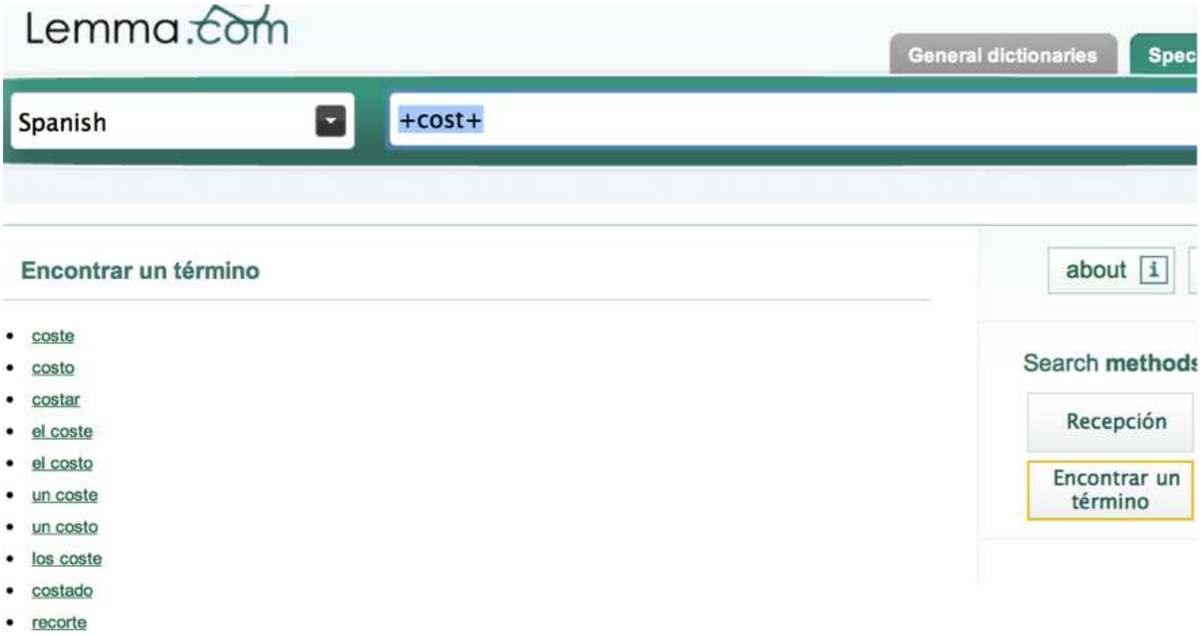
The Accounting Dictionaries are a set of specialized online dictionaries that are the result of a joint project involving teams from the Centre for Lexicography at Aarhus University in Denmark, and the International Centre for Lexicography at the University of Valladolid in Spain (see Fuertes-Olivera and Tarp, 2014 for a description). User's needs and usage situations led to the design of a dictionary project with a triadic structure: (a) a lexicographic database; (b) a user interface where one or more dictionaries are placed; (c) a search engine that mediates between the database and the user interface. The practical application of the philosophy underlying the abovementioned triadic structure has allowed us to convert the originally conceived polyfunctional dictionary into a Model T. Ford one, i.e. a dictionary whose articles and visualized lexicographic data are adapted to the various functions displayed by the dictionary, frequently assisted by different types of interactive options where the users may define themselves and the activity for which they need information (Tarp, 2011). This adaptation has occurred during the post-compilation phase, i.e. lexicographers put the dictionary at users' disposal, observe how it works, and check whether or not users are satisfied. For specialised lexicography, these tasks are inseparable from subjecting the specialised dictionary to a process of regular and continuous updating. Constant modification of language and facts characterise the ontological nature of subject fields, which demands the use of theoretical assumptions, methodologies and technologies that facilitate the process of constant updating that must characterise the design and compilation of specialised online dictionaries. Furthermore, users of the Danish and English set of the Accounting Dictionaries have emailed editors and expressed their interest on using search systems that would be based on user-generated search strategies, i.e. users initiated a search by following their own hints or ”intuitions”. Well-trained translators of specialised texts showed a lot of interest in the possibility of using such search systems as they are well-aware of the necessity of producing revisable translations - i.e. translations that can be easily corrected by subject field experts - when they have to translate texts with very limited knowledge - or no knowledge at all - of the subject field. A user-generated search strategy can be of help for these users as it can easily offer them more than one possibility, thus facilitating well-trained users' consultation process. In other words, we believe that the inclusion of user-generated search strategies is adequate for users with dictionary culture: these might easily start the process of consultation and discriminate among the hits retrieved without being impeded by the so-called Google effect that occurs when users retrieve many more data than needed. To sum up, we believe that users such as translators of specialised texts also need access modes and routes that facilitate their documentation, e.g. by facilitating the retrieval of all the lexicographic data that matches their queries no matter how ”perfect” their queries are. The above lexicographic philosophy resulted in the inclusion of the search button ENCONTRAR UN TÉRMINO (En: Find a term) in the Spanish Accounting Dictionary (Fuertes-Olivera et al., 2013). It is not present in the bilingual sets (English-Spanish/Spanish-English) as users can access the bilingual data from the Spanish string of words. For instance, supposing a user starts the search string +*cont*+, he or she will retrieve 10 clickable strings of data: ”con”, ”*coste*”, ”costo”, ”descontó”, ”descontar”, ”control”, ”recorte”, ”accionista”, ”al contado”, and descontada”. Clicking on any of these will retrieve all the dictionary articles where the searched string is. For example, clicking on ”descontó” (the Spanish past of the verb ”descontar”) retrieves the dictionary article for the verb ”descontar”, which has two meanings. From this dictionary article, our potential user can search ”descontar” by clicking on the button ”Frases y Expresiones” of the Spanish-English Accounting Dictionary (Fuertes-Olivera et al., 2014). The search will retrieve the dictionary articles of several (mostly unrelated) terms: **central bank**, **cash flow**, **interest and other income**, **notes receivable**, **interest rate**, **interest rate implicit in the lease**, **current cost**, **present value**, and **unguaranteed residual value**. In all of these dictionary articles, a form of ”descontar” is present and users can easily find it as it is highlighted. With this consultation process, users can gain different language and knowledge of the verb ”descontar”. To put it more simply, ENCONTRAR UN TÉRMINO allows users to try several search options, seven of which are shown below:

- +string of words + (Boolean sign plus string of words plus blank plus +)
- +string of words+ (Boolean sign plus string of words plus Boolean sign)
- +string of words + string of words (Boolean sign plus string of words plus blank plus + plus blank plus string of words)
- +string of words +string of words (Boolean sign plus string of words plus blank plus + plus string of words)
- +string of words -string of words (Boolean sign plus string of words + blank plus Boolean sign plus string of words)
- +string of words -string of words (Boolean sign plus string of words plus blank plus Boolean sign plus string of words)
- string of words OR string of words. (string of words plus blank plus OR plus blank plus string of words).
- Etc.

Let’s illustrate the working of the user-generated access strategy with two accounting concepts: *coste* (En: cost) and *gasto* (En: expense). The main difference in accounting between these two similar concepts is that *gasto* is part of external accounting and implies a decrease in equity, whereas *coste* is part of internal accounting and does not necessarily refer to decreases in equity. In the Accounting Dictionaries, *coste* and/or *gasto* are included in around 600 dictionary articles, typically as lemmas but also in other lexicographic data, e.g. in examples and collocations. Supposing a user is not sure about a particular meaning, spelling, or usage of *coste* and/or *gasto*, he or she can use the search button ENCONTRAR UN TÉRMINO, which initiates a search process whose ultimate goal must be to confirm or discard the intuition that prompted the search. For reasons of space, we cannot illustrate this access system in full. Four examples with four strings will suffice: (1) +cost+; (2) + cost +; (3) + cost-; (4) + cost OR *gasto*- :

Examples 1 to 4 show what users retrieve with ENCONTRAR UN TÉRMINO used with several search strings. Searching +cost+ (example 1) retrieves terms, grammar, and part of collocations: *coste*, *costo*, *costar*, *el coste*,*el costo*, *un coste*, *un costo*, *los costes*, *costado* and *recorte*.

Searching + *coste* + (example 2) retrieves additional data to the one shown in example 1: there are two new multi-word terms with grammar information: *el coste* fijo and *el coste neto* . Searching + *coste*- (example 3) retrieves new grammar data, i.e. the indefinite articles accompanying the terms *coste* and *costo*: *unos costes* and *unos costos*. Finally, searching + cost OR *gasto*- (example



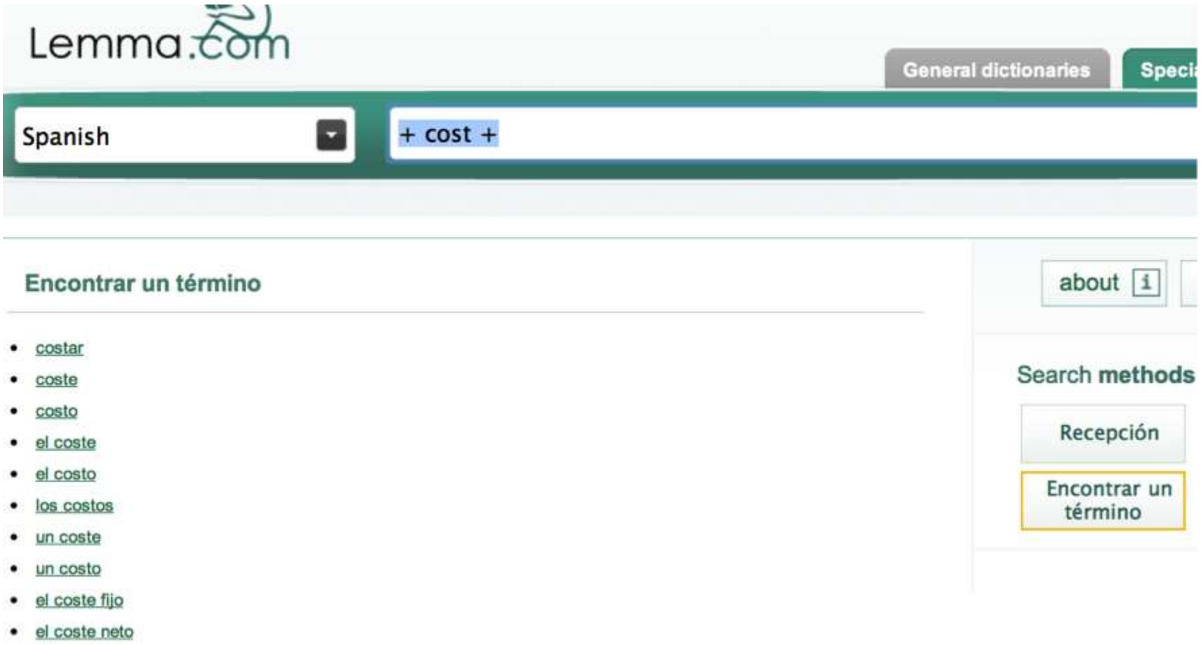
Spanish

+cost+

Encontrar un término

- coste
- costo
- costar
- el coste
- el costo
- un coste
- un costo
- los coste
- costado
- recorte

Example 1: Searching +cost+ with ENCONTRAR UN TÉRMINO



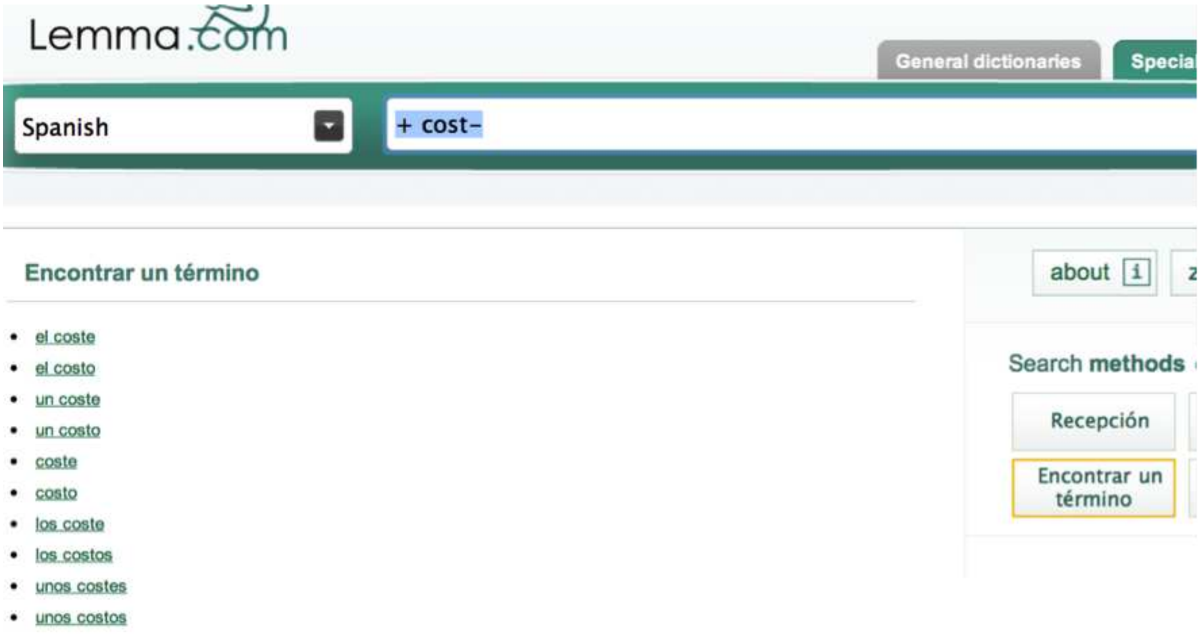
Spanish

+ cost +

Encontrar un término

- costar
- coste
- costo
- el coste
- el costo
- los costos
- un coste
- un costo
- el coste fijo
- el coste neto

Example 2: Searching + cost + with ENCONTRAR UN TÉRMINO

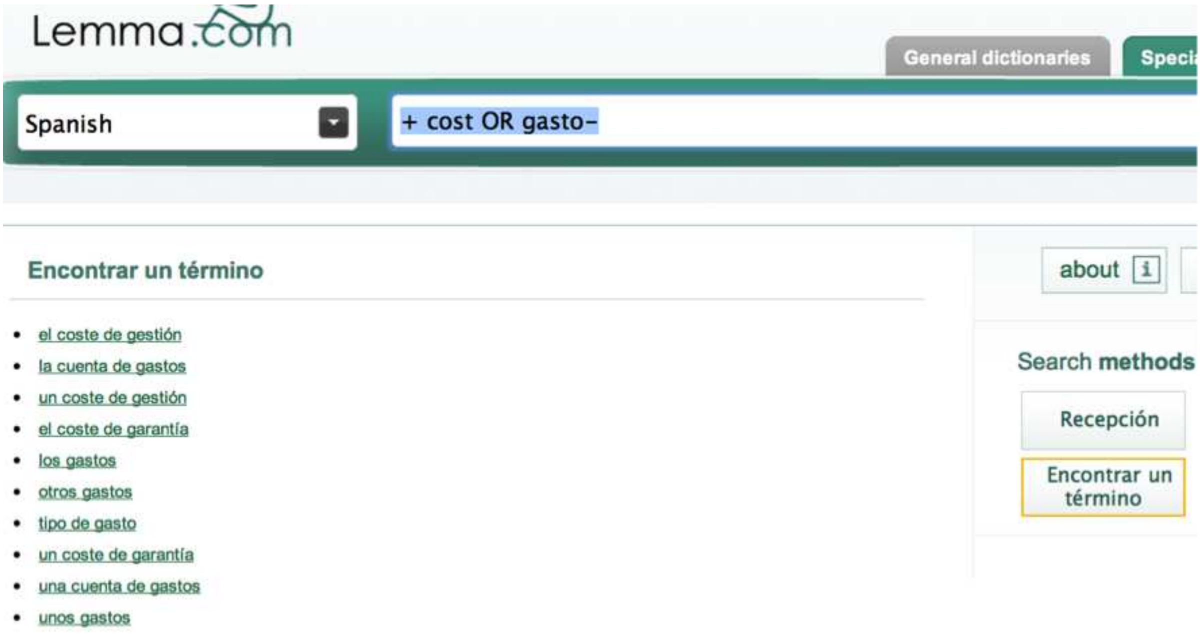


+ cost-

Encontrar un término

- el coste
- el costo
- un coste
- un costo
- coste
- costo
- los coste
- los costos
- unos costes
- unos costos

Example 3: Searching + *coste*- with ENCONTRAR UN TÉRMINO



+ cost OR gasto-

Encontrar un término

- el coste de gestión
- la cuenta de gastos
- un coste de gestión
- el coste de garantía
- los gastos
- otros gastos
- tipo de gasto
- un coste de garantía
- una cuenta de gastos
- unos gastos

Example 4: Searching + cost OR *gasto*- with ENCONTRAR UN TÉRMINO

4), retrieves different terms and part of collocations and examples, some of them with grammar data: *el coste de gestión*, *la cuenta de gastos*, *un coste de gestión*, *el coste de garantía*, *los gastos*, *otros gastos*, *tipo de gasto*, *un coste de garantía*, *una cuenta de gastos* and *unos gastos*. Below, we offer a more detailed description of the search possibilities the search button ENCONTRAR UN TÉRMINO initiates. Clicking on "el *coste* de gestión" (example 4 above) retrieves the dictionary article for *coste* de gestión (example 5)

coste de gestión `<un coste de gestión, el coste de gestión, unos costes de gestión, los costes de gestión>`
Definition
`Los costes de gestión son los costes en los que se incurren por decisiones de la dirección que responden a decisiones voluntarias, con poca o ninguna relación con la capacidad de producción o con la actividad. La publicidad, la investigación y desarrollo, y mantenimiento y mejoras son ejemplos de costes de gestión.`
Collocation
`calcular los costes de gestión`
`estimar los costes de gestión`
See also
`coste de administración (clickable hyperlink)`

Example 5: Dictionary entry in the Spanish Accounting Dictionary (Fuertes-Olivera et al., 2013).

From this dictionary article, users can search *coste de gestión* in the Spanish-English Accounting Dictionary (Fuertes-Olivera et al., 2014). They have four search buttons at their disposal: *Recepción*, *Traducción*, *Conocimiento* and *Frases y Expresiones*:

- Recepción: clicking on the *reception* (Recepión) button allows users to access the definition of *coste de gestión* (see example 5, above) and the English equivalent: **management cost**.

- Traducción: clicking on the *translation* (Traducción) button allows users to access the definition and equivalent of *coste de gestión* plus the English translation of the Spanish collocation (example 5 above), and five hyperlinked synonyms of **management cost** with language label: **discretionary cost**, **programmed cost**, **managed cost**, **policy cost**, and **management fixed cost**.

- Conocimiento: clicking on the *knowledge* (Conocimiento) button allows users to access the data obtained under Reception and Translation plus the inflections of **management cost** and access to open linked data.

- Frases y Expresiones: clicking on the *phrases and expressions* (Frases y Expresiones) button allows users to access two dictionary articles where *coste de gestión* is part of a Spanish phrase or example. These are translated into English and users can easily uncover (possible) similarities and differences among them.

Users can also search in the English-Spanish Accounting Dictionary (Fuertes-Olivera et al., 2012), e.g. by searching **management cost** (i.e. the English equivalent of *coste de gestión*) or any of the English synonyms, antonyms, etc. previously retrieved. For instance, searching **management cost** with the Recepción button retrieves its English definition and Spanish equivalent as well as a Spanish definition of the English term. This is also a novelty of these dictionaries: it was included as we found out that some Spanish users did not have a high command of English accounting. To sum up, the search button ENCONTRAR UN TÉRMINO is used for initiating user-generated results. All of them can be used when users are uncertain or unsure of what they are searching. Clicking on each of them will expand users exploratory searches, thus offering users exploratory options for confirming or rejecting their intuitions regarding their needs in communicative situations, cognitive situations, or operative situations.

5 Conclusion

The concept of access in e-lexicography should be seen as a continuum. At the one end, access is regarded as a static structure from the perspective of the data. Access can be defined as the outer realisation of inner data structures and is realized through data structure determined access modes and search options. At the other end, access is treated as a dynamic process from the perspective of the user and is customized according to the specific user profile, the user situation, and the user reference skills. Data structuring is related and adapted to these situations. In this article, we have made the case for the development of user-situation and user-profile driven access modes and shown how this works in the Spanish Accounting Dictionaries, in which the function button ENCONTRAR UN TÉRMINO provides access in case of failure from the user to initiate a precise search because of the lack of matching search strings. Series of user-generated analogical, multi-targeted associative search strings make it then possible to recover the needed information (term unit), or to explore a certain data field to satisfy systematic knowledge needs, for instance in a learning situation. This is yet a new step forward in the development of modern, functional e-lexicography, as it establishes alternative, user-situation driven exploratory search and access modes which transfer the famous words of Pablo Picasso - ”I do not search, I find” - into a lexicographic context: The incapacity to search yielding the incapacity to find. The approach described in this paper defends that we need a holistic view of the dictionary. Such view implies that the different lexicographic data included in the lexicographic database can be used in any of the above situations, even for performing tasks that were not originally planned, e.g. an example can be extracted and quoted for crafting much more precise descriptions of specialized concepts.

6 Acknowledgments

Thanks are due to the Ministerio de Economía y Competitividad and to the Junta de Castilla y León for financial support (Grant FF12011-22885 and VA 067A12-1); also to the three anonymous reviewers whose comments have greatly improved this paper.

References

DiCoube 1 october 2013 *http://olst.ling.umontreal.ca/dicouebe/*.

G-M De Schryver 2003. Lexicographer’s dreams in the electronic-dictionary age. *International Journal of Lexicography* , 16(2): 143-199.

Henning Bergenholtz and Inger Bergenholtz. 2011. A dictionary is a tool, a good dictionary is a monofunctional tool. In Pedro A. Fuertes-Olivera and Henning Bergenholtz (Eds.), 187-207.

Henning Bergenholtz, and Rufus H Gouws. 2010. A new perspective on the access process. *Hermes Journal of Language and Communication Studies* 44, 103127.

Herbert E. Wiegand. 2008. Zugriffsstrukturen in Printwrterbchern. Ein zusammenfassender Beitrag zu einem zentralen Ausschnitt einer Theorie der Wrterbuchform. *Lexicographica* 24, 209-315.

Marie-Claude LHomme, B. Robichaud and Patrick Leroyer 2012. Encoding collocations in DiCoInfo: from formal to user-friendly representations. In S. Granger and M. Paquot (Eds), 211-236.

Nerina Bosman. 2012. Ease of access in the new Afrikaans-Nederlands/Nederlands-Afrikaans dictionary (2011) in the Dutch L2 classroom a case study. In Ruth E. Vatvedt Fjeld and Julie Matilde Torjusen (Eds.) (2012): *Proceedings of the 15th EURALEX International Congress.* Oslo: Institutt for lingvistiske og nordiske studier. 947- 954.

Pedro A. Fuertes-Olivera and Henning Bergenholtz (Eds.) 2011. *e-Lexicography: The Internet, Digital Initiatives and Lexicography*. London and New York: Continuum.

Pedro A. Fuertes-Olivera and Sven Tarp 2014. *Theory and Practice of Specialised Online Dictionaries: Lexicography versus Terminography*. Berlin, New York: De Gruyter.

Pedro A. Fuertes-Olivera, Henning Bergenholtz, Sandro Nielsen, Pablo Gordo Gómez, Lise Mourier, Marta Niño Amo, Ángel de los Ríos Rodicio, Ángeles Sastre Ruano, Sven Tarp and Marisol Velasco Sacristán. 2012. *Diccionario Inglés-Español de Contabilidad.* Base de Datos y Diseño: R. Almind and J. Skovgård Nielsen. Hamburg: Lemma.com. http://lemma.com/

Pedro A. Fuertes-Olivera, Henning Bergenholtz, Sandro Nielsen, Pablo Gordo Gómez, Marta Niño Amo, Ángel de los Ríos Rodicio, Ángeles Sastre Ruano, Sven Tarp and Marisol Velasco Sacristán. 2013. *Diccionario Español de Contabilidad.* Base de Datos y Diseño: R. Almind and J. Skovgård Nielsen. Hamburg: Lemma.com. http://lemma.com/

Pedro A. Fuertes-Olivera, Henning Bergenholtz, Sandro Nielsen, Pablo Gordo Gómez, Lise Mourier, Marta Niño Amo, Ángel de los Ríos Rodicio, Ángeles Sastre Ruano, Sven Tarp and Marisol Velasco Sacristán. 2014. *Diccionario Español-Inglés de Contabilidad*. Base de Datos y Diseño: R. Almind and J. Skovgård Nielsen. Hamburg: Lemma.com. http://lemma.com/

Rufus H. Gouws. 2001. The use of an improved access structure in dictionaries. *Lexikos* 11: 101111.

Rufus H. Gouws. 2013. Why research regarding dictionary structures remains important. In Kwary, Deny A, Nur Wulan and Lilla Musyahda (Eds.) *Lexicography and Dictionaries in the Information Age. Proceedings abstracts from the 8th Asialex International Conference*, 347.

Serge Verlinde and G. Peeter 2012. Data access revisited: The interactive Language Toolbox. In S. Granger and M. Paquot (Eds), *Electronic Lexicography*. Oxford: Oxford University Press, 147-162

Serge Verlinde, Patrick Leroyer and Jean Binon 2010. Search and you will find. From stand-alone lexicographic tools to user driven task and problem-oriented multifunctional leximats. *International Journal of Lexicography* 23, (1), 1-17.

Sven Tarp. 2008. *Lexicography in the Borderland between Knowledge and Non-knowledge. General Lexicographical Theory with Particular Focus on Learners Lexicography*. Tübingen: Niemeyer.

Sven Tarp. 2011. Lexicographical and other e-tools for consultation purposes: Towards the individualization of needs satisfaction. In Pedro A. Fuertes-Oliver and H. Bergenholtz, (eds.), 54-70.

Sylviane Granger. 2012. Introduction: Electronic lexicography from challenge to opportunity, in S. Granger and M. Paquot (eds), 1-11.

Sylviane Granger and Magali Paquot. 2010. The Louvain EAP Dictionary (LEAD). In A. Dykstra and T. Schoonheim (Eds.), *Proceedings of the XIV Euralex International Congress*. Fryske Akademy, 321-326.

Sylviane Granger and Magali Paquot. 2012. *Electronic Lexicography*. Oxford: Oxford University Press.

Trésor de la langue français informatisé 1. october 2013. http://atilf.atilf.fr/tlf.htm. (TLFi)

Influence of Information Structure on the Salience of Opinions

Josef Ruppenhofer, Daniela Schneevogt
Marienburger Platz 22
Dept. of Information Science and
Natural Language Processing
Hildesheim University
31141 Hildesheim
ruppenho@uni-hildesheim.de, d.schneevogt@gmx.de

Abstract

We study the influence of information structure on the salience of subjective expressions for human readers. Using an online survey tool, we conducted an experiment in which we asked users to rate main and relative clauses that contained either a single positive or negative or a neutral adjective. The statistical analysis of the data shows that subjective expressions are more prominent in main clauses where they are asserted than in relative clauses where they are presupposed. A corpus study suggests that speakers are sensitive to this differential salience in their production of subjective expressions.

1 Introduction

It is well known that subjectivity or sentiment is a complex phenomenon. Not only do individual subjective expressions such as *handsome, beautiful, ugly* differ in intensity and polarity, but also external factors impinge on subjective expressions, modulating their intensity and/or polarity. Accordingly, the best studied questions in this area of research include methods for assigning prior polarity (Hatzivassiloglou and McKeown, 1997) and for recognizing polarity in context (Wilson et al., 2005; Moilanen and Pulman, 2007) ; methods for assigning out-of-context (or: prior) polar intensity scores to adjectives (Sheinman and Tokunaga, 2009; de Melo and Bansal, 2013; Ruppenhofer et al., 2014) and methods for modeling the effects of degree modification (Taboada et al., 2011). Negation and modality have been studied by (Benamara et al., 2012; Wiegand et al., 2010).

Greene and Resnik (2009) look at the influence of what they call syntactic framing on subjectivity, namely questions of causal responsibility, affectedness and salience of new states resulting from events.

What has, to our knowledge, not been investigated at all is the influence of `information structure` on the salience of subjective expressions.[1] Information structure (Lambrecht, 1996) is that part of linguistics that concerns itself with the relation of sentence form to the linguistic and extra-linguistic contexts in which sentences are used to convey propositional information. Importantly, sentences may contain the same propositional information, yet differ in terms of information structure, as shown by examples (1-2).

(1) Peter, who is a really **sweet** guy, lives next door.

(2) Peter, who lives next door, is a really **sweet** guy.

Both sentences convey to the hearer new information about a known topic, namely the referent of *Peter*. Importantly, in each sentence the information in the relative clause is presupposed, that is, presented as if it already is part of the so-called common ground between speaker and hearer. By contrast, the main clause predicate is part of the focus, i.e. it is assumed to provide new information about the topic. The question that we investigate experimentally in this study is whether subjective expressions that are asserted as part of the focus are perceived more saliently than subjective expressions that are embedded in presupposed parts of sentences. In other words, are speakers likely to rate (2) as more subjective than (1) due to *sweet* being part of the focus in the former but not the latter?

Our experiments show that there are clear and stable differences between sentences that contain all the same lexical material and all the same propositions but which structure these propositions in different ways.

This paper is structured as follows. Section 2 presents related work. In section 3, we lay out our experimental design. The results of our data collection are analyzed and discussed in section 4. A supplementary corpus study is presented in section 5 and we conclude in section 6.

2 Related Work

One type of related work looks at how evaluative text is structured and where subjective expressions may be found. Bieler et al. (2007) develop a system for analyzing movie reviews into formal (e.g. author, genre, legal-note etc.) and functional parts (describe vs comment). Degaetano-Ortlieb et al. (2012) study the type and distribution of sentiment expressions occurring in different sections of texts. Their purpose is to study the similarities and differences between related but different scientific disciplines.

Wang et al. (2012) seek to incorporate information about discourse relations such as Contrast, Cause, etc. into the task of classifying reviews. The knowledge they use is pragmatic in nature, but it is orthogonal to information structure. Similarly, Mukherjee and Bhattacharyya (2012) use discourse relation information gleaned without full discourse parsing to improve tweet polarity classification.

Heerschop et al. (2011) use Rhetorical Structure Theory to divide a text into important and less important text spans, subsequently using this to improve the performance of a sentiment classifier. The discourse relations of RST concern the propositional content of pairs of propositions. By contrast, information structural notions such as topic, focus, presupposition and assertion are properties of individual clauses as they concern the relation of a proposition to the knowledges state of the speaker and hearer about the content of that proposition. The effects of information structure are thus distinct from the effects of the kind of discourse structures that RST covers.

Wiebe and Riloff (2005) classify sentences as subjective and objective based on extraction patterns they learn. This work operates on the sentence level but it looks at the detection of subjectivity rather than its salience. It uses extraction patterns but does not model where they occur within the sentences.

Our work can also be compared to work on determining the intensity of subjective expressions, some of which we referenced in the introduction. We work on contextual effects. However, they are effects on sentence subjectivity rather than the intensity of subjective expressions, and we study information structure as an influence rather than the effect of degree modification or negation.

Finally, Kabadjov et al. (2011) investigate the suitability of very highly positive and negative sentences for the purposes of text summarization. They find that sentences found useful for summarization are no different in terms of subjectivity intensity than sentences that were not found similarly useful. While this work looks for salient sentences that are useful for the summaries, it does not take into account how prominent subjective expressions are within the sentences that are either salient for summary purposes or not. Thus, the issue that interests us is not addressed.

3 Experimental design

3.1 Selecting the clause types

The purpose of the study is to test the influence of information structure on the salience of subjective expressions. Information structure can be signaled through various linguistic means, including intonational and lexical means. Since we were looking to perform a self-paced reading experiment and wanted to avoid possible confounding influences introduced by lexical cues to information structure, we decided to focus on different sentence types as signals of particular information structures.

We specifically contrast main clauses and relative clauses. We chose these two clause types because they were among the most common types in a 250 sentence random sample taken from the Huge German Corpus (HGC; Fitschen 2004), which contains around 204M words of newspaper text. The focus on these clause types is meant to avoid any distortion of the results through low-frequency structures. Note that we work only with asserted indicative mood sentences so as to exclude modality as a variable influencing our results.

We did not use complement clauses in our study, although they were among the most frequent clause types in our HGC sample. The reasons for this choice are the following. First, constructing complement clause stimuli would mean using different/additional lexical material, whereas main and relative clause pairs can be constructed so they contain the same lexical material (cf. 3–6). Second, since complement clauses vary by the type of embedding predicate – e.g. whether it is factive (e.g. *know*) or not (e.g. *claim*) – one would need to control for the various subtypes of complement clauses, which would increase the amount of items to be rated and thus the length of the survey. Third, complement clauses with predicates of cognition or communication present the additional difficulty that the source of the subjective expression is an attributed one rather than the sentence's author, whereas for the relative and main clause data the source is always the implicit author. For this first study of information structural influences on sentence subjectivity, it was thus easier, both in terms of stimuli construction and subsequent analysis, not to include complement clauses.

3.2 Selecting the adjectives

Adjectives are a well-studied lexical class clearly associated with evaluation and subjectivity (e.g. (Bruce and Wiebe, 1999; Wiebe, 2000)). They are often the largest class in polarity lexicons (e.g. SoCAL (Taboada et al., 2011) or the Pittsburgh subjectivity clues (Wilson et al., 2005)), and opinion mining systems that limit themselves to use only words of certain parts of speech as features will tend to include adjectives. Accordingly, we decided to focus our experiments on the salience of adjectives in various configurations.

We created a pool of candidate adjectives by merging the adjectives contained in two German polarity dictionaries, SentiWS (Remus et al., 2010) and German Political Clues (Waltinger, 2010). Since we wanted to control for the inherent polarity strength of the adjectives, we decided to select the adjectives in pairs such that both refer to the same semantic scale but one has greater prior intensity than the other. For example, *spannend* 'fascinating' is more positive than *interessant* 'interesting' but both refer to the scale of mental stimulation. Further, in order to control for the influence of word familiarity on the results, we evaluated the frequency of our candidate adjective pairs in two ways. First, we checked against the dlexDB psycholinguistic database (Heister et al., 2011) to make sure they had about the same frequency of occurrence there. And second, we checked that both adjectives were within the same frequency band for adjectives in the HGC corpus. Finally, to avoid results due to lexical idiosyncrasies, we constructed 5 positive and 5 negative pairs of adjectives. We also chose 8 different control adjectives that we expected to be neutral given that they were not listed in the two German polarity lexicons we used.[2] The chosen adjectives are listed in Table 1. The polar adjective pairs are indicated by way of horizontal lines.

3.3 Stimuli

We illustrate our experimental stimuli with the set of examples in (3)–(7). The codes preceding the examples are constructed as two-letter combinations as follows: R = relative clause, M = main clause, C = complement clause; W = weak prior intensity, S = strong prior intensity; N = neutral. In the interest of keeping the length of our survey at around 20 minutes, we decided not to use RN and MN stimuli, since we are most interested in the behavior of the non-neutral adjectives across conditions. We included the neutral sentences as filler material. Note that sentences with polar adjectives were constructed so they do not contain any other subjective expressions.

(3) [RS] Ihr Bruder, der zu allen **unfreundlich**

[2] This expectation was not fully borne out in the experiments, as will be discussed in section 4.3.

Adjective	Gloss	Polarity SentiWS	Polarity elicited
ungeschickt	clumsy	-0.6087	-34.2576
doof	daft	-0.1562	-50.8333
unfreundlich	unfriendly	-0.3407	-62.3485
unhöflich	impolite	-0.0048	-60.2727
eintönig	humdrum	-0.0378	-42.6212
langweilig	boring	-0.0228	-49.6970
entsetzlich	appalling	-0.477	-71.1509
scheußlich	hideous	-0.1834	-75.8868
dumm	dumb	-0.5901	-61.6038
blöd	stupid	-0.1593	-55.4528
hübsch	pretty	0.4629	56.8636
wunderschön	gorgeous	0.7048	78.0152
grandios	grand	0.1843	80.1515
großartig	great	0.4606	78.7879
freundlich	friendly	0.6022	65.2830
nett	nice	0.1405	49.2642
intelligent	intelligent	0.1238	65.6038
klug	smart	0.3532	64.5094
interessant	interesting	0.2488	51.0377
spannend	fascinating	0.7165	50.6415
geheim	secret	neutral	-2.3940
geläufig	prevalent	neutral	4.2576
wissenschaftlich	scientific	neutral	22.4848
gängig	common	neutral	5.3788
objektiv	objective	neutral	19.4340
sachlich	matter-of-fact	neutral	14.8491
häufig	frequent	neutral	2.4340
dünn	thin	neutral	2.2453

Table 1: Adjectives used in human elicitation

ist, wohnt in Berlin. 'Her brother, who is unfriendly to everybody, lives in Berlin.'

(4) [RW] Ihr Bruder, der zu allen **unhöflich** ist, wohnt in Berlin. 'Her brother, who is impolite to everybody, lives in Berlin.'

(5) [MS] Ihr Bruder, der in Berlin wohnt, ist zu allen **unfreundlich**. 'Her brother, who lives in Berlin, is unfriendly to everybody.'

(6) [MW] Ihr Bruder, der in Berlin wohnt, ist zu allen **unhöflich**. 'Her brother, who lives in Berlin, is impolite to everybody.'

(7) [CN] Sie erzählt, dass ihr Bruder in Berlin wohnt. 'She says that her brother lives in Berlin'.

3.4 Task

Our contained main task and distractor task items. The *main task* consisted in rating sentences on a 7 point scale ranging from strongly negative (-3) via neutral $\emptyset$ to strongly positive (+3). The survey was administered to the subjects via the open-source LimeSurvey[3] software. The sentences were organized in groups (such as (3)–(7)) and we randomized both the ordering of the sentences within a given group as well as the ordering of the groups within the survey. This was done to control biases that might arise due to learning, habituation or motivational effects in the course of answering the survey questions. The sentences were displayed singly on the screen and subjects had to press a continue-button to go on to the next item. They could not return to a previous item. Though displaying multiple items per screen would have made completion of the survey faster, grouped display is likely to result in respondents viewing the items as a set and increasing the correlation among them beyond what is due to the stimuli themselves (cf. discussion by Couper et al. (2001)).

In the *distractor task*, subjects were asked to rate the polarity and intensity of adjectives that appeared in the main task. The scale for the adjective intensity rating ranged from -100 to +100. The purpose of the distractor task was to a) prevent participants from focusing consciously too much on the sentence rating task; b) check on the information available from the polarity lexica ; and c) use values for prior adjective intensity that actually fit our population of subjects.

Since we did not have funds to pay our subjects, we tried to keep the duration of the survey within a 20-25 minute time window. Because a pre-test had shown that our initial design took longer than 20 minutes for many participants, we divided our main and distractor task items across two non-overlapping versions A and B for the actual run of the survey. Version A included 3 positive adjective pairs and 2 negative pairs, Version B covered 2 positive adjective pairs and 3 negative ones. Using two complementary surveys allowed us to elicit ratings for the adjectives in the sentence rating task of survey A in the distractor task of survey B, and vice versa.

Besides keeping the survey short in the interest of a higher completion rate, subjects also were shown a progress bar on the screen so they would not abandon surveys when they were already close to completion. In addition, some extrinsic motivation to complete the survey was provided by the chance for participants to win one of three vouchers for use at a large online merchant.

3.5 Subjects

Our subjects mainly are undergraduate students at two German universities. These participants were recruited from the friends and acquaintances of the au-

[3] Available at www.limesurvey.com/.

thors and also via colleagues and referrals from participants. We collected meta-data (gender, age, German proficiency, place of birth and place of residence, occupation) on our subjects but do not use them in our analysis below. Overall, 130 subjects completed the survey altogether. 72 completed survey A and 58 survey B. Note that subjects were assigned randomly to the two versions of the survey.

4 Results and discussion

4.1 Data cleanup

In order to eliminate the influence of participants who might have had difficulty with the task, we proceeded as follows. For each participant we calculated the average of their kappa values with every other subject. Based on the mean and standard deviation of these average kappa values, we excluded those participants that lay outside 2 SDs of the average kappa of the average annotator. In addition, to err on the side of caution, we also decided to exclude the ratings of participants for whom German is not their native language. As a result, we retained the data of 110 participants, 64 from version A and 46 from version B.

4.2 Analysis

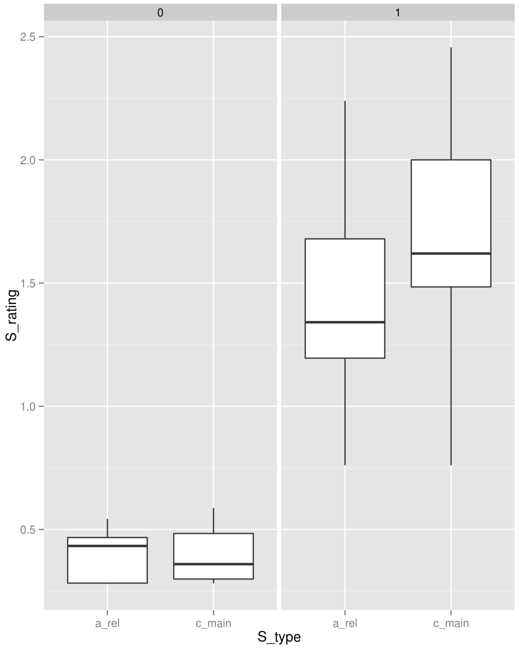


Figure 1: Box plots: absolute sentence rating by sentence type for non-polar (left) and polar (right) adjectives

Figure 1 shows two boxplots for the absolute adjective ratings grouped by sentence type, with the results for non-polar sentences on the left and for polar ones on the right.[4] These results seem in line with our expectations: for non-polar adjectives there are no significant differences. For polar ones, there are differences among the main clauses and relative clauses.

Figure 2 shows plots of average word intensity ratings against average absolute sentence ratings with one dot representing each adjective. The plots show that, as expected, higher adjective intensity goes with greater sentence subjectivity in both relative and main clauses. The comparison of the slopes in the graphs also suggests that the relationship is somewhat stronger in main clauses than in relative clauses.

Since each adjective has a unique absolute prior intensity and each adjective appears with both sentence types, we can draw an imaginary vertical line through an adjective's intensity value on the x-axis and see how the adjective's average sentence rating in main clauses compares to its rating in relative claues. Doing this shows that the average sentence ratings in the main clauses are greater than the ratings in the relative clauses for 19 of the 20 polar adjectives. The remaining case is the adjective *ungeschickt* 'clumsy' in the lower left corner of the graph. For this item, which was rated as the least intense adjective among the ones that we had taken to be polar based on the polarity lexicons, the average scores are identical and the two points lie atop each other.

If we compare the absolute values of individual judges' scores for main and relative clause instances with polar adjectives, we find the pattern shown in Table 2. The results for non-polar adjectives are shown in Table 3.

	0	1	2	3
0	36	45	28	5
1	23	259	191	30
2	3	48	252	59
3	1	4	14	102

Table 2: Confusion matrix for main (columns) and relative clauses (rows) with polar adjectives

For polar adjectives, if a judge does not rate the main and relative clause instances the same, they are 3 times as likely to rate the main clause instance as the more intense type than the relative clause instance. Compare the sum of the cells above the diagonal in Table 2 to the sum of the cells below the diagonal. A chi-square test performed on the confusion matrix in Table 2 is highly significant (X-squared = 733.9092, df = 9, p-value $<$ 2.2e-16).

For non-polar adjectives, the likelihood that main and relative clause instances will be rated the same is

[4] The black dots in the plots represent outliers.

highest, too. However, if the two are not rated the same, it seems the relative order is more or less random: in half the cases, the relative clause instance was rated as more strongly subjective, in the other half the main clause instance. A chi-square test on the groupings displayed in Table 4 shows that the polar adjectives and the non-polar ones differ significantly (X-squared = 62.1251, df = 2, p-value = 3.234e-14).

	0	1	2	3
0	249	40	8	1
1	48	55	10	0
2	7	8	9	1
3	0	0	3	1

Table 3: Confusion matrix for main (columns) and relative clauses (rows) with non-polar adjectives

	polar	non-polar	Total
M>R	358	60	418
M=R	649	314	963
R<M	93	66	159
Total	1100	440	1540

Table 4: Relative magnitude of main (M) vs. relative (R) clauses with the same adjective, both for polar and non-polar adjectives

To study the relative influence of sentence type and adjective intensity, we fit a cumulative link mixed model to the data (Agresti, 2002). We use clmm2 from R's ordinal package for this purpose (Christensen, 2011).

Our dependent variable is the absolute sentence subjectivity rating. We have two independent variables. The first of these is sentence type, that is, relative clauses versus main clause. The second is adjective intensity. Both these variables are treated as ordinal data. Sentence type has two levels, with class 0 corresponding to relative clauses and class 1 to main clauses, where we expect greater salience of predicates. Adjective intensity is treated as an ordinal variable with four levels by assigning class 0 to adjectives with scores in the range from 0-25, class 1 to adjectives with scores in the range from 26-50, etc.[5] We assume the rater effects are independent and identically distributed random variables.

For the maximum likelihood estimates of the parameters we use the adaptive Gauss-Hermite quadrature method to compute the likelihood function. We use the default setting of 10 quadrature nodes. Significance of model terms was assessed using likelihood ratio tests (a = 0.05), and models were compared with the Akaike Information Criterion (AIC). The condition number of the Hessian was used to assess model fit.

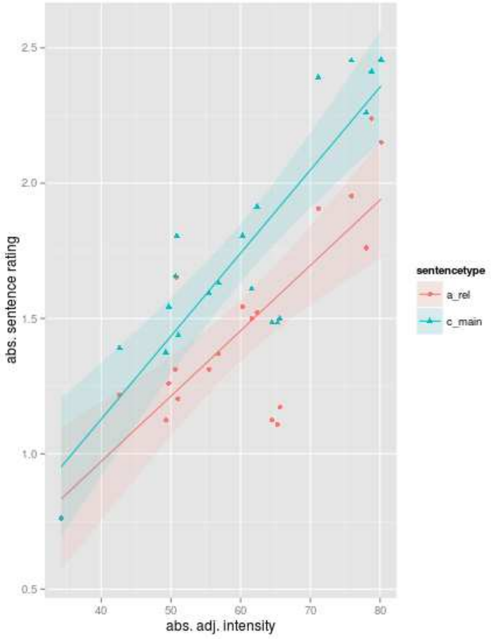


Figure 2: Average absolute sentence rating by average absolute adjective intensity in relative (red dots) and main (green triangles) clauses

High condition numbers correspond to less well defined models that could be simplified, or models where possibly some parameters are not identifiable. As can be seen from Figure (3), in our case the condition number of the Hessian (61.46202) does not indicate a problem with the model. Figure (3) shows that the coefficients for sentence type (0.5967) and adjective intensity (1.7088) are positive. This indicates that both a sentence type with greater predicate salience and greater intrinsic adjective intensity make a higher sentence subjectivty rating more likely. Additional likelihood ratio tests using the anova method show that sentence type and adjective intensity class are significant terms. The same is true of the variance parameter, rater.

4.3 Discussion

To sum up the analysis so far, we have seen that for our stimuli it seems to be the case that absolute sentence subjectivity ratings depend on sentence type, not only on adjective intensity. Given that the sentences we experiment with contain the same lexical material as well as the same structure, we may conclude that it is the positioning of the subjective expressions in

[5] We get very similar results even if we change the class boundaries.

```
clmm2(location = abssentencerating ~
sentencetype + adjintclass ,
random = rater , data = nudat ,
Hess = TRUE, nAGQ = 10)

Random effects :
                  Var    Std.Dev
rater 0.716407 0.8464083

Location coefficients :
                Estimate Std. Error z value Pr(>|z|)
sentencetype   0.5967    0.0723       8.2504 < 2.22e−16
adjintclass    1.7088    0.0471      36.2611 < 2.22e−16

No scale coefficients

Threshold coefficients :
    Estimate Std. Error z value
0|1   0.9486    0.1123      8.4468
1|2   3.5101    0.1319     26.6095
2|3   6.0575    0.1606     37.7232

log−likelihood : −3071.166
AIC: 6154.332
Condition number of Hessian: 61.46202
```

Figure 3: Fitting a cumulative link mixed model to the data

either the focal main clause or the presupposed relative clause that is responsible for the observed differences. This conclusion is strengthened by the fact that we used multiple pairs of adjectives from different semantic fields and matched the adjectives for frequency.

Through our data elicitation, we found that some of the information in the polarity and intensity dictionaries we used to select our adjectives did not match with the results of our elicitation. For instance, for the polar adjectives, Spearman's rank correlation between the elicited ratings and the information in SentiWS was 0.6782 (cf. Table 1). In addition, 3 of the adjectives that we expected to be objective based on the polarity lexicons behaved close to polar adjectives: *objektiv* 'objective', *wissenschaftlich* 'scientific' and *sachlich* 'matter-of-fact'. In part, this may be due to the background of our subjects: as college students they are taught to value objectivity over subjectivity. Conversely, the item *ungeschickt* 'clumsy' seems not to have been perceived as polar by most subjects, based on the evidence of the sentence subjectivity ratings, even though it is quite strongly polar for SentiWS and moderately so in our own elicitations. The lesson we take away from this is that in future experiments, we should first collect the intensity ratings ourselves before we try to construct pairs with specified differences in intensity.

5 Corpus study

In the rating survey we elicited subjects' *perception* of the intensity of sentences differing only in terms of information structure. We interpreted the results as showing that sentence type influences perceived intensity. However, since the experimental situation is an artificial one–with constructed stimuli and a lack of context–we are interested in complementary evidence that would show that people *use* subjective and objective adjectives in a way that reflects the perceptions we elicited. Accordingly, we performed a corpus study to test the following hypothesis: because main-clause use ensures greater salience of the expression, if speakers want to express opinions with subjective adjectives, they will use them as main clause predicates more often than they would objective adjectives, which do not (directly) serve to express opinions.

We randomly selected 9 adjectives from our pool, 3 each of the negative, positive and objective sets. For each adjective we collected 100 randomly chosen predicative uses in finite clauses from a corpus of German Amazon product reviews (Prettenhofer and Stein, 2010) and classified them as to clause type. Note that we extracted only corpus instances whose word form matched an adjective's invariant predicative form in the positive degree. That is, for e.g. *dumm* 'dumb' we only looked for the word form *dumm* but not for *dümmer*. We extracted the instances from a corpus of reviews so that we could assume that the adjectives are used in a context where the authors generally intend to convey opinions.

We performed the classification manually so as to avoid errors due to erroneous POS-tagging or parsing. In Table 5 we present the results.[6]

	main	relative	other
dumm	73	8	19
entsetzlich	56	1	1
unfreundlich	27	5	9
hübsch	69	4	27
spannend	88	1	11
grandios	97	2	1
wissenschaftlich	79	7	14
geheim	50	21	29
geläufig	59	30	11

Table 5: Main and relative clause occurrences per 100 predicative uses

We can aggregate the numbers for the positive, negative and objective adjectives, as shown in Table 6, and perform a χ^2-test on it. The difference in the distribution of the different types of adjectives across the clause types is highly significant (X-squared = 58.7103, df = 4, p-value = 5.413e-12). As the expected numbers in parentheses show, there are too few instances of relative clause use for the negative and

[6] The "other" category includes e.g. uses in complement clauses, subordinate clauses, etc. Note that *entsetzlich* appalling' and *unfreundlich* 'unfriendly' have fewer than 100 predicative uses in the data we use.

	main	relative	other
negative	156 (149)	14 (20)	29 (30)
positive	254 (225)	7 (30)	39 (45)
objective	188 (225)	58 (30)	54 (45)

Table 6: Aggregate results (expected numbers shown in parentheses)

positive adjectives in the observed sample, while there are too many relative clause uses for the objective adjectives. With respect to main clause uses, objective adjectives do not have enough of them, while especially positive adjectives have more of them than expected. This is also the case for negative adjectives, but less so.

Thus, although the set of adjectives analyzed is small, the results generally support the original hypothesis: for subjective adjectives, placement in main versus relative clauses matters much more than for objective adjectives. In line with that, subjective adjectives are used in relative clauses much less often than objective adjectives.

As shown by the counts in Table 5, *wissenschaftlich* 'scientific' behaves exceptionally. For the polarity dictionaries, this is an objective adjective. However, in our human data elicitation we found that it behaves much like a polar adjective. And this is also what we find here, as can be seen in example (8) from our data:

(8) Dennoch sind die Beispiele und Erklärung esoterisch und nicht **wissenschaftlich**.
'Nonetheless, the examples and the explanation are esoteric and not scientific'.

If we had treated *wissenschaftlich* as a non-objective, positive adjective (as indicated by the dashed line in Table 5), the results of the χ^2-test would have come out even more extreme than they have. However, looking at the corpus data shows that it is not clear that *wissenschaftlich* is inherently positive or negative. Besides frequent uses such as (8), one finds others where *wissenschaftlich* is used negatively.

(9) Das Buch ist natürlich recht **wissenschaftlich** und daher dann und wann vielleicht etwas trocken .
'However, the book is quite scientific and therefore maybe a bit dry every now and then.'

Given the existence of both uses like (8) and (9), it seems correct to say that *wissenschaftlich* is inherently an objective adjective. However, when it is used in context to convey or imply evaluation, it behaves distributionally like an inherently subjective adjective, occurring more often in main clauses than expected and less often in relative clauses.

6 Conclusion

In this paper, we presented the first study showing that in addition to degree modification, negation and modality, information structure has an influence of its own on the salience of subjective expressions. We probed this influence through an online experiment in which we had subjects rate controlled stimuli differing only in information structure. In addition, we performed a corpus study whose results indicate that the differing salience of subjective expressions that was found in the rating experiment also guides people's production of subjective expressions, at least in a context that is geared towards the expression of opinions.

In future work, we plan to extend our study to the occurrence of predicative adjectives in complement clauses as well as to the occurrences of attributive adjectives. Also, since information structure has not been previously identified as a separate variable impacting the perception of subjective expressions, we will want to study in a controlled way how it interacts with other well known variables such as degree modification. Finally, we want to follow up on the question pursued by (Kabadjov et al., 2011) and investigate whether differences in the salience with which an opinion is expressed influence how helpful these opinions are in opinion summarization.

Acknowledgments

We would like to thank our colleague Berry Claus for discussion and advice on the construction of our stimuli and the statistical analysis.

References

Alan Agresti. 2002. *Categorical data analysis*, volume 359. John Wiley & Sons.

Farah Benamara, Baptiste Chardon, Yannick Mathieu, Vladimir Popescu, and Nicholas Asher. 2012. How do negation and modality impact on opinions? In *Proceedings of the Workshop on Extra-Propositional Aspects of Meaning in Computational Linguistics*, pages 10–18. Association for Computational Linguistics.

Heike Bieler, Stefanie Dipper, and Manfred Stede. 2007. Identifying formal and functional zones in film reviews. *Proceedings of the 8th SIGDIAL*, pages 75–78.

Rebecca F Bruce and Janyce M Wiebe. 1999. Recognizing subjectivity: a case study in manual tagging. *Natural Language Engineering*, 5(2):187–205.

Rune Haubo B Christensen. 2011. A Tutorial on fitting Cumulative Link Mixed Models with clmm2 from the ordinal Package. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.224.8041&rep=rep1&type=pdf.

Mick P Couper, Michael W Traugott, and Mark J Lamias. 2001. Web survey design and administration. *Public opinion quarterly*, 65(2):230–253.

Gerard de Melo and Mohit Bansal. 2013. Good, great, excellent: Global inference of semantic intensities. *Transactions of the ACL.*

Stefania Degaetano-Ortlieb, Elke Teich, and Ekaterina Lapshinova-Koltunski. 2012. Domain-specific variation of sentiment expressions: a methodology of analysis for academic writing. In *Proceedings of the 12th conference on natural language processing (KONVENS)*, pages 291–295.

Stephan Greene and Philip Resnik. 2009. More than words: Syntactic packaging and implicit sentiment. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 503–511. Association for Computational Linguistics.

Vasileios Hatzivassiloglou and Kathleen R McKeown. 1997. Predicting the semantic orientation of adjectives. In *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics*, pages 174–181. Association for Computational Linguistics.

Bas Heerschop, Frank Goossen, Alexander Hogenboom, Flavius Frasincar, Uzay Kaymak, and Franciska de Jong. 2011. Polarity analysis of texts using discourse structure. In *Proceedings of the 20th ACM international conference on Information and knowledge management*, pages 1061–1070. ACM.

Julian Heister, Kay-Michael Würzner, Johannes Bubenzer, Edmund Pohl, Thomas Hanneforth, Alexander Geyken, and Reinhold Kliegl. 2011. dlexDB–eine lexikalische Datenbank für die psychologische und linguistische Forschung. *Psychologische Rundschau*, 62(1):10–20.

Mijail Kabadjov, Alexandra Balahur, and Ester Boldrini. 2011. Sentiment intensity: Is it a good summary indicator? In Zygmunt Vetulani, editor, *Human Language Technology. Challenges for Computer Science and Linguistics*, volume 6562 of *Lecture Notes in Computer Science*, pages 203–212. Springer Berlin Heidelberg.

Knud Lambrecht. 1996. *Information structure and sentence form: Topic, focus, and the mental representations of discourse referents*. Cambridge University Press.

Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *Proceedings of the Recent Advances in Natural Language Processing International Conference*, pages 378–382.

Subhabrata Mukherjee and Pushpak Bhattacharyya. 2012. Sentiment analysis in Twitter with lightweight discourse analysis. In *Proceedings of COLING 2012*, pages 1847–1864, Mumbai, India, December. The COLING 2012 Organizing Committee.

Peter Prettenhofer and Benno Stein. 2010. Cross-language text classification using structural correspondence learning. In *Proceedings of the ACL.*

Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. SentiWS - A Publicly Available German-language Resource for Sentiment Analysis. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, May. European Language Resources Association (ELRA).

Josef Ruppenhofer, Michael Wiegand, and Jasper Brandes. 2014. Comparing methods for deriving intensity scores for adjectives. In *Proceedings of the 14th Conference of the European Chapter of the Association for Computational Linguistics, volume 2: Short Papers*, pages 117–122, Gothenburg, Sweden, April. Association for Computational Linguistics.

Vera Sheinman and Takenobu Tokunaga. 2009. Adjscales: Visualizing differences between adjectives for language learners. *IEICE TRANSACTIONS on Information and Systems*, 92(8):1542–1550.

Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. 2011. Lexicon-based methods for sentiment analysis. *Computational linguistics*, 37(2):267–307.

Ulli Waltinger. 2010. GermanPolarityClues: A Lexical Resource for German Sentiment Analysis. In Nicoletta Calzolari, Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, Mike Rosner, and Daniel Tapias, editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Fei Wang, Yunfang Wu, and Likun Qiu. 2012. Exploiting discourse relations for sentiment analysis. In *Proceedings of COLING 2012: Posters*, pages 1311–1320, Mumbai, India, December. The COLING 2012 Organizing Committee.

Janyce Wiebe and Ellen Riloff. 2005. Creating subjective and objective sentence classifiers from unannotated texts. In *Proceedings of the 6th International Conference on Computational Linguistics and Intelligent Text Processing*, CICLing'05, pages 486–497, Berlin, Heidelberg. Springer-Verlag.

Janyce Wiebe. 2000. Learning subjective adjectives from corpora. In *AAAI/IAAI*, pages 735–740.

Michael Wiegand, Alexandra Balahur, Benjamin Roth, Dietrich Klakow, and Andrés Montoyo. 2010. A survey on the role of negation in sentiment analysis. In *Proceedings of the workshop on negation and speculation in natural language processing*, pages 60–68. Association for Computational Linguistics.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proceedings of the conference on human language technology and empirical methods in natural language processing*, pages 347–354. Association for Computational Linguistics.

Verb Polarity Frames: a New Resource and its Application in Target-specific Polarity Classification

Manfred Klenner
Computational Linguistics
University of Zurich
Switzerland
klenner@cl.uzh.ch

Michael Amsler
Computational Linguistics
University of Zurich
Switzerland
mamsler@cl.uzh.ch

Nora Hollenstein
Computational Linguistics
University of Zurich
Switzerland
hollenstein@cl.uzh.ch

Abstract

We discuss target-specific polarity classification for German news texts. Novel, verb-specific features are used in a Simple Logistic Regression model. The polar perspective a verb casts on its grammatical roles is exploited. Also, an additional, largely neglected polarity class is examined: controversial texts. We found that the straightforward definition of 'controversial' is problematic. More or less balanced polarities in a text are a poor indicator of controversy. Instead, non-polar wording helps more than polarity aggregation. However, our novel features proved useful for the remaining polarity classes.

1 Introduction

We focus on fine-grained sentiment analysis in a document-level, target-specific polarity classification task. By fine-grained we refer to a sentiment analysis that captures sentiment composition at the phrase or even clause level based on reliable lexical resources, e.g., polarity lexicons. The task includes the recognition of targets and whether a (nearby) polar expression relates to it and how. Existing approaches have focused on different aspects of this task: the identification of targets and their components (Popescu and Etzioni, 2005), the induction of contextual polarity (Wilson et al., 2005), subjectivity word sense disambiguation (Akkaya et al., 2009), sentence-level composition (Moilanen and Pulman, 2007), and the specification of fine-grained lexical resources that help to better distinguish between factual and subjective language or even relate the polarity of expressions to emotion categories (Neviarouskaya et al., 2009). While recent research relying on a recursive neural tensor network (Socher et al., 2013) has shown that a high scoring sentiment analysis system that even copes with some effects and scopes of negation and with compositionality can also be trained with machine learning techniques, such an approach relies heavily on the annotated resources available, a sentiment treebank in this case.

Moving from English to other languages (German, in our case) confronts one with the lack of comparable resources, be it fine-grained polarity lexicons or – more seriously – the lack of gold standard data for training and evaluation of machine learning approaches. In order to change this situation, we have started to create a fine-grained polarity lexicon and a verb resource similar but not identical to (Neviarouskaya et al., 2009). We have also implemented a fast system carrying out sentiment composition, but one problem remained: how to evaluate in the absence of a (phrase- and sentence-level) gold standard[1]. Fortunately, we have access to a large text corpus (80,000 texts) where newspaper texts and dedicated actors in them are classified as positive,

[1] The MLSA corpus (Clematide et al., 2012) could have been a starting point, but is small and only captures NP-level composition.

negative, neutral or controversial[2]. This way, an extrinsic, i.e., application-oriented evaluation was possible. The goal was to reproduce the human-annotated target-specific classifications on the basis of our newly created resources. Could such a system be used to filter the huge amount of daily upcoming texts in order to, e.g., more directly access interesting (positive, negative, neutral or controversial) texts on a given target? The disadvantage of this resource is that it requires a demanding classification task, namely target classification including a class "controversial". There are only few approaches trying to cope with that problem (e.g. (Tsytsarau et al., 2010)). However, to withdraw these texts from our corpus was no option, since it would have made the intended application impossible. Unfortunately, no interannotator agreement (IAA) was measured for the text corpus. Thus, we conducted a small study (200 texts) in order to find out how well human annotators could reproduce the demanding "controversial" (expected) gold standard classifications. IAA turned out to be surprisingly low: if we take human performance as an upper bound, our system must beat 33% precision) – a poor value (overall accuracy was 66%).

In the present study, we combine text classification and features derived and aggregated from sentiment composition in an extrinsic evaluation in order to evaluate the impact of our newly created resources. No special attention was paid to "controversial target recognition". We not only believe that this task needs special treatment (as we argue in section 2), but also that no conclusions can be drawn given a gold standard class that even humans cannot reliably reproduce. (In addition to this, it should be mentioned that target specific sentiment analysis is considered to be more difficult in news texts than in other text genres (Balahur et al., 2010)).

The rest of this paper is structured as follows. After the related work section, we briefly discuss the origin and intended usage of our text corpus, introduce our resources and describe our approach to sentiment composition. In the experimental sections, we describe and measure the impact of our various features, given different partitions (and, thus, class distributions) of our text corpus. Finally, we draw some conclusions.

2 Related Work on Controversial Texts

There are only a few approaches dealing with the classification of controversial targets. In (Choi et al., 2010) and (Tsytsarau et al., 2010), the hypothesis is that a topic is controversial if the difference between negative and positive phrase level polarity is within a heuristically determined range. This is in line with the annotation guidelines of our gold standard corpus, where target evaluations are considered controversial if positive and negative aspects are balanced and no polarity clearly prevails. We have included features capturing positive-negative ratios of various types of polar expressions (lexicon-based, composition-based etc.) in our experiments - without success.

(Choi et al., 2010) try to detect (new) controversial topics (and subtopics) from text collections, while we focus on intra-text detection of controversial discussions.

Dori-Hacohen and Allan (2013) try to find out if a web page discusses a (known) controversial topic. A web page is controversial if it is similar to a controversial Wikipedia article (on that topic).

3 Text Corpus

Our text corpus, used as a gold standard, was created by the *fög* institute (Research Institute for the Public Sphere and Society)[3] carrying out quantitative-qualitative media content and media reputation analysis. This institute analyses the media reputation of the key sectors of financial and real economies. Media reputation is defined by ((Deephouse, 2000), p. 1097) as "the overall evaluation of the firm presented in the media resulting from the stream of media stories about the firm". The content analysis examines how frequently and strongly (centrality) the media report on specific companies and how they were evaluated (polarity). The recorded encodings (positive, neutral, negative and controversial) allow the institute to build a Media Reputation Index.

[2] No fine-grained annotations are available, e.g. no phrase- or sentence-level polarities.

[3] *http://www.foeg.uzh.ch/*

4 Fine-grained Polarity Lexicon

We aim at a compositional treatment of phrase- and sentence-level polarity. In order to assure high quality, we rely on a manually crafted polarity lexicon specifying the polarities of words (not word senses). Recently, fine-grained distinctions have been proposed that distinguish between various forms of positive and negative polarities, e.g. (Neviarouskaya et al., 2009). For instance, the appraisal theory (Martin and White, 2005) suggests to distinguish between appreciation ("sick friend"), judgement ("deceitful friend") and emotion ("angry friend") . Especially if polarity composition comes into play, it might be crucial to keep these different kinds of polarity separate. We want to properly distinguish cases like "admire a sick friend" (no polarity expectation conflict) from "admire a deceitful friend" - where a polarity conflict occurs (in general, "admire" expects a positive direct object, however a factually negative NP with a non-active connotation does not seem to violate this condition).

We have adopted the categories of the appraisal theory. Our German polarity lexicon comprises about 7,000 single-word entries (nouns, adjectives, adverbs), manually annotated for positive and negative prior polarity where each class further specifies whether a word is factually, morally or emotionally polar. We also coded whether the word involves an active part of the related actor (where applicable) and whether it is weakly or strongly polar. Our ultimate goal is to combine this resource with our verb resource (described in section 5.2) in order to predict the polarity of the arguments of a verb or even to be able to deal with conflicts arising from violated polarity expectations of the verb. In the present study, we use the fine-grained polar values from the lexicon as features (e.g. we count how many words with a prior polarity from the factual axis appear together with the target). But we also enumerate the number of positive and negative arguments stemming from verb expectations and effects (see next section).

Also part of our lexicon are shifters (inverting the polarity, e.g., "a good idea" (positive) vs. "no good idea" (negative)), intensifiers and diminishers.

5 Sentiment Composition

5.1 Phrasal Level

According to the principle of compositionality and along the line of other scholars (e.g. (Moilanen and Pulman, 2007)), after mapping polarity from the lexicon to the words of the text, in the next step we calculate the polarity of nominal and prepositional phrases, i.e., based on the lexical marking and taking into account syntactic (dependency) structure, we conduct a composition of polarity for the phrases.

In general, the polarities are propagated bottom-up to their respective heads of the NPs/PPs in composition with the other subordinates. To conduct this composition we convert the output of a dependency parser (Sennrich et al., 2009) into a constraint grammar format and use the `vislcg3`-tools (VISL-group, 2014) which allows us to write the compositional rules in a concise manner.

5.2 Verb Polarity Frames: Effects and Expectations

In order to merge the polar information of the NPs/PPs on the sentence level one must include their combination via their governor which is normally the verb. Neviarouskaya et al. (2009) propose a system in which special rules for verb classes relying on their semantics are applied to attitude analysis on the phrase/clause-level. Reschke and Anand (2011) show that it is possible to set the evaluativity functors for verb classes to derive the contextual evaluativity, given the polarity of the arguments. Other scholars carrying out sentiment analysis on texts that bear multiple opinions toward the same target also argue that a more complex lexicon model is needed and especially a set of rules for verbs that define how the arguments of the subcategorization frame are affected - in this special case concerning the attitudes between them (Maks and Vossen, 2012).

Next to the evidence from the mentioned literature and the respective promising results, there is also a strong clue coming from error analysis concerning sentiment calculation in which verbs are treated in the same manner as the composition for polar adjectives and nouns described above. This shows up especially if one aims at a target specific (sentence-level) sentiment analysis: in a given sentence "*State attorney X accuses Bank Y of investor fraud.*" one can easily infer that *accuse* is a verb carrying a negative polarity. But in this example the direct object *Bank Y* is accused and should therefore receive a negative "effect" while the *State attorney X* – as the subject of the verb – is not negatively affected (it is his duty to investigate and prosecute financial fraud). Second, the PP *of investor fraud* is a modification of the accusation (giving a reason) and there is intuitively a tendency to expect a negative polarity of this PP - otherwise the accusation would be unjust (In the example given, the negative expectation matches with the composed polarity stemming from the lexically negative "fraud"). So it is clear that the grammatical function must be first determined in order to accurately calculate the effects and expectations that are connected to the lexical-semantic meaning of the verb.

Furthermore, the meaning of the verb (and therefore the polarity) can change according to the context (cf. "report a profit" (positive) vs. "report a loss" (negative) vs. "report an expected outcome"(neutral)). This leads to a conditional identification of the resulting verb polarity (or verbal phrase respectively) in such a manner that the polarity calculated for the head of the object triggers the polarity of the verb. In German, for instance, there are verbs that not only change their polarity in respect to syntactic frames (e.g. in reflexive form) but also in respect to the polarity of the connected arguments, too (see Tab. 1). Of course, any further modifiers or complements of the verb must also be taken into account.

German	English	Polarity
für die Kinder sorgen	to take care of the kids	positive
für Probleme[neg.] sorgen	to cause problems	negative
für Frieden[pos.] sorgen	to bring peace	positive
sich sorgen	to worry	negative

Table 1: Several examples for the use of the German verb "sorgen".

We therefore encode the impact of the verbs on polarity concerning three dimensions: effects, expectations and verb polarity. While effects should be understood as the outcome instantiated through the verb, expectations can be understood as anticipated polarities induced by the verb. The verb polarity as such is the evaluation of the whole verbal phrase. To sum up: in addition to verb polarity, we introduce effects and expectations to verb frames which are determined through the syntactic pattern found (including negation), the lexical meaning concerning polarity itself and/or the conditional polarity respective to the bottom-up calculated prevalent polarities. This results at the moment in over 120 classes of verb polarity frames with regard to combinations of syntactic patterns, given polarities in grammatical functions, resulting effects and expectations, and verb polarity.

As an example we take the verb class *fclass_subj_neg_obja_eff_verb_neg* which refers to the syntactic pattern (subject and direct object) and at the same time indicates which effects and/or expectations are triggered (here negative effect for the direct object). If the lemma of the verb is found and the syntactic pattern is matched in the linguistic analysis, then we apply the rule and assign the impacts to the related instances. However, the boundary of syntax is sometimes crossed in the sense that we also include lexical information if needed. For instance, if we specify the lemma of the concerning preposition in the PP as in *fclass_neg_subj_eff_reflobja_prepobj[um]_verb_neg* (in this case "um" (for); note the encoded reflexive direct object), we leave the pure syntax level.

As mentioned above, one of the goals is the combination of the resources (polarity lexicon and verb annotation). This combination provides us with new target specific sentiment calculations which were not possible in a compositional sentiment analysis purely relying on lexical resources and cannot be reliably inferred via a fuzzy criterion like nearness to other polar words. The effects and expectations of an instantiated syntactic verb pattern in combination with bottom-up propagated and composed polarity can therefore be used to approach the goal of sentence-level sentiment analysis based on a deep linguistic analysis. Furthermore our system offers a possibility to detect violations of expected polarities (“admire a deceitful friend”), i.e., if the bottom-up composed polarity and the effects or expectations coming from the verb frame have an opposite polarity (see (Klenner et al., 2014b) and (Hollenstein et al., 2014)).

As a side-effect of this combination of resources our system can be used in future on the one hand to improve the polarity lexicon through automatic detection of good candidates for the lexicon in the case of reoccuring words on polar expectation for grammatical functions (e.g. “threaten so. with X”; X has a negative polarity expectation, see (Klenner et al., 2014a) for a similar approach). On the other hand, new syntactic patterns in combination with specific verbs can also be detected for annotation in the case of reoccurring bottom-up composed polarity. This procedure as a whole can then be applied especially for gathering domain specific resources.

6 Pipeline Architecture

The documents of our text corpus are parsed, transformed to VISL format and then composition takes place. Targets are identified at that stage as well, and if they are assigned as an argument (e.g. subject) to a modelled verb frame, expectations or effects are asserted. A feature selector then operates on the VISL output, extracting and accumulating polar information (see the next section). Clearly, polar features seem to be better suited to predict the positive, negative or controversial polarity of a target than its neutral polarity. Since text classification has proved successful in document-level polarity classification (Pang et al., 2002), we defined a pipeline where the class probabilities of a text classifier form additional input features to a second classifier. Our hypothesis was that both approaches, text classification and classification on the basis of polar feature vector turn out to be complementary.

More technically, in the first step, a text classifier is trained and applied to our text corpus using 5-fold cross validation. The results of the (test) folds are merged and the class probabilities are extracted and kept as features for the next step - the target polarity classification based on feature vectors comprising prior polarities, phrase level polarities produced by sentiment composition etc. (see next section).

We have experimented with various machine learning algorithms and frameworks, including SVM, Naïve Bayes, Logistic Regression, k-nearest Neighbor. We compared the results of the Stanford classifier[4] to those of Mallet, Megam and Rainbow. We found that Rainbow (McCallum, 1996) produced the best results for our text classification needs. On the other hand, Simple Logistic Regression as provided by Weka (Hall et al., 2009) performed best given our combined feature set. We experimented with feature selection, but none of the feature lists produced were able to outperform the class-specific feature selection automatically carried out by Simple Logistic Regression (cf. (Sumner et al., 2005)).

7 Feature Extraction

We have developed a feature extraction pipeline that extracts information about various polarity levels in words, phrases and sentences of the newspaper articles in our data set. Our feature selection chooses five sets of features which are then combined with the probabilities of the Rainbow text classification system to train a Simple Logistic classifier. With this method we allow features based on ordinal text classification as well as features based on our sentiment analysis resource.

In order to use our sentiment composition approach for machine learning we extract five different sets of features, resulting in a total of 150 features.

In short, our features are constructed as follows (referred to in Table 3):

1. *Text classification probabilities (Rainbow) (8 features)*: We take the output probabilities of Rainbow for each text as features for training the Simple Logistic classifier.

2. *Lexicon-based features (26 features)*: On the one hand, these comprise simple frequency counts of positive and negative words in the documents, taking into account the fine-grained information provided in our polarity lexicon. This means that we extracted

[4] *http://nlp.stanford.edu/software/classifier.shtml*

additional special features which are only concerned with the factual, moral or emotional values of the polar words in the training documents (as described in section 4), e.g. the sum of morally negative adjectives and nouns. On the other hand, we also include features capturing positive-negative ratios mapped to various dimensions. Moreover, we represent structural information by extracting features oriented at the title and the lead of the newspaper articles.

3. *Composition-based features (15 features)*: This feature set describes the information found in nominal and prepositional phrases mapped to the functional heads. Once more, it is possible to distinguish between features which represent frequency counts and features which represent polarity ratios.

4. *Verb-specific features (20 features)*: The goal of the verb-specific features is to extract the information modelled by our verb resource. For instance, we sum all occurrences of subjects and direct objects that receive a positive/negative "effect" from a verb. These features include the "effects" and "expectations" of a given verb as well as the polarity of the verb itself. Furthermore, we model the ratio between polar verbs and the amount of tokes in a text as well as the ratio between positive and negative verbs. These ratios can also be found in the lexicon-based and composition-based feature sets.

5. *Target-specific features (81 features)*: This last feature set is the largest one as it contains all of the information presented in the previous feature sets (2.)-(4.) in connection with phrases or sentences that include a target mention, e.g. the frequency of sentences in which a polar verb that has a direct relation to the target, or the frequency of a target appearing in a polar nominal or prepositional phrase. We also included different positive-negative ratios such as the ratio between targets which appear inside a positive phrase and targets which appear inside a negative phrase. Finally, we combined all the target-related features into two features which represent the complete amount of positive/negative information in the target sentences of one document.

We trained a Simple Logistic classifier on the described set of 150 features. Remarkably, fewer features reduced performance, although Simple Logistic always selected a proper subset of the features.

The impact of the five feature sets and the improvements achieved in comparison to the baseline system will be discussed in the next section.

8 Experiments

In our experiments, we seek to clarify three questions. What is the effect of polar features on classification accuracy? Does this effect depend on the text domain (e.g. finance versus insurance) and can we build high-precision classifiers by filtering text classification results accordingly?[5]

Articles	neut	neg	pos	contr	Entropy
5,000	0.18	0.36	0.19	0.28	0.584
10,000	0.35	0.28	0.14	0.22	0.580

Table 2: Class distribution.

8.1 Experiment I

In order to find out how strong the contribution of our new polarity resources and the features derived from it are, we draw a 5,000 document subset from the text corpus that maximizes target-verb-linkages. If a target is assigned as an argument to one of our verbs (e.g. is the subject or object of the verb), it inherits often a polarity (an effect or an expectation). Thus, the more such dependency links are found in a document, the stronger the impact should be. In other words, is it reasonable to extend our verb resource? Does it help to improve accuracy? Or is the performance independent of the applicability (the fitness) of our resource? We compared the results for the 5,000 set to a second subset comprising 10,000 documents, randomly drawn, but adhering to the distribution of the whole population (see Tab. 2).

[5] Our results in section 8.1 as well as the domain-specific results in section 8.2 based on accuracy all proved significant under the McNemar's paired test.

Description	5,000 articles					10,000 articles				
Feature sets	**Acc**	**neut**	**neg**	**pos**	**contr**	**Acc**	**neut**	**neg**	**pos**	**contr**
Baseline	50.02	34.2	62.8	49.6	**39.4**	52.32	62.2	56.7	37.1	**33.4**
Rainbow	49.86	34	63.1	49.7	37.7	52.58	62.2	57.4	37.2	32.1
Lexicon-based	50.98	33.6	64.5	53.4	36.7	52.66	62.7	57.3	37.7	30.6
Composition-based	50.78	33.1	64.6	53.0	36.5	52.75	62.9	57.3	37.7	30.6
Verb-specific	51.30	32.4	65.4	53.9	36.9	52.89	62.9	57.6	37.2	31.2
Target-specifc	**51.78**	**35.3**	**65.6**	**55.0**	36.7	**53.32**	**63.2**	**58.2**	**38.8**	31.8

Table 3: Results for dataset with 5,000 and 10,000 articles showing overall accuracy and f-measures for each class.

Description		Accuracy			Class Distribution			
Domain	**Articles**	**SA150**	**TC**	**TC+SA150**	**pos**	**neg**	**neut**	**contr**
Retail trade	1515	41.45	42.13	44.82	0.27	0.22	0.29	0.23
Pharma	3845	41.45	48.64	49.67	0.24	0.28	0.24	0.24
Transport	3155	44.98	48.54	50.11	0.13	0.33	0.27	0.27
Media	1310	46.64	47.25	50.53	0.11	0.35	0.27	0.27
Telecom	1438	48.09	51.02	50.54	0.19	0.24	0.38	0.19
Industry	1476	45.94	52.31	54.13	0.33	0.20	0.27	0.21
Insurance	4983	47.56	54.56	56.74	0.19	0.25	0.37	0.19
Banks	31373	51.94	61.43	63.34	0.12	0.43	0.36	0.18
Political inst.	3110	60.03	65.03	65.03	0.07	0.16	0.59	0.17
Unions	3685	71.46	72.20	73.33	0.05	0.11	0.67	0.17

Table 4: Domain-specific sentiment analysis (TC = Text Classification, SA150 = 150 sentiment analysis features).

In Tab. 3, the baseline (label Baseline) is taken from the output of rainbow (its class decision). We took also the class probabilities of rainbow as features (label Rainbow), followed by our polar features as described in the previous section. The improvement in accuracy is moderate (from 50.02% to 51.78%). However, those classes that should profit most from our features, namely negative and positive, actually do show a clear improvement: from 62.8% to 65.6% (negative) and from 49.6% to 55% (positive).

The baseline in accuracy on the right-hand side of Tab. 3 (10,000 texts) is higher (52.32% compared to 50.02%). However, the impact of our features is lower (1% giving 53.32%). Especially the impact on positive and negative classes is lower compared to the 5,000 subset which maximizes fitness of (our) resources.

Note that in both scenarios the (text classification) baseline accuracy of "controversial" decreases as our features are added. As mentioned in the introduction, we cannot deal with this kind of target evaluations, currently.

8.2 Experiment II

We wanted to know whether the classifier performance is stable in different domains, i.e. whether our resources and system components establish a (more or less) domain-independent machinery. We grouped the texts into their domains (e.g. finance, insurance etc.) and run the classifier. Tab. 4 shows that while the text classifier (TC) sets a different baseline depending on the domain (e.g. 42.13% Retail Trade; 72.20% Unions), the contribution of the polar features (TC+SA150) remains, compared to baseline variance, constant: the mean improvement is 1.4% (incl. one accuracy drop and one constant value). Note that in this experiment the full dataset is used (80,000). This explains performance drop compared to the (deliberately chosen) well fitting 5,000 subset.

There is one domain where performance stays constant (Political institutions) and one where it drops (Telecom). In both cases the majority class is neutral, indicating, again, that our polar features better capture positive and negative than neutral and controversial cases.

Tab. 4 also shows the performance of the two classifiers independently from each other (SA150 compared to TC). We can see that text classification always produces higher values. For instance, for Retail trade: 41.45% compared to 42.13%. Since the sum of neutral and controversial (except for one case) together forms the majority of documents (see Tab. 4: Class distribution), this might just be a reflection of the slightly biased data (SA150 is good with positive and negative classes).

Since we have included a text classifier in our pipeline whose accuracy correlates with the probability of the decision (i.e. the confidence value), we wanted to know if we could create a scenario where we only give a classification for cases, where a certain probability is reached – implicating that accuracy would then also increase or at least not decrease. This scenario faces the challenge of the *fög* to cope with large amounts of newspaper articles every day. It is not only expensive to have human annotators classify the data, it might also be ineffective, since choosing a random sample of texts is always in danger of flaws concerning the representativeness of the sample. A high precision system would allow the *fög* to search for interesting texts, either from one of the classes, or even w.r.t. the polar load of texts.

As a further precondition, we set the minimum of the percentage of the documents that have to be classified (this number naturally decreases if one uses the probability of the classifier as a threshold) to 80%. Then we determine the concerning confidence value threshold and tried the classifier without and with our sentiment features only for those documents. It has to be noticed, that the high percentage of processed articles could only be reached with a Naïve Bayes (NB) classifier since the Maximum Entropy classifier (rainbow) had only high probabilities (relative to all probabilities in connection with good accuracy) for very small percentages.

Tab. 5 shows that this time the boost in accuracy when adding the sentiment features for the classification task is relatively stable over several domains. We can see that there is a gain in using the sentiment features along with the text classifier for the task even if "most difficult" cases for the text classifier are filtered out. This means that the improvement through the sentiment features does not only occur in the cases where the text classifier itself has decided badly.

Domain	**NB**	**NB+SA150**	**% articles**
Banks	60.97%	62.03%	90.4%
Pol. inst.	63.49%	64.23%	96.9%
Unions	72.8%	73.1%	96.5%
Insurance	55.7%	57.4%	90.6%
Transport	48.82%	50.96%	84.7%
Pharma	49.18%	49.96%	88.2%

Table 5: Results for different domains, filtered by probability of the text classifier (NB = Naive Bayes text classification, SA150 = 150 sentiment analysis features, % articles = percentage of articles processed under the corresponding accuracy).

9 Conclusion

We have introduced an approach for target-specific sentiment analysis that combines the output of a text classifier with features derived from fine-grained, compositional sentiment analysis. These two components are (at least in part) complementary: text classification better deals with class-specific wording (e.g. words indicating contrastive language), while polarity-based features better capture (and aggregate) the polar load of target-specific descriptions.

Our experiments have shown that operationalization of a class like "controversial" is difficult since there is no clear borderline to news texts which are slightly polar (positive, negative) or neutral. This is reflected in the fact that even human annotators reach only a poor interannotator agreement. Maybe a level of polarity (positive, negative) in combination with a single measurement for controversy could provide more reliable results since the (somehow subjective) decision could then be left to human judgement or to ex-post definitions.

The experiment with articles concerning different domains have shown some remarkable differences in the results. The baseline set by the text classifier varies considerably, whereas the contribution of our polar features is more or less stable.

This seems to indicate that the performance of text classification is much more domain-specific than features based on sentiment composition (and a general polarity lexicon).

Our experiments with a data subset of 5,000 texts that maximizes fitness of our resources have shown that the contribution of our features actually improve results on the proper polar classes, namely positive and negative. This is good news, since performance gain can now be coupled to the further development of our resources, especially the verb resource. However, especially with respect to the controversial dimension an in-depth error and data analysis is needed. We also hope to improve our evaluation process by creating more fine-grained annotated text, i.e., with annotation of certain text areas which lead the human annotator to his judgement relating to a specific target.

Acknowledgments

We would like to thank the *fög* institute, especially Mario Schranz and Urs Christen, for allowing us to use their huge target-level annotated text corpus. Also, we would like to thank the reviewers for their helpful comments and suggestions.

References

Cem Akkaya, Janyce Wiebe, and Rada Mihalcea. 2009. Subjectivity word sense disambiguation. In *EMNLP*, pages 190–199.

Alexandra Balahur, Ralf Steinberger, Mijail Kabadjov, Vanni Zavarella, Erik van der Goot, Matina Halkia, Bruno Pouliquen, and Jenya Belyaeva. 2010. Sentiment analysis in the news. In *Proceedings of the Seventh conference on International Language Resources and Evaluation (LREC'10)*, pages 2216–2220, Valletta, Malta, May.

Yoonjung Choi, Yuchul Jung, and Sung-Hyon Myaeng. 2010. Identifying controversial issues and their sub-topics in news articles. In Hsinchun Chen, Michael Chau, Shu-hsing Li, Shalini Urs, Srinath Srinivasa, and G.Alan Wang, editors, *Intelligence and Security Informatics*, volume 6122 of *Lecture Notes in Computer Science*, pages 140–153. Springer Berlin Heidelberg.

Simon Clematide, Stefan Gindl, Manfred Klenner, Stefanos Petrakis, Robert Remus, Josef Ruppenhofer, Ulli Waltinger, and Michael Wiegand. 2012. MLSA - a multi-layered reference corpus for German sentiment analysis. In *Proceedings of the 8th International Conference on Language Resources and Evaluation (LREC-2012)*, pages 3551–3556.

David L. Deephouse. 2000. Media reputation as a strategic resource: An integration of mass communication and resource-based theories. *Journal of Management*, 26(6):1091–1112.

Shiri Dori-Hacohen and James Allan. 2013. Detecting controversy on the Web. In *CIKM '13*.

Mark Eisenegger and Kurt Imhof. 2008. The true, the good and the beautiful: Reputation management in the media society. In Betteke van Ruler Ansagr Zerfass and Sriramesh Krishnamurthy, editors, *Public Relations Research: European and International Perspectives and Innovation*. VS Verlag für Sozialwissenschaften.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The WEKA data mining software: an update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

Hollenstein, Nora and Amsler, Michael and Bachmann, Martina and Klenner, Manfred. 2014. SA-UZH: Verb-based Sentiment Analysis. Proceedings of the 8th International Workshop on Semantic Evaluation (SemEval 2014), Dublin, Ireland.

Manfred Klenner, Michael Amsler, and Nora Hollenstein. 2014a. Inducing Domain-specific Noun Polarity Guided by Domain-independent Polarity Preferences of Adjectives. In Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment & Social Media Analysis (WASSA 2014), Baltimore, USA.

Manfred Klenner, Susanna Tron, Michael Amsler, and Nora Hollenstein. 2014b. The Detection and Analysis of Bi-polar Phrases and Polarity Conflicts. In Proceedings of the 11th International Workshop on Natural Language Processing and Cognitive Science (NLPCS 14), Venice, Italy.

Isa Maks and Piek Vossen. 2012. A lexicon model for deep sentiment analysis and opinion mining applications. *Decision Support Systems*, 53(4):680–688.

J. R. Martin and P. R. R. White. 2005. *Appraisal in English*. Palgrave, London.

Andrew Kachites McCallum. 1996. Bow: A toolkit for statistical language modeling, text retrieval, classification and clustering. http://www.cs.cmu.edu/ mccallum/bow.

Karo Moilanen and Stephen Pulman. 2007. Sentiment composition. In *Proc. of RANLP-2007*, pages 378–382, Borovets, Bulgaria, September 27-29.

Alena Neviarouskaya, Helmut Prendinger, and Mitsuru Ishizuka. 2009. Semantically distinct verb classes involved in sentiment analysis. In Hans Weghorn and Pedro T. Isaías, editors, *IADIS AC (1)*, pages 27–35. IADIS Press.

Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. 2002. Thumbs up? Sentiment classification using machine learning techniques. *CoRR*, cs.CL/0205070.

Ana-Maria Popescu and Oren Etzioni. 2005. Extracting product features and opinions from reviews. In *Proc. of HLT-EMNLP-05*, pages 339–346, Vancouver, CA.

Kevin Reschke and Pranav Anand. 2011. Extracting contextual evaluativity. In *Proceedings of the Ninth International Conference on Computational Semantics*, pages 370–374.

Rico Sennrich, Gerold Schneider, Martin Volk, and Martin Warin. 2009. A new hybrid dependency parser for German. In *Proc. of the German Society for Computational Linguistics and Language Technology*, pages 115–124.

Richard Socher, Alex Perelygin, Jean Y. Wu, Jason Chuang, Christopher D. Manning, Andrew Y. Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proc. of the Conference on Empirical Methods in Natural Language Processing (EMNLP'13)*, pages 1631–1642, Seattle, USA.

Marc Sumner, Eibe Frank, and Mark A. Hall. 2005. Speeding up logistic model tree induction. In Alípio Jorge, Luís Torgo, Pavel Brazdil, Rui Camacho, and João Gama, editors, *PKDD*, volume 3721 of *Lecture Notes in Computer Science*, pages 675–683. Springer.

Mikalai Tsytsarau, Themis Palpanas, and Kerstin Denecke. 2010. Scalable discovery of contradictions on the Web. In Michael Rappa, Paul Jones, Juliana Freire, and Soumen Chakrabarti, editors, *WWW*, pages 1195–1196. ACM.

VISL-group. 2013. *VISL CG-3. http://beta.visl.sdu.dk/cg3.html*. Institute of Language and Communication (ISK), University of Southern Denmark.

Theresa Wilson, Janyce Wiebe, and Paul Hoffmann. 2005. Recognizing contextual polarity in phrase-level sentiment analysis. In *Proc. of HLT/EMNLP 2005*, Vancouver, CA.

Resource interoperability revisited

Kerstin Eckart
Universität Stuttgart, IMS
Paffenwaldring 5b
70569 Stuttgart
eckartkn@ims.uni-stuttgart.de

Ulrich Heid
Universität Hildesheim, IWIST
Marienburger Platz 22
31141 Hildesheim
heid@uni-hildesheim.de

Abstract

We present a set of refined categories of interoperability aspects and argue that the representational aspect of interoperability and its content-related aspects should be treated independently. While the implementation of a generic exchange format provides for representational interoperability, content-related interoperability is much harder to achieve. Applying a task-based approach to content-related interoperability reduces complexity and even allows for the combination of very different resources.[1]

1 Introduction

The interoperability of resources is a property which describes how well two resources can interact or be applied together. Thus interoperability is a relevant factor for many processes in the field of natural language processing. Over the years a valuable set of language resources has been created, and this set is still growing. These resources provide the basis for linguistic studies as well as for the development of applications. The more different the resources, the more different are also the forms of interoperability: hence we will start in Section 1.1 with an overview of slightly different notions of interoperability.

Since the creation of a language resource is costly, it is useful to ensure sustainability of the created resources, such that they can be easily reused and extended. One aspect of sustainability is the possibility of existing resources to be in some way combined or utilized together. This way existing and emerging tasks can benefit from the information available in the respective resources. In addition it is important to make transparent where and to which degree resources are interoperable, to increase their acceptance within the user community and to thereby possibly also extend the field in which these language resources are applied.

When we talk about language resources here, we employ a broad definition of the concept. Not only corpora and lexical knowledge bases are subsumed by this notion; also tools for natural language processing, their statistical language models, rule sets, or grammars, as well as data from studies and experiments are to be understood as language resources. Our considerations are intended to capture language data based on different modalities, although the mentioned examples relate to written texts.

1.1 Notions of interoperability

Interoperability of language resources has been discussed in various approaches which focus on different aspects of interoperability and define the concept of interoperability in slightly different ways. In the remainder of this section, we summarize major theoretical viewpoints on the notion of interoperability; below, in Section 4, we will comment on existing implemented applications where interoperability plays a role and discuss them in terms of the theoretical views we will

[1]

develop in this paper.

Witt et al. (2009) state that the most general notion of interoperability of language resources conveys the idea that these resources are able to interact with each other. Consequently, they classify scenarios of interoperability according to the types of resources to be combined, e.g., applying tools to a corpus vs. combining corpora to create a common subset. Additionally they distinguish between (i) a transfer philosophy of interoperability, where a mapping from the information of one resource to the representation of the other resource is applied, and (ii) an interlingua philosophy of interoperability, where data from both resources are mapped to a new representation that generalizes over both. Ide and Pustejovsky (2010) define interoperability as a measure for the degree to which resources are able to work together and thus aim at an operational definition of interoperability. They describe conditions for interoperability for the following four thematic areas: metadata, data categories, publication of resources and software sharing. Additionally they distinguish between syntactic interoperability and semantic interoperability, adopting these notions from the study of interoperability of software systems and adapting them to the field of computational linguistics. According to them, syntactic interoperability is characterized by properties that ensure that different systems are able to exchange data and to process them either without any conversion or including only a trivial conversion step; while semantic interoperability is the capability to interpret the data in an informed and consistent way. Stede and Huang (2012) focus on linguistic annotation and discuss the role of standard formats for interoperability in an interlingua approach. With respect to the contents of resources, they state that comparability of resources also involves methodology issues, taking the process of creating annotation guidelines into account.

We will adopt the general definition of Witt et al. (2009) that defines interoperability of resources as the ability for these resources to interact, work together or be combined. Our approach also distinguishes between representational and content-related aspects, as Ide and Pustejovsky (2010) do, but we will introduce an additional classification on the content side. Thus our definition of syntactic and semantic interoperability is slightly different from theirs. Like Stede and Huang (2012) we will in particular take the aspect of the combination of linguistic annotations into account.

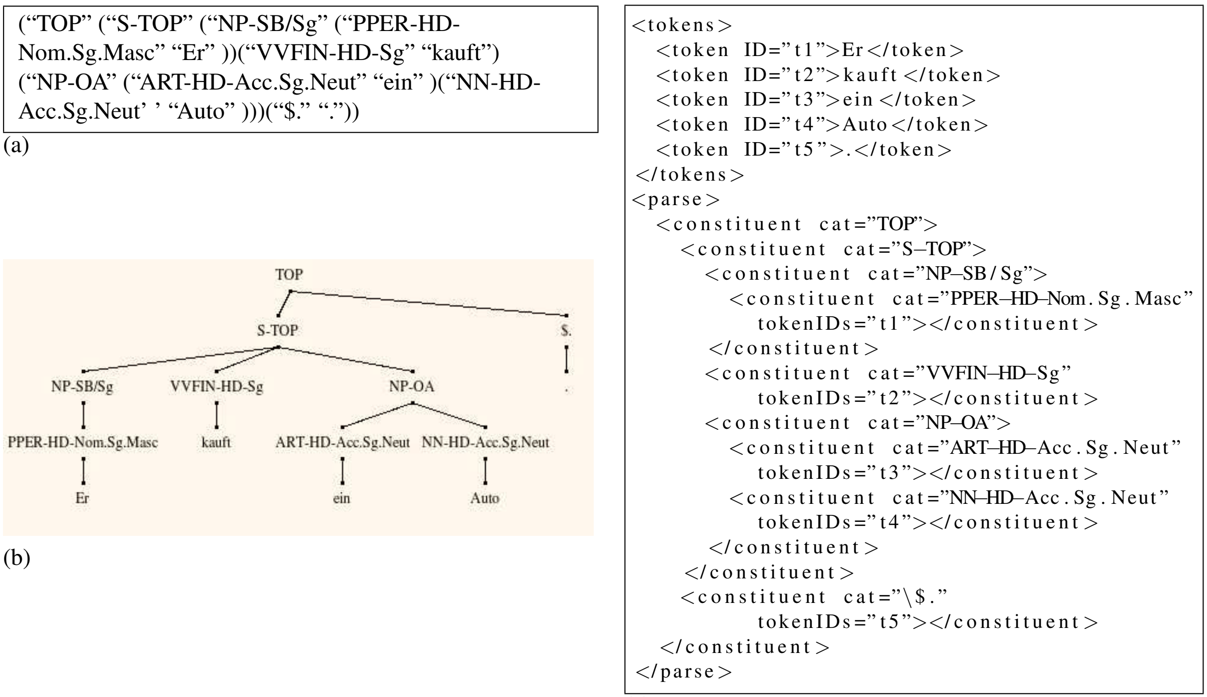
1.2 Outline

Our contribution is twofold: We will (i) propose refined categories of interoperability aspects (cf. Section 2) and a pertaining classification for interoperability approaches to the combination of different resources (cf. Section 3). Since content-related interoperability, especially with respect to the semantics of the content is most difficult to handle, we cannot expect to be able to solve this issue in a general and comprehensive manner. Therefore we will (ii) introduce an application-oriented proposal for the handling of content-related interoperability issues in a task-based setting and illustrate it with a case study from the task-based combination of syntactic annotations (cf. Section 5). Next to the theoretical set-up in Sections 2 and 3 and the exemplification of our application-oriented proposal in Section 5 we discuss further existing applications in Section 4 in which interoperability plays a role either as the main concern of the approach or as an aspect that has to be dealt with in the actual approach.

2 Categories of interoperability

To introduce a detailed classification we present an extended and refined concept of interoperability, especially with respect to annotations. On a high level we distinguish between *representational interoperability* and *content-related interoperability* and we subdivide the latter into *syntactic interoperability* and *semantic interoperability*.

Representational interoperability focuses on the different possibilities of representation, i.e. encodings of information. For example, syntactic information is usually structured as a tree, but this tree can be represented by the introduction of brackets to the original input, or it can be encoded in an XML representation, embedded in a figure or arranged in a tabular format. Figure 1 shows three different representations of exactly the same output content produced by the BitPar



```
(“TOP” (“S-TOP” (“NP-SB/Sg” (“PPER-HD-
Nom.Sg.Masc” “Er” ))(“VVFIN-HD-Sg” “kauft”)
(“NP-OA” (“ART-HD-Acc.Sg.Neut” “ein” )(“NN-HD-
Acc.Sg.Neut’ ’ “Auto” )))(“$.” “.”))
```

(a)

(b)

```
<tokens>
  <token ID=”t1”>Er</token>
  <token ID=”t2”>kauft</token>
  <token ID=”t3”>ein</token>
  <token ID=”t4”>Auto</token>
  <token ID=”t5”>.</token>
</tokens>
<parse>
  <constituent cat=”TOP”>
    <constituent cat=”S–TOP”>
      <constituent cat=”NP–SB/Sg”>
        <constituent cat=”PPER–HD–Nom.Sg.Masc”
            tokenIDs=”t1”></constituent>
      </constituent>
      <constituent cat=”VVFIN–HD–Sg”
            tokenIDs=”t2”></constituent>
      <constituent cat=”NP–OA”>
        <constituent cat=”ART–HD–Acc.Sg.Neut”
            tokenIDs=”t3”></constituent>
        <constituent cat=”NN–HD–Acc.Sg.Neut”
            tokenIDs=”t4”></constituent>
      </constituent>
    </constituent>
    <constituent cat=”\$.”
            tokenIDs=”t5”></constituent>
  </constituent>
</parse>
```

(c)

Figure 1: Three representations of the same linguistic content

parser (Schmid, 2004) for sentence (1).

(1) Er kauft ein Auto.
He buys a car.

With respect to linguistic information, i.e., data categories and their structured combination, these three analyses are identical – each of them encodes the same phrase structure tree based on the same grammar and tagsets[2]. Yet, at first sight, it is hard to even see if they are similar. Figure 1(a) is an inline representation of the annotation, where linguistic information is introduced into the original sentence by means of brackets (structure) and tags (part-of-speech, syntactic and morphological information), similar to a well-known representation format of the Penn Treebank (Marcus et al., 1993). Here (``NP-OA'' denotes the start of the noun phrase *ein Auto* which is the direct object of the sentence. Exactly the same linguistic information is represented differently in Figure 1(b). There, we see a graphical representation of the annotated linguistic structure of the sentence. No brackets are applied, but two edges connect the node labelled NP-OA to its children, the parts of the noun phrase. Figure 1(c) is an XML standoff representation of the annotation as an excerpt of the TCF format (Heid et al., 2010). Here the output of the BitPar Parser is represented in its own layer (<parse/>), i.e. separated from the actual tokens (<tokens/>).

While the examples in Figure 1 show how difficult a manual comparison will be, also an automatic comparison of the output would involve either thorough investigation or complex conversion procedures. Thus we claim that representational interoperability is often the first step towards interoperability of resources and that it should not be confused with the linguistically motivated structural decisions reflected in the content. Especially these content-related structural decisions should not get mingled with representational aspects in the process of comparison or conversion.

Content-related interoperability however comprises all linguistically motivated decisions. Here we introduce the additional distinction between

[2] Part-of-speech: ART – determiner, NN – common noun, PPER – personal pronoun, VVFIN – full verb (finite), $. – punctuation symbol at the end of the sentence; syntactic labels: TOP – root, S – sentence, NP – noun phrase, HD – head, OA – direct object, SB – subject; morphological labels: Sg – singular, Acc – accusative, Nom – nominative, Masc – masculine, Neut – neuter

syntactically and semantically motivated differences.

Syntactic interoperability takes structural decisions into account and thus evaluates the similarity of the underlying models: Is the information based on a tree model, i.e. do we have hierarchical categories and no crossing branches, or will we only be able to capture all intended correlations by a directed acyclic graph? Is a node in the tree allowed to have more or less than two children? Are the correlations labelled? To bring out the difference with representational interoperability, in the latter case, the question of where these labels are attached, i.e. to nodes or to edges, would be a representational question. The question important for syntactic interoperability is if correlations are at all intended to include additional information. On a high level, differences with regard to structural interoperability include for example the differences between phrase structure and dependency trees. Figure 2 shows three syntactic annotations for phrase (2)[3]. Figure 2(a) shows a dependency tree based on the output of a parser of the Mate Tools (Bohnet, 2010) and Figures 2(b) and (c) show two phrase structure trees, based on the output of the parser described in Björkelund et al. (2013) (b) and BitPar (c). While in the dependency tree a token is directly connected to its head, phrase structure trees introduce additional nodes for each phrase[4]. Another aspect of syntactic interoperability can be seen in Figure 2(b) and (c). In Figure 2(b) a flat structure is applied, while in Figure 2(c) *deutsche Elf* is considered a phrase of its own.

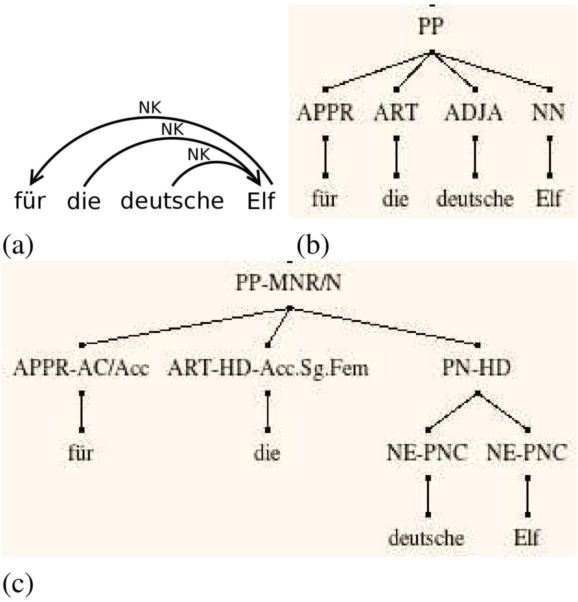


Figure 2: Aspects of syntactic interoperability

(2) für die deutsche Elf
for the German eleven
for the German football team

Semantic interoperability focuses on the concepts that are applied within the resources. These are often subsumed by a tagset, where every tag stands for a concept with which parts of the resource can be labelled. A typical example is part-of-speech tagging, where categories such as `noun`, `verb` or `pronoun` are attached to words or word combinations. Distinctions regarding semantic aspects can be found in the annotation guidelines and in the coverage of the single concepts. In the simplest case two different names are applied for the same concept, e.g. `NN` or `N[comm]` for common nouns. More difficulties arise when the same name is applied for different concepts, e.g. when different approaches to dependency syntax use the term `head`, either to refer to a lexical or to a functional head. A further issue is granularity, i.e. cases where a specific concept is applied in one resource, while the it is split into several concepts in another one. The hardest case is one where two concepts only cover part of each other, and no mapping scheme can be applied.

Thus when aiming at interoperability of resources, we need to assess the above mentioned three categories individually: representational closeness, syntactic closeness and semantic closeness of the respective resources. Even if these aspects are often interrelated, two resources might show discrepancies to a different degree with respect to each of these categories. Taking this separation into account, it is easier to assess what

[3] The full sentence is: Kevin Kuranyi schoß in Prag beide Tore für die deutsche Elf. *Kevin Kuranyi scored in Prague both goals for the German football team.*

[4] Additional tags: part-of-speech: ADJA – attributive adjective, NE – named entity; syntactic labels: NK – noun kernel, PN – proper noun, PP – prepositional phrase, AC – adpositional case marker, MNR – modifier of a noun phrase to the right, PNC – proper noun component; morphological labels: fem – feminine

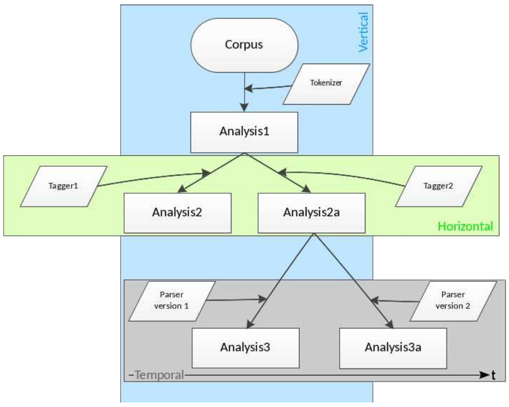


Figure 3: Dimensions of analysis relations as described by Eberle et al. (2012)

is the most beneficial way for a researcher or a project to invest work into achieving resource interoperability.

3 Interoperability: from analysis steps to resources

Since our focus is on interoperability with respect to annotations, we will first discuss which relations can possibly exist between different sets of annotations. Thereafter we introduce classification types for the combination of different resources, which take content-related interoperability into account.

3.1 Interoperable annotations

Linguistic annotations are usually created in an analysis process consisting of several steps. Eberle et al. (2012) describe three dimensions of analysis relations, which are illustrated, for a typical corpus annotation workflow, in Figure 3.

Vertical analysis relations When analysing language data, the analysis steps often reflect a multi-layered structure of language. For textual data the usual (automatic) processing steps include segmentation into sentences and tokens, annotation of part-of-speech tags, generation of syntactic trees representing the structure of each sentence and maybe some further annotation produced e.g. by named entity recognition, coreference resolution, etc. Thus vertical analysis relations exist between 'higher' and 'lower' annotation layers: a 'higher' annotation layer may depend on the information from a 'lower' level. With respect to interoperability, this means that the annotation guidelines and the respective tagset of the 'lower' level need to fit the requirements of the 'higher' level. In an automatic processing chain, these content-related interoperability requirements come with additional ones in terms of representational interoperability: a parser that expects tabular information on segmentation and part-of-speech will not be able to handle plain XML input, even if the tagset is interpretable by the parser.

Horizontal analysis relations For each analysis layer there are several proposals on how to annotate them, starting from different decisions on what a token is, up to the distinction between phrase-structure trees and dependency graphs. Horizontal analysis relations thus exist between alternative analyses from the same linguistic description layer. Since there are many approaches to combine information from different annotations on the same analysis layer (Fiscus, 1997; van Halteren et al., 2001; Björkelund et al., 2013), interoperability is an important factor for the combination of these annotations.

Temporal analysis relations Annotation schemes, annotation tools and knowledge bases may evolve over time. This is catered for by temporal analysis relations. If the same input is annotated by two different instances of the analysis process, the resulting annotation layers might also differ. This relation type is important for the constant development and enhancement of language resources, but is usually rather uncritical with respect to interoperability.

3.2 Interoperable resources

Similar relations exist of course between the resources that are created with vertical, horizontal and temporal analysis steps. An audio corpus, which has been processed by two different systems for the same level of annotation thus yields two resources that are related by a horizontal analysis relation. Documentation on the creation process of a resource therefore often helps to assess if two resources can be applied together. Information about vertical, horizontal and temporal analysis relations can be captured by means of process metadata, stating which input has been processed by which tool(s) in which version.

An important use case for the assessment of interoperability also arises in situations where different resources are to be combined (in a horizontal or a vertical way). In the following we introduce a classification of combination types for such resources, regarding content-related interoperability. Combination type 1 is the case with the highest degree of interoperability, while combination type 3 is a case where neither syntactic nor semantic interoperability are given.

Combination type 1 applies, when two resources are based on the same concepts and the same structural decisions. In this case the resources are fully interoperable with respect to content-related aspects. Examples are different development versions of the same resource or a set of systems taking part in a shared task, where all systems are trained on the same training data.

Combination type 2 applies, when two resources are similar with respect to their structure, but differ with respect to their concepts. Thus semantic interoperability has to be provided while the resources are already syntactically interoperable. Examples are different part-of-speech taggers, that can be applied to the same tokenization; or two lexical resources with a word-based structure but different annotations; or a dependency parser trained on different training sets, that provide for a similar structure with respect to aspects such as projectivity, head-type and coordination.

Combination type 3 applies, when resources differ in structural as well as in semantic criteria. Examples are the combination of prosodic and semantic information based on different segmentations of the primary data; or a query tool for dependency treebanks and a corpus annotated with constituency trees; or a labelled wordnet and a classical lexicon.

4 Existing approaches from new perspectives

In the following some types of realizations from areas like standardization and conversion, shared tasks and evaluation, and processing chains will be classified with respect to our refined concept of interoperability, cf. Section 2, and the classification from Section 3.

Standardization and converter frameworks Stede and Huang (2012) observe that standard formats play an important role in interoperability and tend to be applied as a pivot representation in an interlingua approach, to exchange data between more resource-specific formats without loosing information in the process of mapping. One of these generic exchange formats is GrAF (Ide and Suderman, 2007), the serialization of the Linguistic Annotation Framework LAF (ISO 24612:2012). LAF introduces a layered graph structure, where graphs consist of nodes, edges and annotations. The annotations implement the full power of feature structures and can be applied to nodes and edges alike. All standard annotation layers for linguistic corpora can be mapped onto this model, and since references to the primary data are implemented based on the encoding of their minimal addressable unit, such as characters for a textual representation, or frames for video data, several modalities are covered. LAF/GrAF does thus provide for representational interoperability. Representing each of the three analyses in Figure 1 in GrAF produces an identical result for each of the original representations, and would thus reduce the comparison cost to a minimum. Of course in a typical setting where resources should be combined, the resource annotations are not identical. However mapping them onto a common representation, that is guaranteed to still reflect all resource-specific annotation decisions, helps to bring out the actual content-related differences. To some extent LAF/GrAF can also be used to abstract over features which we relate to syntactic interoperability, such as e.g. condensing annotations from a non-branching path[5] into a combined edge label.

However, by design, LAF itself does not handle semantic interoperability but provides a mechanism for annotation items to link to external concept definitions. Such concept definitions can be set up and referred to in ISOcat[6], a Data Category Registry, based on ISO 12620:2009. There,

[5] Such a non-branching path is e.g. called a unary chain in parsing results.

[6] http://www.isocat.org/

concept definitions are entered in a grass roots approach by the community: if the concept which is needed for a specific resource is not available, it can be entered to the registry. To take care of uncontrolled growth that might result from the grass roots approach, thematic domain groups are supposed to select and recommend specific concepts relevant to thematic domains such as metadata, lexicography, morphosyntax or sign language. Data Category Registries or Concept Registries thus provide most valuable support for semantic interoperability: if two different labels from different resources link to the same concept entry in the registry, they can easily be mapped; if two labels with the same name, but links to different concepts exist in the resources, extra care needs to be taken when the respective resources are to be combined.

In addition, frameworks such as SaltNPepper (Zipser and Romary, 2010) support conversion from one annotation format into another. Salt, the internal meta model of the Pepper converter framework, handles representational differences, and the system also allows to introduce semantic information by external references to ISOcat.

Shared tasks and evaluation projects Shared tasks are usually set up to foster the creation and to enhance and evaluate the quality of language processing systems for a specific task such as machine translation, named entity recognition or dependency parsing. They are however also a platform for the creation of interoperable resources with regard to horizontal relations. In a typical shared task, a certain amount of data is made available that shows the targeted input/output combination. This material can be used to statistically train, or otherwise build a respective system to produce high-value output with respect to the theory or setting the output is based on. At a specific point in time, test data is released, which is processed by the participating systems, and their output is evaluated and ranked by specific metrics. Thus a set of systems emerges, where each system is able to handle the same input data and is aiming to produce the same output information, including the same structure and tagset. These systems are thus possible candidates for an easy combination on the horizontal level.

The project PASSAGE (de la Clergerie et al., 2008), invited parsing systems for French to take part in a collaborative annotation approach of textual data from various sources, including oral transcriptions. The goal was to create a valuable and comprehensive corpus resource for French, by combining the output of different parsing systems in a bootstrapping approach. To be able to combine and merge the annotations, a rather abstract set of categories was defined on which all participating systems could agree. This category set comprised six categories of chunks and fourteen categories of dependencies. On the one hand, this setting brought up an actual use case, where interoperable systems on the same horizontal level were combined to create a new resource. On the other hand, this interoperability was achieved at the cost of abstracting over the content-related differences of the systems, which precisely include the most valuable information in combination approaches.

A similar argumentation applies for the shared tasks regularly conducted in conjunction with the Conference on Natural Language Learning (CoNLL). In 2006 and 2009 the task was on dependency parsing for different languages (Buchholz and Marsi, 2006; Hajič et al., 2009). There the content-related specifications of the system output were not based on the least common denominator like in PASSAGE, but predetermined by the chosen data set for each language. While this allows for more detailed analyses, it still excludes the need for combination of different content-related aspects. However, the CoNLL shared tasks address content-related interoperability in some other respects. Firstly, since the expected output does not only comprise dependency information but also part-of-speech tagging, lemmatization and the identification of morphosyntactic features, the approach thus also fosters interoperability for vertical analysis relations. And secondly, the setup leads to systems that are applicable to many languages. Thereby a language-independent and thus interoperable workflow of training and testing procedures has emerged. Additionally the CoNLL shared tasks gave rise to tabular annotation representations, which have become a de-facto-standard in the field. It thus provides for increasing interoperability on the representational level in horizontal as well as vertical approaches.

An approach to increase content-related, and specifically syntactic interoperability of parser output is embedded in the evaluation methods described by Tsarfaty et al. (2011) and Tsarfaty et al. (2012). In their approach (multi-)function trees are introduced, to which different parse trees can be mapped. In the actual evaluation, tree edit distance is utilized but does not take edits into account which adhere to theory-specific aspects. In multi-function trees, e.g., unary chains over grammatical functions can be condensed into a single edge with a respective label set, thus increasing the syntactic interoperability of the analyses.

Processing chains Processing chains usually implement one path of vertical analysis relations, e.g., starting from the tokenization of primary data and leading up to syntactic and semantic annotations and probably data extraction procedures. Frameworks that implement processing chains are for example UIMA[7] and GATE[8]. A platform for processing chains set up in the context of the CLARIN project[9] is WebLicht[10]. WebLicht lists a set of web services from which the user can build a chain to process some input data. Each web service thereby encodes a natural language processing tool in a so-called wrapper. The output of one web service constitutes the input for another one, until the required annotation level is reached. Thus the processing chain has to deal with three levels of formats: the original input and output format of the underlying tool, the processing format to exchange information between the web services, and, if applicable, an additional output format at the end of the processing chain. In this setting the wrapper ensures content-related interoperability to a certain extent by the way the original tool formats are mapped to the exchange format. Among the different wrappers, representational interoperability is ensured by means of the common processing format that needs to strike a balance between the need for a detailed set of linguistic annotations, and the processing efficiency typically required in a web-based approach.

5 Exemplification: handling interoperability in pieces

In the following we will exemplify how the separation of representational and content-related aspects of interoperability allows to cope with the single aspects individually.

Representational interoperability can effectively be achieved by an interlingua approach based on generic exchange formats such as LAF/-GrAF. Providing for content-related interoperability is even a more difficult task. Different resources are usually based on different approaches and often also on different linguistic theories. Concepts that are important for one resource might not appear at all in another one, or they might partly overlap with concepts utilized in a third resource. Since there is no general ontology to be found that the concepts from all theories and approaches can be mapped to, differences have to be tackled on a case-by-case basis. On the other hand, it is often exactly the heterogeneity of the data that brings upon the benefit of utilizing different resources together. However, for many tasks it is not necessary to provide two fully interoperable resources in order to be able to apply them together.

We illustrate the handling of representational interoperability by means of a relational database approach and the handling of content-related interoperability in a study applying a combination of output from different parsers.

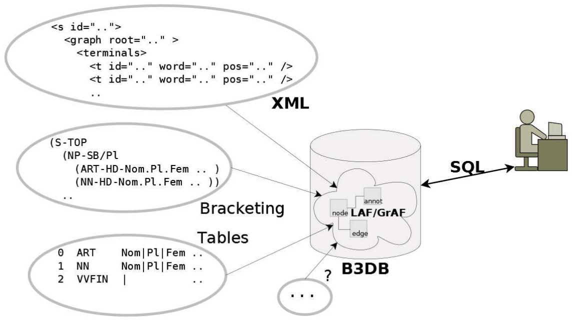
The B3 database (B3DB, Eckart et al. (2010)) is a relational database management system to track workflow aspects and data from computational linguistic projects. The workflow is represented on the macro-layer of the database and the data is structurally represented on its micro-layer. The data structures of the micro-layer are designed on the basis of the LAF/GrAF data model, and are thus generic in the sense that all kinds of different linguistic annotations can be mapped to them, provided these annotations do not exceed the representational power of a graph model. Entering data to the B3DB micro-level thus instantly

[7] http://uima.apache.org/

[8] http://gate.ac.uk/

[9] http://www.clarin.eu/, http://www.clarin-d.de/

[10] Web-based Linguistic Chaining Tool, http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/Main_Page



```
<s id="..">
  <graph root=".." >
    <terminals>
      <t id=".." word=".." pos=".." />
      <t id=".." word=".." pos=".." />
      ..
```

```
(S-TOP
  (NP-SB/Pl
    (ART-HD-Nom.Pl.Fem .. )
    (NN-HD-Nom.Pl.Fem .. ))
  ..
```

```
0  ART    Nom|Pl|Fem ..
1  NN     Nom|Pl|Fem ..
2  VVFIN  |          ..
```

Figure 4: The B3 database as an infrastructure for representational interoperability

provides for representational interoperability of this data.

Figure 4 visualizes the infrastructure setting of the B3DB. Since the data mapping takes place only on the representational level, no explicitly encoded information is lost. A potential user can then conduct SQL-queries on content-related similarities and disagreements of different analyses.

The study by Haselbach et al. (2012) provides an example of task-based handling of interoperability on the level of syntactic analyses. Since Haselbach et al. (2012) are interested in the argument structure of German *nach*-particle verbs, they parsed a web corpus with two different parsing systems: a data-driven state-of-the-art dependency parser trained on news text that provides fully specified analyses (Bohnet, 2010), and a rule-based dependency parser that generates analyses which can be underspecified with respect to head and dependency labels (Schiehlen, 2003). Analysing web data influenced the performance of the systems, but combining information from both systems regarding particle verbs, accusative arguments and dative arguments increased the reliability of the syntactic information and the benefit of the parse results for the overall task. Neither did the actual labels of the different analyses have to be combined, nor was it necessary to resolve all underspecified information. The relevant features for the task were extracted from the output of each system based on its own basic formalism, and only these features were subject to a combination scheme which preferred one or the other analysis, according to the reliability of each parsing system in the respective case.

Such a task-based approach is beneficial in three respects. First it makes it more easy to take the heterogeneity of the information into account and to thus benefit from the differences of the information. Second it supports the handling of information which is specified to a different degree, thus profiting also from underspecified analyses. And third it focuses the effort of the handling of content-related interoperability to the task-related aspects.

6 Conclusion

In this paper, we presented a refined typology of interoperability aspects and argued that the representational aspect of interoperability and its content-related aspects should be treated independently. Regarding content-related aspects, a general and comprehensive solution is not expectable due to the fact that the resources are based on different linguistic theories or approaches. However in many cases such a general or comprehensive solution is not needed to reach a sufficient degree of interoperability for the task at hand. Applying a task-based approach to content-related interoperability reduces complexity to the task-related aspects, and even allows for different combination approaches, depending on the type of task. While it is helpful to have a generic exchange format that provides for representational interoperability in a general fashion, regarding content-related interoperability it might often be more useful to postpone effort of handling it until a specific use case arises.

Acknowledgements

This work was funded by the Deutsche Forschungsgemeinschaft (DFG) in the context of the collaborative research centre SFB 732, project B3.

References

Anders Björkelund, Ozlem Cetinoglu, Richárd Farkas, Thomas Mueller, and Wolfgang Seeker. 2013. (Re)ranking Meets Morphosyntax: State-of-the-art Results from the SPMRL 2013 Shared Task. In *Proceedings of the Fourth Workshop on Statistical Parsing of Morphologically-Rich Languages*, pages 135–145, Seattle, Washington, USA. Association for Computational Linguistics.

Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of COLING 2010*, pages 89–97, Beijing.

Sabine Buchholz and Erwin Marsi. 2006. CoNLL-X Shared Task on Multilingual Dependency Parsing. In *Proceedings of CoNLL-X*, pages 149–164, New York.

Eric Villemonte de la Clergerie, Olivier Hamon, Djamel Mostefa, Christelle Ayache, Patrick Paroubek, and Anne Vilnat. 2008. PASSAGE: from French Parser Evaluation to Large Sized Treebank. In Nicoletta Calzolari (Conference Chair), Khalid Choukri, Bente Maegaard, Joseph Mariani, Jan Odijk, Stelios Piperidis, and Daniel Tapias, editors, *Proceedings of the Sixth International Conference on Language Resources and Evaluation (LREC'08)*, Marrakech, Morocco. European Language Resources Association (ELRA).

Kurt Eberle, Kerstin Eckart, Ulrich Heid, and Boris Haselbach. 2012. A Tool/Database Interface for Multi-Level Analyses. In *Proceedings of the eighth conference on International Language Resources and Evaluation (LREC'12)*, Istanbul, Turkey. European Language Resources Association (ELRA).

Kerstin Eckart, Kurt Eberle, and Ulrich Heid. 2010. An Infrastructure for More Reliable Corpus Analysis. In *Proceedings of the Workshop on Web Services and Processing Pipelines in HLT: Tool Evaluation, LR Production and Validation (LREC'10)*, pages 8–14, Valletta, Malta.

Jonathan G. Fiscus. 1997. A Post-Processing System To Yield Reduced Word Error Rates: Recognizer Output Voting Error Reduction (ROVER). In *IEEE Workshop on Automatic Speech Recognition and Understanding*, pages 347–354.

Jan Hajič et al. 2009. The CoNLL-2009 Shared Task: Syntactic and Semantic Dependencies in Multiple Languages. In *Proceedings of CoNLL 2009: Shared Task*, pages 1–18, Boulder, Colorado.

Boris Haselbach, Kerstin Eckart, Wolfgang Seeker, Kurt Eberle, and Ulrich Heid. 2012. Approximating theoretical linguistics classification in real data: the case of German *nach* particle verbs. In *Proceedings of COLING 2012*, Mumbai.

Ulrich Heid, Helmut Schmid, Kerstin Eckart, and Erhard Hinrichs. 2010. A corpus representation format for linguistic web services: the D-SPIN Text Corpus Format and its relationship with ISO standards. In *Proceedings of LREC-2010, Linguistic Resources and Evaluation Conference*, Malta. [CD-ROM].

Nancy Ide and James Pustejovsky. 2010. What Does Interoperability Mean, anyway? Toward an Operational Definition of Interoperability for Language Technology. In *Proceedings of the Second International Conference on Global Interoperability for Language Resources (ICGL 2010)*, Hong Kong, China.

Nancy Ide and Keith Suderman. 2007. GrAF: A Graph-based Format for Linguistic Annotations. In *Proceedings of the Linguistic Annotation Workshop*, pages 1–8, Prague, Czech Republic. Association for Computational Linguistics.

ISO 12620:2009 Terminology and other language and content resources – Specification of data categories and management of a Data Category Registry for language resources.

ISO 24612:2012 Language resource management – Linguistic annotation framework (LAF).

Mitchell P. Marcus, Beatrice Santorini, and Mary Ann Marcinkiewicz. 1993. Building a Large Annotated Corpus of English: The Penn Treebank. *Comput. Ling.*, 19(2):313 – 330.

Michael Schiehlen. 2003. A Cascaded Finite-State Parser for German. In *Proceedings of EACL 2003*, pages 163–166, Budapest.

Helmut Schmid. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In *Proceedings of the 20th International Conference on Comput ational Linguistics (COLING 2004)*, Geneva, Switzerland.

Manfred Stede and Chu-Ren Huang. 2012. Interoperability and reusability: the science of annotation. *Language Resources and Evaluation*, 46(1):91–94.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2011. Evaluating Dependency Parsing: Robust and Heuristics-Free Cross-Annotation Evaluation. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 385–396, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Reut Tsarfaty, Joakim Nivre, and Evelina Andersson. 2012. Cross-Framework Evaluation for Statistical Parsing. In *Proceedings of the 13th Conference of the European Chapter of the Association for Computational Linguistics*, pages 44–54, Avignon, France. Association for Computational Linguistics.

Hans van Halteren et al. 2001. Improving Accuracy in Word Class Tagging through the Combination of Machine Learning Systems. *Computational Linguistics*, 27(2):199 – 229.

Andreas Witt, Ulrich Heid, Felix Sasaki, and Gilles Sérasset. 2009. Multilingual language resources and interoperability. *Language Resources and Evaluation*, 43(1):1–14.

Florian Zipser and Laurent Romary. 2010. A model oriented approach to the mapping of annotation formats using standards. In *Workshop on Language Resource and Language Technology Standards, LREC 2010*, Valetta, Malta.

Resources, Tools, and Applications at the CLARIN Center Stuttgart

Cerstin Mahlow Kerstin Eckart Jens Stegmann André Blessing
Gregor Thiele Markus Gärtner Jonas Kuhn
Institute for Natural Language Processing (IMS)
University of Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart
firstname.lastname@ims.uni-stuttgart.de

Abstract

This NECTAR track paper (NECTAR: new scientific and technical advances in research) summarizes recent research and curation activities at the CLARIN center Stuttgart. CLARIN is a European initiative to advance research in humanities and social sciences by providing language-based resources via a shared distributed infrastructure. We provide an overview of the resources (i.e., corpora, lexical resources, and tools) hosted at the IMS Stuttgart that are available through CLARIN and show how to access them. For illustration, we present two examples of the integration of various resources into Digital Humanities projects. We conclude with a brief outlook on the future challenges in the Digital Humanities.[1]

1 Introduction

CLARIN-D[2] is the German branch of the European CLARIN initiative[3]. The overall goal is to implement a web-based and center-based infrastructure to facilitate research in the social sciences and humanities. This is achieved by providing linguistic data, tools, and services in an integrated, interoperable, and scalable infrastructure. CLARIN-D is funded by the German Federal Ministry for Education and Research (BMBF).

The Institute for Natural Language Processing (IMS) at the University of Stuttgart is one of currently nine German centers. CLARIN centers undergo thorough external and internal evaluation regarding mostly technical requirements—e.g., metadata, repository system, documentation, legal issues, authentication, and authorization. The IMS was awarded the *Data Seal of Approval* in March 2013 and gained the status of an official CLARIN center in June 2013.[4]

The integration of existing linguistic resources and tools includes efforts towards availability of resources as well as towards the creation and publication of metadata to enable the discovery of resources. All German centers closely collaborate on technical aspects and issues in the curation of language resources. Exchange on the European level is facilitated via the annual CLARIN ERIC conference and specific task forces.

The IMS provides a number of well-established as well as some recently created lexical and corpus resources; it also offers various tools in order to process linguistic data. They are usually made available both as a download package (to be installed and executed locally by the user) and as a web service. The latter is clearly in line with the general CLARIN philosophy of seamless access and usability of resources via the WWW. One particular interest is domain adaptation, resulting in

[1] Page numbers and proceedings footer are added by the organizers. License details: http://creativecommons.org/licenses/by/4.0/

[2] http://www.clarin-d.de.

[3] http://www.clarin.eu.

[4] http://hdl.handle.net/1839/00-DOCS.CLARIN.EU-95

many follow-up questions—e.g., related to the extendability of resources, the feedback of expert users, and the design of pertinent user interfaces. The IMS is involved in the development of several applications and showcases that demonstrate their potential for enabling Digital Humanities research.

In the rest of this NECTAR paper we first provide an overview of the resources developed and hosted at the IMS Stuttgart that are available through CLARIN-D (section 2). In section 3 we present two examples of how to use those resources in actual Digital Humanities projects. Section 4 concludes with a summary and a brief outlook on the future challenges in the Digital Humanities.

2 Resources

We use the term *linguistic resource* in a broad sense. Resources can be text, speech, multimodal corpora, and lexical knowledge bases, but also the tools utilized to create, annotate, and query linguistic information and data collected within experiments or studies. This also includes web services, i.e., tools that can be applied via a web browser and run on servers of the providing organizations. Similarly, important parts of these tools, such as grammars or statistically trained language models are also resources on their own.

The objective of CLARIN, however, is not only to provide resources, but to set up an infrastructure to support the applicability and interaction of these resources. Important aspects are (a) the possibility to find existing resources and to determine whether they fit one's own needs, (b) the possibility to store, access, execute, process, and cite linguistic resources, and (c) the possibility to reproduce experiments or studies based on specific versions of resources. All aspects contribute to the sustainability of the respective resources.

To be able to search for linguistic resources the *Virtual Language Observatory* (VLO)[5] has been set up (van Uytvanck et al., 2012). The faceted browser allows for a search based on free text, but also provides facets which allow users to filter resources by specific features, e.g., by language or resource type. A large number of resources are already listed in the VLO. Current development focuses on improvement of user interaction.

In the VLO, resources are described by their metadata. Since relevant metadata aspects are not easy to be defined a priori for all resource types, CLARIN proposed the flexible *Component MetaData Infrastructure* (CMDI, (Broeder et al., 2012)). In CMDI, metadata schemes reflecting the specific needs of the different resource types can be created by common means. This way CLARIN also helps to improve the documentation of resources, since metadata are one prerequisite for a resource to become part of the CLARIN infrastructure. For all resources we present in this paper, CMDI descriptions have been created or enriched.

The metadata and also the resource itself can be stored and made available via data repositories. Such repositories are hosted at the CLARIN centers. The metadata stored can be automatically harvested via the *Open Archives Initiative Protocol for Metadata Harvesting* (OAI-PMH). The term *harvesting* means that automatic collector services—e.g., from the VLO—regularly access a pertinent service exposed by the repository and copy all the disseminated metadata. Therefore it is not necessary to explicitly register resources at the VLO or to commit changes there. Since metadata do not contain any part of the resource itself and usually do not contain sensitive information, the CLARIN requirements stipulate that they have to be free to read and free to harvest via the web. The metadata of all the resources presented here are freely harvestable via the OAI-PMH service of the IMS repository.[6]

While metadata are freely available, this is often not the case for the resources themselves. We usually find a legal limbo with respect to language resources. However, we can distinguish

[5] http://catalog.clarin.eu/vlo/.

[6] http://clarin04.ims.uni-stuttgart.de/oaiprovider/oai?verb=ListRecords&metadataPrefix=cmdi.

(a) resources which can be freely distributed, (b) resources which are restricted to research purposes, and (c) resources with additional restrictions. Free resources, for example, can provide a download link in the VLO. Restricted resources, however, have to be adressed via (a) specific legal licensing schemes and (b) an authentication and authorization strategy that respects given restrictions. CLARIN adresses the former by providing licensing templates for resource creators corresponding to the respective categories mentioned above.[7]

The solution to the latter is the implementation of web-based single-sign on via authentication and authorization infrastructures (AAI) using the Shibboleth[8] technology. For example, the University of Stuttgart is a member of the DFN-AAI[9] federation (identity providers), so all researchers and students can login to CLARIN-D web applications using their University of Stuttgart credentials[10]. Additionally, the IMS CLARIN center has registered as a service provider in the DFN-AAI federation. We are currently in the process of reorganizing the mode of access via the IMS repository to make use of the federated Shibboleth-based approach.

Due to fast and constant development of resources, it is necessary to not only cite publications about resources, but also the datasets or resources themselves. This allows for a more precise citation and also supports the reproducibility of findings, when the exact version of the applied resources can be identified. Within CLARIN *persistent identifiers* (PIDs) are registered for data and metadata alike. These PIDs act as links to the sets of metadata, the download of the resource, and a landing page of the resource. They can be resolved in the address line of a browser, similar to DOIs. The advantage of PIDs is that they are not supposed to change. If a website moves, and thus the previous URL becomes invalid, it is not possible to find all places on the web that provided a link to the original website. If, however, all references are made to the PID, only the PID needs to be realigned with the new address of the web page, and the page and the resource remain accessible. It is a prerequisite for a resource in the CLARIN infrastructure to be identifiable by a PID. The PIDs have to be part of the CMDI metadata provided and can be registered via services provided by members of the EPIC consortium[11]. The IMS uses the service offered by GWDG.[12]

We now present resources hosted or created at the CLARIN-D center Stuttgart. For all resources metadata have been created and PIDs have been registered[13]. Most of the provided web services are also accessible via WebLicht, the CLARIN-D Web-based Linguistic Chaining Tool[14].

2.1 Corpora

The *Huge German Corpus* (HGC)[15] is a collection of German texts (newspaper and law texts) of about 204 million tokens including punctuation in 12.2 million sentences (about 180 million "real" words). The corpus was automatically segmented into sentences, and lemmatized and part-of-speech tagged by the TreeTagger (Schmid, 1994) using the STTS tagset. The corpus is partly based on data taken from the European Corpus Intitiative Multilingual Corpus I (EMI/MCI). This corpus is now also maintained by the IMS.

SdeWaC[16] is based on the deWaC web corpus of the WaCky-Initative[17]. It contains parsable sentences from deWaC documents of the .de domain. (Faaß and Eckart, 2013) SdeWaC is limited to the sentence context. The sentences were

[7] https://kitwiki.csc.fi/twiki/bin/view/FinCLARIN/KielipankkiLicenceCategories.

[8] https://shibboleth.net/.

[9] https://www.aai.dfn.de/.

[10] This also works on the European level to access services provided by members of the CLARIN Service Provider Federation.

[11] http://www.pidconsortium.eu/.

[12] http://handle.gwdg.de:8080/pidservice/.

[13] We thus give the respective PIDs for each resource

[14] http://weblicht.sfs.uni-tuebingen.de/weblichtwiki/index.php/Main_Page.

[15] http://hdl.handle.net/11858/00-247C-0000-0022-F7B4-4.

[16] http://hdl.handle.net/11858/00-247C-0000-0022-F7BA-7.

[17] http://wacky.sslmit.unibo.it

sorted and duplicate sentences within the same domain name were removed. In addition, some heuristics based on Quasthoff et al. (2006) have been applied. SdeWaC-v3 comes in two formats: (a) one sentence per line and (b) one token per line including part-of-speech and lemma annotation.

The *TIGER corpus*[18] is a German newspaper corpus enriched with part-of-speech annotation, morphological and lemma information and syntactic structure (Brants et al., 2004). Versioning is an important aspect of the proper modelling of linguistic resources via metadata. We use the TIGER corpus as testbed for exploring different possibilities in this respect. Questions related to versioning highlight aspects of the more general question of how to deal with relations among resources.

The *Discourse Information Radio News Database for Linguistic Analysis* (DIRNDL)[19] is a corpus resource based on hourly broadcast German radio news (Eckart et al., 2012). The textual version of the news is annotated with syntactic information. Syntactic phrases are labeled with information status categories (given vs. new information). The speech version is prosodically annotated, i.e., with pitch accents and prosodic phrase boundaries. The textual and the speech version slightly deviate from each other due to slips of the tongue, fillers, and minor modifications. A (semi-automatic) linking of the two versions was carried out and the results were stored inside the database. With the help of these newly established links, all annotation layers can be accessed for exploring the relations between prosody, syntax, and information status.

GECO[20] has been created to investigate phonetic convergence in German spontaneous speech (Schweitzer and Lewandowski, 2013). The database consists of 46 dialogs of approximately 25 minutes length each, between previously unacquainted female subjects. Of these 46 dialogs, 22 dialogs were in a unimodal setting, where participants could not see each other, while the remaining 24 dialogs were recorded with subjects facing each other. The database was automatically annotated on the segment, syllable, and word levels using forced alignment with manually generated orthographic transcriptions.

2.2 Lexical Resources

The *German Logical Metonymy Database*[21] is the result of a corpus study for German verbs (*anfangen (mit)* ('to start (with)'), *aufhören (mit) ('to stop')*, *beenden* ('to end'), *beginnen (mit)* ('to begin (with)'), *genießen* ('to enjoy')), based on data obtained from the deWaC corpus. (Zarcone and Rued, 2012) The database contains 2'661 metonymies and 1'886 long forms with two expert annotations.

The *IMSLex dictionary database*[22] covers information on inflection, word formation, and valence for several ten thousand German base forms. (Fitschen, 2004)

The *German Verb Subcategorization Database*[23] contains verb subcategorization information from German MATE dependency parses of SdeWaC. The subcategorization database is represented in a compact but linguistically detailed and flexible format, comprising various aspects of verb information, complement information and sentence information, within a one-line-per-clause style. The SdeWaC subcategorization database comprises 73'745'759 lines (representing the number of extracted target verb clauses), resulting in 6.3 GB in compressed format.

2.3 Tools

For all tools we have CMDI data for a downloadable local executable version and for the webser-

[18] `http://hdl.handle.net/11858/00-247C-0000-000D-FFB5-1`.

[19] `http://hdl.handle.net/11858/00-247C-0000-0022-F7B2-8`.

[20] `http://hdl.handle.net/11858/00-247C-0000-0023-5137-2`.

[21] `http://hdl.handle.net/11858/00-247C-0000-0023-5147-D`.

[22] `http://hdl.handle.net/11858/00-247C-0000-0022-F7B8-B`.

[23] `http://hdl.handle.net/11858/00-247C-0000-0023-8BCD-01`.

vice version we provide for the CLARIN-D infrastrucure.

SMOR[24] is a German finite-state morphology implemented in the SFST programming language (Schmid et al., 2004). It is integrated in the CLARIN-D infrastructure by means of a web service, there is also an SMOR download tool.

We deployed a new morphology web service called *Stuttgart Morphology* for German which derives the morphological analysis from *RFTagger's* (see below) internal analysis.

The *TreeTagger*[25] is a tool for annotating text with part-of-speech and lemma information (Schmid, 1994). We deployed a new version (i.e., TreeTagger2013) of TreeTagger as web service implemented in Java. The new release achieves better performance.

RFTagger[26] is a part-of-speech tagger providing also morphological information and makes use of fine-grained tagsets (Schmid and Laws, 2008). The RFTagger web service is implemented in Java.

We deployed a new *NER web service for German*[27] based on the Conditional Random Field-based Stanford Named Entity Recognizer by Finkel and Manning (2009) which includes semantic generalization information from large untagged German corpora. (Faruqui and Padó, 2010)

BitPar[28] is a parser for highly ambiguous probabilistic context-free grammars (such as treebank grammars). BitPar uses bit-vector operations to speed up the basic parsing operations by parallelization (Schmid, 2004). It is integrated in the CLARIN-D infrastructure by means of a web service.

The *Bohnet Toolchain*[29] includes a lemmatizer, a part-of-speech tagger, a morphological tagger, and a state-of-the-art dependency parser for German (Bohnet, 2010). We deployed a new version of the Bohnet Toolchain web service. The new release includes some bugfixes and performance improvement. The Bohnet Toolchain is available as *MATE Tools* for download[30]; additionally, it is deployed at the High Performance Computing Center Garching as web service.

The *Interactive Text Analysis Tool* is a prototype system based on RESTful web services implementing an interactive relation extraction system (Blessing et al., 2012). It comprises a retrainable web service on top of a web service processing chain (tokenizer, tagger, parser) which merges automatic linguistic annotation on several levels. The system aims to demonstrate the dynamic interaction between such software and human users from the Digital Humanities.

The *TIGERSearch*[31] software helps to explore linguistically annotated texts. It is a specialized search engine for retrieving information from a database of graph structures (treebank) (Lezius, 2002). The text corpus to be searched by TIGERSearch must have been annotated beforehand, e.g., with grammatical analyses (syntax trees).

3 Case Studies

3.1 ICARUS

ICARUS[32] is a search and visualization tool that primarily targets dependency trees (Gärtner et al., 2013). It allows users to search dependency treebanks given a variety of constraints, including

[24] http://hdl.handle.net/11858/00-247C-0000-0022-F7BC-3.

[25] http://hdl.handle.net/11858/00-247Z-0000-0007-5EC0-4.

[26] http://hdl.handle.net/11858/00-247C-0000-000D-FFB4-3.

[27] http://www.nlpado.de/~sebastian/software/ner_german.shtml.

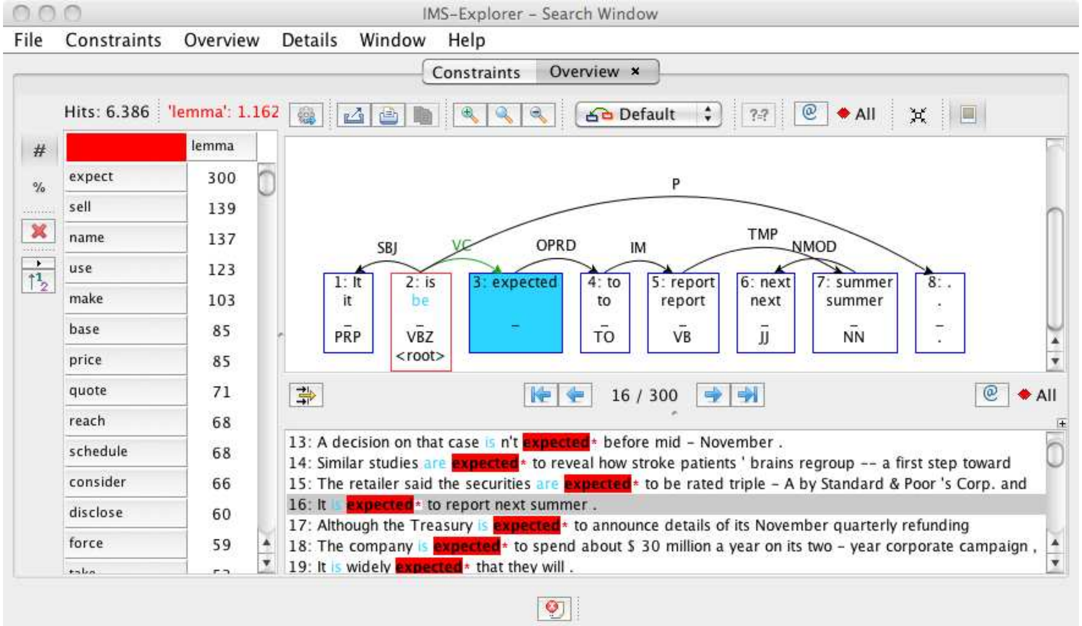
[28] http://hdl.handle.net/11858/00-247C-0000-0022-F7B0-C.

[29] http://hdl.handle.net/11858/00-247Z-0000-0007-6A0D-E.

[30] http://code.google.com/p/mate-tools

[31] http://hdl.handle.net/11858/00-247C-0000-0022-F7BE-0.

[32] http://hdl.handle.net/11858/00-247C-0000-0022-F7B6-F.



Hits: 6.386 'lemma': 1.162

	lemma
expect	300
sell	139
name	137
use	123
make	103
base	85
price	85
quote	71
reach	68
schedule	68
consider	66
disclose	60
force	59

13: A decision on that case is n't expected* before mid - November .
14: Similar studies are expected* to reveal how stroke patients ' brains regroup -- a first step toward
15: The retailer said the securities are expected* to be rated triple - A by Standard & Poor 's Corp. and
16: It is expected* to report next summer .
17: Although the Treasury is expected* to announce details of its November quarterly refunding
18: The company is expected* to spend about $ 30 million a year on its two - year corporate campaign ,
19: It is widely expected* that they will .

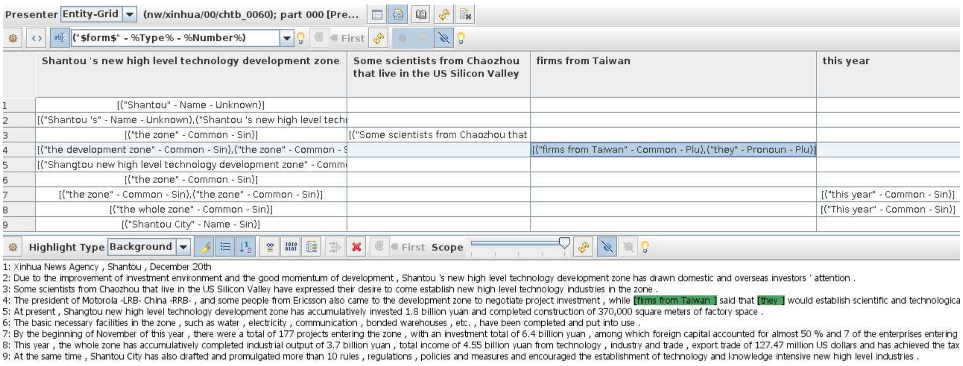
Figure 1: Passive constructions in the treebank grouped by lemma and sorted by frequency.

searching for particular subtrees. Emphasis has been placed on functionality that makes it possible for users to switch back and forth between a high-level, aggregated view of the search results and browsing of particular corpus instances. Users can create queries graphically and results will be returned as frequency lists and tables (i.e., quantitatively) as well as qualitatively by connecting the statistics to the matching sentences and allowing the user to browse them graphically. The first application using ICARUS is a search engine to explore dependency trees in treebanks as shown in Figure 1.

ICARUS provides plugins for the integration of existing tools or pipelines like the Bohnet Toolchain. So far, two additional applications have been developed: ICE, the *ICARUS Coreference Explorer* (Gärtner et al., 2014), and a graphical interface for automatic error mining of annotation in corpora (Thiele et al., 2014). Both applications use annotated corpora and make use of the general ICARUS features.

ICE is an interactive tool to browse and search coreference annotation. The annotation can be displayed as tree, as entity grid, or as text. Figure 2 shows the entity grid with the predicted annotations and the complete text. Different annotations of the same text can be compared, thus facilitating evaluation. Two usergroups are in focus: NLP developers designing coreference resolution systems—here ICE serves as interactive diagnosis and evaluation tool towards a gold standard—and corpus linguists—here ICE serves as research instrument. The built-in search engine of ICARUS is adapted to allow queries over sets of documents to actually allow searching a corpus. ICE is the first graphical coreference exploration tool offering three different visualizations and thus supporting various user needs.

The ICARUS error mining extension is a tool for finding annotation errors and inconsistencies in large annotated corpora. It implements the automatic error mining algorithms proposed in (Dickinson and Meurers, 2003) and (Boyd et al., 2008) for part-of-speech and dependency annotations, respectively. The tool allows the user to find potential annotation errors by presenting a list of candidates generated by the algorithm. It presents statistics on the label distribution of the candidate and connects the error candidate with the sentences in the corpus in which it occurs. The user can then decide if the annotation is indeed erroneous and needs to be corrected. Figure 3 illustrates the candidate list for the part-of-



Presenter Entity-Grid (nw/xinhua/00/chtb_0060); part 000 [Pre...

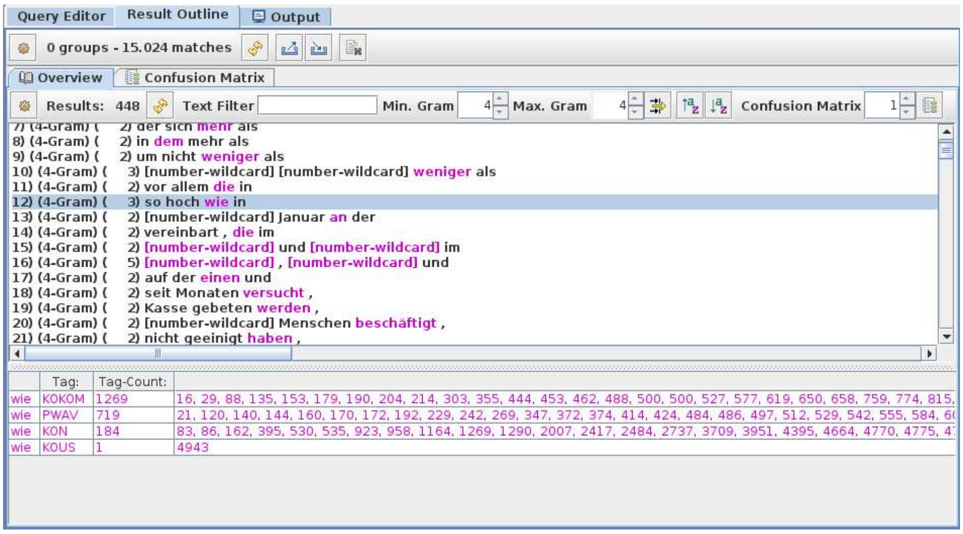
("$form$" - %Type% - %Number%)

	Shantou 's new high level technology development zone	Some scientists from Chaozhou that live in the US Silicon Valley	firms from Taiwan	this year
1	[{"Shantou" - Name - Unknown}]			
2	[{"Shantou 's" - Name - Unknown},{"Shantou 's new high level techı			
3	[{"the zone" - Common - Sin}]	[{"Some scientists from Chaozhou that		
4	[{"the development zone" - Common - Sin},{"the zone" - Common - S		[{"firms from Taiwan" - Common - Plu},{"they" - Pronoun - Plu}]	
5	[{"Shangtou new high level technology development zone" - Comm			
6	[{"the zone" - Common - Sin}]			
7	[{"the zone" - Common - Sin},{"the zone" - Common - Sin}]			[{"this year" - Common - Sin}]
8	[{"the whole zone" - Common - Sin}]			[{"This year" - Common - Sin}]
9	[{"Shantou City" - Name - Sin}]			

Highlight Type Background

1: Xinhua News Agency , Shantou , December 20th
2: Due to the improvement of investment environment and the good momentum of development , Shantou 's new high level technology development zone has drawn domestic and overseas investors ' attention .
3: Some scientists from Chaozhou that live in the US Silicon Valley have expressed their desire to come establish new high level technology industries in the zone .
4: The president of Motorola -LRB- China -RRB- , and some people from Ericsson also came to the development zone to negotiate project investment , while [firms from Taiwan] said that [they] would establish scientific and technologica
5: At present , Shangtou new high level technology development zone has accumulatively invested 1.8 billion yuan and completed construction of 370,000 square meters of factory space .
6: The basic necessary facilities in the zone , such as water , electricity , communication , bonded warehouses , etc. , have been completed and put into use .
7: By the beginning of November of this year , there were a total of 177 projects entering the zone , with an investment total of 6.4 billion yuan , among which foreign capital accounted for almost 50 % and 7 of the enterprises entering
8: This year , the whole zone has accumulatively completed industrial output of 3.7 billion yuan , total income of 4.55 billion yuan from technology , industry and trade , export trade of 127.47 million US dollars and has achieved the tax
9: At the same time , Shantou City has also drafted and promulgated more than 10 rules , regulations , policies and measures and encouraged the establishment of technology and knowledge intensive new high level industries .

Figure 2: Entity grid over the predicted clustering in the example document.



0 groups - 15.024 matches

Results: 448 Text Filter Min. Gram 4 Max. Gram 4 Confusion Matrix 1

```
7) (4-Gram) (    2) der sich mehr als
8) (4-Gram) (    2) in dem mehr als
9) (4-Gram) (    2) um nicht weniger als
10) (4-Gram) (    3) [number-wildcard] [number-wildcard] weniger als
11) (4-Gram) (    2) vor allem die in
12) (4-Gram) (    3) so hoch wie in
13) (4-Gram) (    2) [number-wildcard] Januar an der
14) (4-Gram) (    2) vereinbart , die im
15) (4-Gram) (    2) [number-wildcard] und [number-wildcard] im
16) (4-Gram) (    5) [number-wildcard] , [number-wildcard] und
17) (4-Gram) (    2) auf der einen und
18) (4-Gram) (    2) seit Monaten versucht ,
19) (4-Gram) (    2) Kasse gebeten werden ,
20) (4-Gram) (    2) [number-wildcard] Menschen beschäftigt ,
21) (4-Gram) (    2) nicht geeinigt haben ,
```

	Tag:	Tag-Count:	
wie	KOKOM	1269	16, 29, 88, 135, 153, 179, 190, 204, 214, 303, 355, 444, 453, 462, 488, 500, 500, 527, 577, 619, 650, 658, 759, 774, 815,
wie	PWAV	719	21, 120, 140, 144, 160, 170, 172, 192, 229, 242, 269, 347, 372, 374, 414, 424, 484, 486, 497, 512, 529, 542, 555, 584, 6(
wie	KON	184	83, 86, 162, 395, 530, 535, 923, 958, 1164, 1269, 1290, 2007, 2417, 2484, 2737, 3709, 3951, 4395, 4664, 4770, 4775, 4
wie	KOUS	1	4943

Figure 3: N-gram view of the error mining application based on ICARUS.

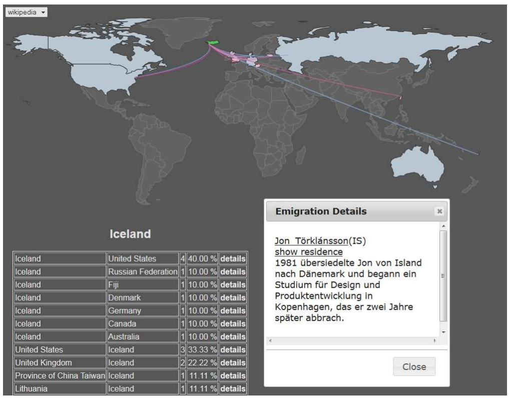
speech error mining algorithm. In the upper part, candidate tokens are shown with their surrounding context. In the lower part, a label distribution for the candidate token is shown. Clicking on the candidate or on one of the labels opens the corpus browser with the appropriate sentences. The user can inspect the relevant sentences and decide if there are erroneous annotations. This tool is thus intended to support corpus creation and curation, the processing step before corpus linguists may actually query the corpus to answer dedicated research questions. Annotations to be checked for errors and inconsistencies may stem from both manual or automatic processing.

3.2 Textual Emigration Analysis

Textual Emigration Analysis (TEA)[33] is a web-based application that transforms raw textual data into a graphical display of migration source and target countries. (Blessing and Kuhn, 2014) The tool serves as showcase demonstrating the use of language technology to support research in the humanities. It is used in ongoing research projects. For instance, from the sentence "*Erika Lust grew up in Kazakhstan and emigrated to Germany in 1989.*" we can extract the triple `emigrate(Erika Lust, Kazakhstan, Germany)` by using several web services (tokenizer, TreeTagger, Bohnet Toolchain, NER) provided by the CLARIN-D in-

[33] `http://clarin01.ims.uni-stuttgart.de/geovis/showcase.html`.

frastructure. Those triples are then visualized on a map.



Iceland

Iceland	United States	4	40.00 %	details
Iceland	Russian Federation	1	10.00 %	details
Iceland	Fiji	1	10.00 %	details
Iceland	Denmark	1	10.00 %	details
Iceland	Germany	1	10.00 %	details
Iceland	Canada	1	10.00 %	details
Iceland	Australia	1	10.00 %	details
United States	Iceland	3	33.33 %	details
United Kingdom	Iceland	2	22.22 %	details
Province of China Taiwan	Iceland	1	11.11 %	details
Lithuania	Iceland	1	11.11 %	details

Emigration Details

Jon Törklánsson(IS)
show residence
1981 übersiedelte Jon von Island nach Dänemark und begann ein Studium für Design und Produktentwicklung in Kopenhagen, das er zwei Jahre später abbrach.

Figure 4: Screenshot of the user interface of the TEA web application showing emigration from and to *Iceland*.

TEA is intended to be used by humanities scholars; it offers a visual impression of the aggregated data as well as means for qualitative inspection of the underlying sources. Figure 4 shows a screenshot of the TEA-user-interface. The user selects a country (*Iceland* in the given example) to get the list of related migration events. The details of the row *Iceland-Denmark-1* are selected and the user sees the textual source which describes that Jon Törklánsson emigrated from Iceland to Denmark. This way, the graphical visualization is more transparent, which leads to a better acceptance of automatic tools in the humanities; users can always refer to the corresponding sources.

4 Summary and Perspectives

In this paper, we presented the resources (corpora, lexical resources, and tools) provided at the CLARIN center at the IMS Stuttgart. We created CMDI metadata and registered PIDs for all resources, so they can be discovered and accessed by users. As examples for the use of those resources in actual applications, we elaborated on two use cases, the ICARUS search and visualization tool and the Textual Emigration Analysis to be used in Digital Humanities research.

On a technical level, an important focus for future work at the Stuttgart CLARIN center is on metadata: Currently, relations between resources are not covered by the provided CMDI data. Similarly, there is no agreed upon standard to describe different versions of a resource due to improvements of tools, extension or extraction of corpora, or the like. CMDI in general offers to describe relations and versions, however, various possibilities could be used. The use in the VLO requires some information and sets some costraints, but consistent procedures are still missing. For example we can register a PID for a resource and a PID for the respective CMDI description, but we cannot define which is depending on which. As mentioned before, we use the TIGER corpus as testbed for versioning and the creation of corresponding metadata to hopefully develop a proposal for general use.

Taking a broader Digital Humanities perspective, experience shows that an operational technical infrastructure is an important ingredient for innovative avenues of research, but there are remaining methodological challenges that cannot be resolved on a purely technical level. It is very important to engage in open-minded interdisciplinary collaborations and learn to better understand each other's working assumption and methodological conventions. The IMS is involved in several such interdisciplinary projects using the CLARIN-D infrastructure and the resources provided, and contributing to the formation of a Digital Humanities methodology. In the BMBF-funded project "e-Identity", a large corpus of newspaper texts from Austria, Germany, Ireland, France, the UK, and the USA is investigated with respect to national identities in critical political situations after the Cold War (Kolb et al., 2009; Blessing et al., 2013; Kliche et al., 2014). In the BMBF-funded project "ePoetics",[34] hermeneutic and algorithmic methods are combined to investigating a corpus of German scholarly aesthetics and poetics from 1770 to 1960 (Richter, 2010). The CLARIN center also collaborates closely with the infrastructure project of SFB 732 "Incremental specification in con-

[34] `http://www.epoetics.de`.

text",[35] a joint effort of theoretical and computational linguistics in which corpus resources and analysis tools play a central role. In the third funding period, the SFB focuses on the generalization of models and theories to non-canonical data types and phenomena and aims to build up a large collection of annotated corpora, adopting a "silver standard" approach of transparent and quality-controlled automatic annotation.

With the recent advances in computational linguistics and language technology, including machine learning paradigms that can be easily extended beyond a linguistically oriented approach to large text collections, there is no doubt about the great potential lying in these techniques for the broader Digital Humanities. But to intergrate them effectively with the established body of knowledge in the humanities and social sciences, the field needs a more systematic methodology that breaks down analytical processes into building blocks whose "deeper" functionality is transparent to the users in the humanities, so they are in a position to make their own critical assessment of the reliability of a particular component or component chain—and arrange for adjustments as necessary. Crucially, the meta-architecture to be established should include best practices for non-computational intermediate steps too, which are required to bridge the methodological gap between data-based empirical results and higher-level disciplinary research questions. Ultimately, Digital Humanities scholars should feel fully competent to draw upon a flexible methodological toolbox so they can try backing up any partial results from one component with evidence obtained from other sources, make informed adjustments to the components, or attempt an entirely different way of approaching the available information sources.

In other words, the mid- to long-term goal should not have IT specialists optimize a tool chain for fully automatic analysis so as to achieve the best possible performance for some specified task—which is bound to be imperfect for any non-trivial question anyway, thus requiring a responsible integration into higher-level research questions. The Digital Humanities should rather aim to create transparency within a complex multi-purpose network of interacting information sources of variable quality or reliability—in plain extension of the classical competences humanities scholars have always had regarding approaches to their object of study. Contrary to the assumptions one can make about the typical users in a standard web-oriented application scenario of language technology and visual analytics (where users rarely have any philological or other meta-level attachment to the text basis from which they are seeking information), humanities scholars have far-reaching competences and intuitions about their objects of study and their sources. This makes the goal of developing an interactive framework for a network of knowledge sources a promising endeavor, drawing on techniques for aggregation, diagnostic and explorative visualization, quantitative analysis and linking back to data instances and (re-)annotation tools, but crucially also including "soft" non-technical components, i.e., theoretically informed steps of analysis and reflection.

Acknowledgements

This work has been supported by the German Federal Ministry of Education and Research (BMBF) in the CLARIN-D project, and the Ministry of Science, Research, and Art (MWK) of the state of Baden-Württemberg.

The authors would like to thank Heike Zinsmeister who made significant contributions to the Stuttgart CLARIN center during her time in Stuttgart, and Anders Björkelund and Wolfgang Seeker for their advice and contributions, in particular in the context of the ICARUS tool.

References

André Blessing and Jonas Kuhn. 2014. Textual emigration analysis. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan

[35] www.uni-stuttgart.de/linguistik/sfb732/.

Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 2089–2093, Paris. European Language Resources Association (ELRA).

André Blessing, Jens Stegmann, and Jonas Kuhn. 2012. SOA meets relation extraction: Less may be more in interaction. In *Proceedings of the Workshop on Service-oriented Architectures (SOAs) for the Humanities: Solutions and Impacts, Digital Humanities*, pages 6–11.

André Blessing, Jonathan Sonntag, Fritz Kliche, Ulrich Heid, Jonas Kuhn, and Manfred Stede. 2013. Towards a tool for interactive concept building for large scale analysis in the humanities. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 55–64, Stroudsburg, PA. Association for Computational Linguistics.

Bernd Bohnet. 2010. Top accuracy and fast dependency parsing is not a contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics (Coling 2010)*, pages 89–97, Beijing, China. Coling 2010 Organizing Committee.

Adriane Boyd, Markus Dickinson, and Detmar Meurers. 2008. On Detecting Errors in Dependency Treebanks. *Research on Language and Computation*, 6(2):113–137.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620.

Daan Broeder, Menzo Windhouwer, Dieter van Uytvanck, Twan Goosen, and Thorsten Trippel. 2012. CMDI: a Component Metadata Infrastructure. In *Proceedings of the Workshop on Describing Language Resources with Metadata (LREC'12)*, Paris. European Language Resources Association (ELRA).

Markus Dickinson and W. Detmar Meurers. 2003. Detecting Errors in Part-of-Speech Annotation. In *Proceedings of the 10th Conference of the European Chapter of the Association for Computational Linguistics (EACL-03)*, pages 107–114, Stroudsburg, PA. Association for Computational Linguistics.

Kerstin Eckart, Arndt Riester, and Katrin Schweitzer. 2012. A Discourse Information Radio News Database for Linguistic Analysis. In Christian Chiarcos, Sebastian Nordhoff, and Sebastian Hellmann, editors, *Linked Data in Linguistics. Representing and Connecting Language Data and Language Metadata*, pages 65–75. Springer, Heidelberg.

Gertrud Faaß and Kerstin Eckart. 2013. SdeWaC—A Corpus of Parsable Sentences from the Web. In Iryna Gurevych, Chris Biemann, and Torsten Zesch, editors, *Language Processing and Knowledge in the Web*, volume 8105 of *Lecture Notes in Computer Science*, pages 61–68. Springer, Heidelberg.

Manaal Faruqui and Sebastian Padó. 2010. Training and evaluating a german named entity recognizer with semantic generalization. In *Proceedings of KONVENS 2010*, Saarbrücken.

Jenny Rose Finkel and Christopher D. Manning. 2009. Nested named entity recognition. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, pages 141–150, Stroudsburg, PA. Association for Computational Linguistics.

Arne Fitschen. 2004. *Ein computerlinguistisches Lexikon als komplexes System.* AIMS // Arbeitspapiere des Instituts für Maschinelle Sprachverarbeitung, Lehrstuhl für Computerlinguistik, Universität Stuttgart. Universität Stuttgart, Institut für maschinelle Sprachverarbeitung.

Markus Gärtner, Gregor Thiele, Wolfgang Seeker, Anders Björkelund, and Jonas Kuhn. 2013. Icarus—an extensible graphical search tool for dependency treebanks. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics: System Demonstrations*, pages 55–60, Stroudsburg, PA. Association for Computational Linguistics.

Markus Gärtner, Anders Björkelund, Gregor Thiele, Wolfgang Seeker, and Jonas Kuhn. 2014. Visualization, search, and error analysis for coreference annotations. In *Proceedings of the 52nd Conference of the Association for Computational Linguistic: System Demonstrations*, Stroudsburg, PA. Association for Computational Linguistics.

Fritz Kliche, André Blessing, Jonathan Sonntag, and Ulrich Heid. 2014. The e-identity exploration workbench. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14)*, pages 691–697, Paris. European Language Resources Association (ELRA).

Peter Kolb, Amelie Kutter, Cathleen Kantner, and Manfred Stede. 2009. Computer- und korpuslinguistische Verfahren für die Analyse massenmedialer politischer Kommunikation: Humanitäre und militärische Interventionen im Spiegel der Presse. In Wolfgang Hoeppner, editor, *Technischer Bericht Nr. 2009-01. GSCL-Symposium Sprachtechnologie* *und eHumanities*, pages 62–71, Duisburg. Universität Duisburg-Essen.

Wolfgang Lezius. 2002. TIGERSearch – Ein Suchwerkzeug für Baumbanken. In Stephan Busemann, editor, *Proceedings der 6. Konferenz zur Verarbeitung natürlicher Sprache (KONVENS 2002)*, Saarbrücken.

Uwe Quasthoff, M. Richter, and C. Biemann. 2006. Corpus portal for search in monolingual corpora. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, pages 1799–1802, Paris. European Language Resources Association (ELRA).

Sandra Richter. 2010. *A History of Poetics: German Scholarly Aesthetics and Poetics in International Context, 1770-1960.* De Gruyter, Berlin, New York.

Helmut Schmid and Florian Laws. 2008. Estimation of conditional probabilities with decision trees and an application to fine-grained POS tagging. In *Proceedings of the 22nd International Conference on Computational Linguistics (COLING 2008)*, pages 777–784, Stroudsburg, PA. Coling 2008 Organizing Committee.

Helmut Schmid, Arne Fitschen, and Ulrich Heid. 2004. Smor: A german computational morphology covering derivation, composition, and inflection. In *Proceedings of the Fourth International Conference on Language Resources and Evaluation (LREC'04)*, Paris. LREC, European Language Resources Association (ELRA).

Helmut Schmid. 1994. Probabilistic Part-of-Speech Tagging Using Decision Trees. In *Proceedings of International Conference on New Methods in Language Processing (NeMLaP 1994)*, pages 44–49, Manchester,UK.

Helmut Schmid. 2004. Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In *Proceedings of the 20th International Conference on Computational Linguistics (COLING 2004)*, Stroudsburg, PA.

Antje Schweitzer and Natalie Lewandowski. 2013. Convergence of articulation rate in spontaneous speech. In *Proceedings of the 14th Annual Conference of the International Speech Communication Association (Interspeech 2013)*, pages 525–529.

Gregor Thiele, Wolfgang Seeker, Markus Gärtner, Anders Björkelund, and Jonas Kuhn. 2014. A graphical interface for automatic error mining in corpora. In *Proceedings of the Demonstrations at the 14th Conference of the European Chapter of the Association for Computational Linguistics*, pages 57–60, Stroudsburg, PA. Association for Computational Linguistics.

Dieter van Uytvanck, Herman Stehouwer, and Lari Lampen. 2012. Semantic metadata mapping in practice: the Virtual Language Observatory. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Paris. European Language Resources Association (ELRA).

Alessandra Zarcone and Stefan Rued. 2012. Logical metonymies and qualia structures: an annotated database of logical metonymies for German. In Nicoletta Calzolari, Khalid Choukri, Thierry Declerck, Mehmet Uğur Doğan, Bente Maegaard, Joseph Mariani, Jan Odijk, and Stelios Piperidis, editors, *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, Paris. European Language Resources Association (ELRA).

Centering Theory in natural text: a large-scale corpus study

Annemarie Friedrich **Alexis Palmer**
Department of Computational Linguistics
Saarland University, Saarbrücken, Germany
{afried,apalmer}@coli.uni-saarland.de

Abstract

We present an extensive corpus study of Centering Theory (CT), examining how adequately CT models coherence in a large body of natural text. A novel analysis of transition bigrams provides strong empirical support for several CT-related linguistic claims which so far have been investigated only on various small data sets. The study also reveals genre-based differences in texts' degrees of entity coherence. Previous work has shown unsupervised CT-based coherence metrics to be unable to outperform a simple baseline. We identify two reasons: 1) these metrics assume that some transition types are more coherent and that they occur more frequently than others, but in our corpus the latter is not the case; and 2) the original sentence order of a document and a random permutation of its sentences differ mostly in the fraction of entity-sharing sentence pairs, exactly the factor measured by the baseline.

1 Introduction

Centering Theory (CT) models the degree of local coherence between adjacent utterances within paragraphs with respect to patterns of entity mentions and the choice of referring expressions (Grosz et al., 1995). CT regards a text as a sequence of utterances $U_1, U_2, \ldots, U_n$. The entities mentioned (*realized*) in an *utterance* U_i are referred to as *centers* and make up its set of *forward-looking centers* $CF(U_i)$, which are ranked according to their salience, i.e., how likely they are to be mentioned in the following utterance. Each utterance is assigned a single *backward-looking center* $CB(U_i)$, defined as the highest-ranked element of $CF(U_{i-1})$ also realized in U_i, and a *preferred center* $CP(U_i)$, the highest-ranked center of U_i. CT identifies different types of transitions between adjacent utterances and assumes that the types have different degrees of coherence. We define these as in Table 1 (following Brennan et al. (1987) and Kameyama (1986)), with NoCB as the case U_i and U_{i-1} have no shared center, so U_i has no CB.

Contributions. We present the largest corpus study of CT to date, confirming and consolidating previous results by investigating multiple predictions of the theory using a uniform implementation of CT over a large amount (14096 sentences) of natural text. CT has inspired various automatic methods for measuring coherence (Lapata and Barzilay (2005), Elsner and Charniak (2011), among others). In this paper we aim not to improve upon these methods, but rather to better understand when and why they work and what the reasons are for their limitations. Our main finding is that analysis of natural text, which can be assumed to be coherent, fails to support some of the predictions of CT which inform automatic coherence evaluation methods. Many adjacent sentences do not mention the same entities, and there is no clear preference for certain

	COHERENCE $CB(U_i) = CB(U_{i-1})$	$\neg$ COHERENCE $CB(U_i) \neq CB(U_{i-1})$
SALIENCE $CB(U_i) = CP(U_i)$	CONTINUE	SMOOTH-SHIFT
$\neg$ SALIENCE $CB(U_i) \neq CP(U_i)$	RETAIN	ROUGH-SHIFT

$CB(U_i) = undef.$	NOCB
$CB(U_{i-1}) = undef$ and $CB(U_i) = def.$	ESTABLISH

Table 1: **Definitions of Centering Theory transitions** used in this study.

CT transition types. The coherence experiments we study compare documents in their original orderings to randomly sentence-permuted texts; our analysis shows that the main difference is an increased number of NoCB transitions. This explains why no simple CT-based coherence metric outperforms a baseline that simply considers whether two adjacent sentences mention the same entity. However, some linguistic claims made by CT hold up when treated as patterns observable in large amounts of data rather than single texts: we show that transitions have different preferences for the transitions that follow them, supporting the assumptions of the RETAIN-SHIFT pattern (Brennan et al., 1987), and that cheapness and salience are the most important factors for transition preferences (Strube and Hahn, 1999; Kibble, 1999).

Related work. Previous empirical studies of CT use small corpora of limited domains; for example, Poesio et al. (2000) and Poesio et al. (2004) inspect the effect of various parameter settings on the percentage of utterances that obey the constraints and rules of CT, using about 500 sentences from pharmaceutical leaflets and descriptions of museum objects. While this study sheds light on many aspects of CT, pharmaceutical leaflets exhibit a special structure, and museum object descriptions belong to a limited domain. Karamanis et al. (2009) extend this corpus with news and other texts and report results on about 4500 sentences of natural text. Similar but smaller quantitative studies on various aspects of CT have been conducted by Hurewitz (1998), on about 400 spoken and written transitions, by Di Eugenio (1998) for Italian, by Strube and Hahn (1999) in order to evaluate functional information structure as a ranking function for centers, and more recently by Maat and Sanders (2009) for Dutch and by Taboada (2008) for spoken text.

documents (total)	535
news (479), essay (41), letters (15)	
sentences (total)	14,096
paragraphs (total)	5,605
one-sentence paragraphs	1,405
avg. # of sentences per par.	3.02
all CT transitions	13,561
transitions within paragraphs	8,491

Table 2: **Corpus statistics.**

2 Data and implementation of CT

This section describes the data our corpus study is based on, and the decisions we made when implementing our version of CT.

Data. Our corpus is the portion of the Wall Street Journal for which OntoNotes 4.0 (Hovy et al., 2006) provides manual coreference annotations.[1] For syntactic information, constituent parses from Penn TreeBank 2.0 (Marcus et al., 1993; Vadas and Curran, 2007) are automatically converted to dependency parses using the tool from Johansson and Nugues (2007).

OntoNotes annotates both *identical* coreference as in 'She had a *good suggestion* and *it* was accepted' and *appositive* coreference, as in '*Washington*, *the capital city*'. Additionally, we assume coreference between two nouns if they share a lemma.

We use only documents labeled as *news*, *essay* or *letters* by Webber (2009), omitting the other genres due to low frequency. Table 2 gives a statistical overview of the corpus.

Implementation. Implementing CT requires some parameter-setting; we follow the findings of Poesio et al. (2000), taking sentences as the unit

[1] Coreference information is necessary to appropriately link entities across utterances; the same data set (using OntoNotes 2.9) is used in (Louis and Nenkova, 2010).

of **utterances**, and identify **paragraphs** by empty lines in the source data. We consider nouns and personal and possessive pronouns to **realize entities**. Elements of $CF(U_n)$ are ranked by *grammatical function*, with SUBJ $>$ OBJ $>$ OTHER. After ranking subject and object of the main clause, remaining entity mentions are ranked according to their *surface order*. Nouns modifying other nouns directly follow their heads.

3 Corpus analysis

We investigate several aspects of CT on our corpus and implementation; here we describe these aspects and the results of our analysis.

3.1 Rule 1: pronominalization

Our first finding is strong support for **Rule 1** of CT (Grosz et al., 1995), which expresses the intuition that only the most salient entities of an utterance are pronominalized. According to this rule, if the CB of an utterance is not pronominalized, neither should any other entity in the utterance. The corpus contains 5907 utterances with non-pronominal CBs. 64.7% of these contain no pronouns at all. 4.9% contain expletive pronouns, and 26.4% contain pronouns that have antecedents in the same sentence such as in example (1). We do not regard these cases as violations. Only 4.0% of all utterances with a non-pronominal CB have pronouns with antecedents outside the sentence, violating Rule 1.

(1) More broadly, $[_{CB}$Mr. Boren$]$ hopes that Panama will shock *Washington* out of *its* fear of using military power. (wsj0771)

3.2 Preferences for transition types

It has been proposed that different CT transitions contribute differently to the perceived degree of coherence of a text. In their algorithm for centering and pronoun binding, Brennan et al. (1987) assume a simple ranking of transitions with respect to their assumed degree of coherence: CONTINUE > RETAIN > SMOOTH-SHIFT > ROUGH-SHIFT. Figure 1 shows that these four transitions occur with similar frequencies in our corpus, both within and between paragraphs. Hence, it is not the case that the transitions that are more coherent according to Brennan are in fact used more often by authors, even in perfectly coherent texts.

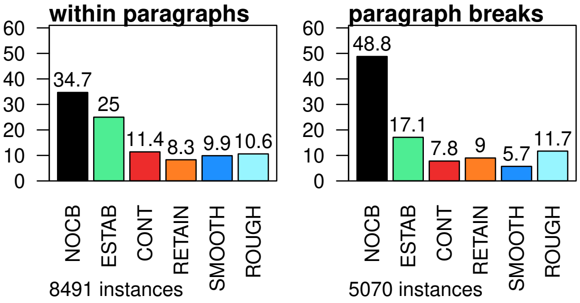


Figure 1: **Distribution of CT transitions** in percent.

The percentage of NOCB transitions is much higher at paragraph boundaries than within paragraphs. However, more than 50% of paragraph-initial sentences mention an entity realized at the end of the previous paragraph, with the salient transitions (see Table 1) being less likely than the non-salient transitions. This indicates that new paragraphs usually change focus when they relate to previous centers. The relatively high percentage of ESTABLISH is due to the high frequency of NOCB, after which only NOCB or ESTABLISH can follow. The *essay+letters* subset of documents has more NOCB transitions than *news* (within paragraphs 43.4% versus 32.6%), indicating that entity coherence matters more in news text, and that *essay+letters* more often reference entities indirectly (not shown in Figure 1).

3.3 Kibble (2001): reformulation of Rule 2

Kibble (2001) suggests that the standard preference ordering of transitions is unmotivated and suggests ranking transition types by considering the interaction of several criteria. Our analysis supports his claims that *cheapness* and *salience* are most important in determining transition preferences, and *cohesion* is of least importance. His proposal draws motivation from natural language generation work (Kibble, 1999), but no corpus study has previously been done. Here we consider only within-paragraph transitions, under the assumption that they do not contain topic changes. Of these transitions, 65.3% have a CB. Of those with a CB: 52.1% have *salient* CBs (i.e., the CB is also the CP of the utterance); 53.9% are *cheap* (the CB of an utterance matches the CP of the *previous* utterance); and 30.2% have the same CB as the previous utterance (*cohesion*).

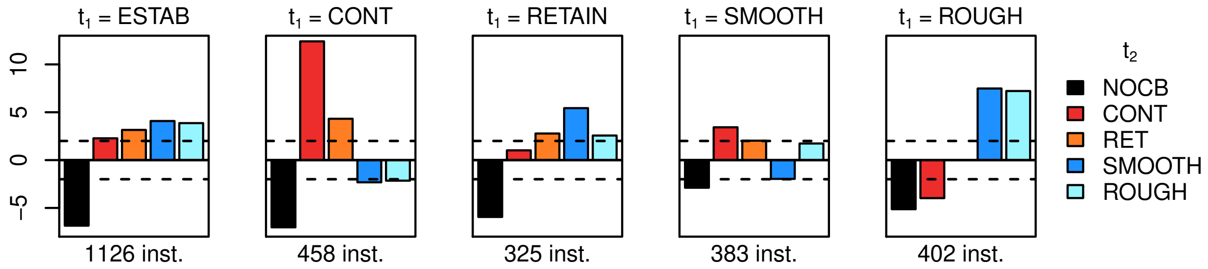


Figure 2: **Residuals of χ^2-tests**: based on 4291 within-paragraph pairs. In 1597 pairs, t_1 is NoCB (not shown).

3.4 Transition bigram distributions

Rule 2 of CT as originally formulated by Grosz et al. (1995) states: *"Sequences of continuation are preferred over sequences of retaining, which are in turn preferred over sequences of shifting."* Thus, in this part of the study, we ask: are there any patterns regarding sequences of transitions?

We first compute $P_{bigram} = P(t_2|t_1)$, the distributions of transitions t_2 conditioned on their previous transition t_1, using the within-paragraph subset. We want to find out whether some transition pairs occur more often than expected. As some transitions are much more frequent than others, it is hard to draw conclusions directly from looking at P_{bigram}. Instead, we apply a statistical test: we compare each P_{bigram} to $P_{unigram}$, the overall distribution of transitions that follow some other transition. We compute Pearson's χ^2-test and plot the residuals in Figure 2. Residuals with absolute value > 2 are considered major contributors to significance, indicated by the dashed lines. We find significant differences between P_{bigram} and $P_{unigram}$ for each t_1 ($p < 0.01$).

We conclude the following: (a) Although NoCB is by far the most frequent transition type overall, it occurs less often than expected after any of the other five transition types. This indicates that there are entity-coherent portions of texts, where multiple utterances share and develop centers. A similar intuition has been proposed, but not tested, in the framework of Rhetorical Structure Theory (Knott et al., 2001). (b) There is a strong tendency that after a Continue transition, there will be another Continue or a Retain. Once a segment strongly focuses on a center, it is likely that the center will be kept. Shifting is less likely after Continue than expected by the overall distribution of transitions. (c) After Retain, there are not more Continues than expected, but many more Smooth-Shifts. This supports the assumption of the Retain-Shift pattern, which may signal introduction of a new discourse topic (Brennan et al., 1987; Strube and Hahn, 1999).

The transition following a Smooth-Shift tends to continue with or retain the new center, and after Rough-Shifts, more shifts and far fewer Continues than expected occur. From this, we can conclude that salience influences the author's choice for the next transition: when the current center is salient (as in Continue and Smooth-Shift), there is a tendency to keep the center. When the current center is not salient (as in Retain and Rough-Shift), there is a tendency to shift to a new center. This observation again supports the principle of *cheapness* (Strube and Hahn, 1999). In fact, all transition pairs with the largest positive residual are classified as cheap by Strube and Hahn (1999), while most other pairs are considered expensive. The only exception is Rough-Rough, which is considered expensive by Strube and Hahn but has a large positive residual. The most frequently occurring bigrams excluding NoCB-bigrams are Estab-Cont (213), Estab-Rough (190), Cont-Cont (181), Estab-Smooth (180) and Rough-Rough (103).

4 Centering-based coherence metrics

Using a small corpus from a limited domain, Karamanis et al. (2009) find that CT-based metrics have no success in improving upon a baseline dubbed M.NoCB, which simply uses whether two sentences share a center or not. In order to shed light on the utility of CT for coherence assessment, we replicate their *information ordering* experiment using our corpus data. The assumption underlying this experimental method is that the original sentence order (OSO) of a text should be scored higher by coherence metrics than any permutation of the text's sentences. We exhaustively enumerate all permutations for texts with fewer than 10 sentences and use a random sample of 1,000,000 permutations for each longer text (only the first 30 sentences of each text are considered in this case). CT transitions are computed for the OSO and for each permutation. We use noun lemma matching as well as gold-standard coreference chains. This oracle style of entity reference resolution has also been applied by Lapata and Barzilay (2005), among others.

We compare the following CT-based metrics described by Karamanis et al. (2009): M.NoCB counts NoCB transitions; M.KP counts NoCBs as well as all violations of cheapness, coherence and salience (following Kibble and Power (2000)); M.BFP prefers the ordering with the most Continues; if equal, the one with most Retains etc.[2] (following Brennan et al. (1987)); and M.Cheap sums up violations of cheapness (following Strube and Hahn (1999)). Karamanis et al. (2009) do not consider NoCBs to be violations of cheapness. As the permutations in general contain more NoCBs than the OSO, they contain fewer violations of cheapness. Using absolute counts of violations of cheapness hence leads to classification error rates worse than chance. We count NoCBs also as violations of cheapness, and hence actually test a combination of continuity and cheapness.

We score the OSO and the permutations with each CT-based metric. In order to evaluate the performance of metric M, the *classification error rate* is computed as $better(M, OSO) + 0.5 * equal(M, OSO)$ where $better(M, OSO)$ is the percentage of permutations scored higher than the OSO, and $equal(M, OSO)$ is the percentage of permutations achieving the same score as the OSO. The lower the classification error rate of M, the better its performance. A rate greater than 50% means that the metric scores the permutation higher than the OSO in the majority of cases.

Metric	Our corpus	Karamanis
M.KP†	0.219∗	0.561
M.NoCB	0.226∗	0.217
M.Cheap†	0.265	0.698
M.BFP	0.285	0.280
documents	535	542
sentences	14,096	4,380

Table 3: **Classification error rates**. ∗ Rates do **not** differ significantly ($p < 0.01$) according to a two-sided binomial test. † Considers NoCB to be a violation of cheapness.

	NoCB	Estab	Cont	Ret	Smooth	Rough
OSO	7.0	3.7	2.3	2.5	2.3	4.0
permutations	10.2	3.4	0.8	1.3	0.7	2.1
difference	*+3.2*	*-0.3*	*-1.5*	*-1.2*	*-1.6*	*-1.9*

Table 4: **Average frequencies** of transition types per document.

Table 3 shows the classification error rates we obtained on our data set, with the results of Karamanis et al. (2009) for comparison. The texts in our data set contain 26 sentences on average (Karamanis et al.: 8 sentences per text on average). Similar to their findings, M.NoCB is among the best-performing metrics, but in contrast to their results, we find that M.KP performs best, though not significantly differently from M.NoCB, and M.BFP performs worst in our experiments. This is in line with the results presented in Section 3.1, and indicates that a feature-based approach to CT-based coherence metrics, using indicators such as coherence, salience and cheapness, works better than the more coarse-grained transition-based approach.

Table 4 shows the average frequencies of the transition types per document both for the original documents and for their permutations. When comparing the numbers for OSOs and permutations, the numbers of the other transition types are all reduced to approximately the same extent. The major difference between OSOs and permutations is that the latter have more NoCBs, which explains the fact that M.NoCB could not be outperformed by the CT-based coherence metrics proposed in the literature to date.

On the 56 documents of *letters+essay*, lower

[2] This metric doesn't make use of Estab.

classification error rates are achieved (0.055 for M.NoCB). This is surprising given that the original documents contain more NoCBs than *news* text. A possible explanation is that these texts change their focus on different entities as they progress, while news texts keep referring to the same set of entities, and hence a larger number of acceptable orderings is possible.

We conclude that CT-based coherence metrics are attractive as they are completely unsupervised and domain-independent, but they seem to reach their upper bound at a classification error rate of around 20% on our corpus. However, other CT-inspired coherence metrics such as the entity-grid model (Lapata and Barzilay, 2005; Barzilay and Lapata, 2008) achieve much better performance by means of a supervised training step.

5 Conclusion, discussion, future work

We have presented the largest study of CT based on natural text to date. While CT adequately describes some linguistic patterns according to our study, these can only be found by analysing collections of texts, not single texts. We show that the different transition types are used in natural text with no clear preference and that genre may play a role in choice of coherence device. We find strong empirical support for CT's claims regarding pronominalization of entity mentions, as well as for the claim that cheapness and salience play a greater role than cohesion.

Our replication of previous information ordering experiments indicates that it is not possible to leverage CT transitions to design unsupervised domain-independent metrics measuring the coherence of normal-length texts due to sparsity.[3] No metric significantly outperforms a baseline that uses only the number of NoCB transitions.

Miltsakaki and Kukich (2000) find ROUGH-SHIFTS to be a predictor of incoherence for student essays, but these are a domain very different from our corpus of financial news written by professional journalists. We suggest that if it is clear to the reader which entity is referred to in an utterance, it may even be easy to process a large number of shifts, as example (2) shows.

(2) (a) Two dozen scientists reported results with variations of the *experiments* [...] by Fleischmann and Pons.

(b) The $[_{CB}$*experiments*$]$ involve plunging the two *electrodes* into ”heavy” water. (ESTABLISHMENT)

(c) When an electric current is applied to the $[_{CB}$*electrodes*$]$, the heavy *water* did begin to break up, or dissociate. (ROUGH-SHIFT)

(d) Ordinarily the breakup of the $[_{CB}$*water*$]$ would consume almost all of the electrical energy. (ROUGH-SHIFT)
(wsj1550, shortened)

This kind of discourse organization, in which an element introduced in an utterance (*rheme*) is used as the *theme* (known information) in the next utterance, has been described as *simple linear textual progression* (Danes, 1974) or *focus-topic chaining* (Smith, 2003). We argue that shifting centers may be what makes a text interesting to readers.

CT focuses on entity-based coherence. However, in many perfectly coherent text passages no direct coreference links are found. Consider example (3):

(3) (a) Competition has glutted the market with both skins and coats, driving prices down.

(b) The animal-rights movement hasn't helped sales. (NoCB)

(c) Warm winters over the past two years have trimmed demand, too, furriers complain. (NoCB) (wsj1586)

Some utterance pairs are instead connected via reference to the same situations or events, which is one direction for future research; Christensen et al. (2013) and Hou et al. (2013) propose promising approaches to identifying mentions referring to the same situtation or event. Other interesting directions include investigating relationships between entity coherence and other coherence devices such as discourse relations (Louis and Nenkova, 2010); and combining CT-based features with, e.g., features reflecting semantic content or licensing particular syntactic realizations. Finally, further analysis of CT on a greater variety of genres is warranted.

Acknowledgments

We thank the anonymous reviewers, Manfred Pinkal, Michaela Regneri and Caroline Sporleder for helpful

[3] Initial experiments trying to leverage the bigram patterns found in Section 3.4 were not successful as bigram distributions suffer even more from sparsity than unigrams.

comments. This research was supported in part by the MMCI Cluster of Excellence, and the first author is supported by an IBM PhD Fellowship.

References

Regina Barzilay and Mirella Lapata. 2008. Modeling local coherence: An entity-based approach. *Computational Linguistics*, 34(1):1–34.

Susan E Brennan, Marilyn W Friedman, and Carl J Pollard. 1987. A centering approach to pronouns. In *Proceedings of the 25th annual meeting on Association for Computational Linguistics*, pages 155–162. Association for Computational Linguistics.

Janara Christensen, Stephen Soderland Mausam, and Oren Etzioni. 2013. Towards coherent multi-document summarization. In *Proceedings of NAACL-HLT*, pages 1163–1173.

Frantisek Danes. 1974. Functional sentence perspective and the organization of the text. *Papers on functional sentence perspective*, 113.

Barbara Di Eugenio. 1998. Centering in Italian. In Marilyn A. Walker, Aravind K. Joshi, and Ellen F. Prince, editors, *Centering Theory in Discourse*, chapter 7, pages 115–137. Clarendon Press, Oxford.

Micha Elsner and Eugene Charniak. 2011. Extending the Entity Grid with Entity-Specific Features. In *HLT*, pages 125–129.

Barbara J Grosz, Aravind Joshi, and Scott Weinstein. 1995. Centering : A Framework for Modeling the Local Coherence of Discourse. *Computational Linguistics*.

Yufang Hou, Katja Markert, and Michael Strube. 2013. Global inference for bridging anaphora resolution. In *HLT-NAACL*, pages 907–917.

Eduard Hovy, Mitchell Marcus, Martha Palmer, Lance Ramshaw, and Ralph Weischedel. 2006. OntoNotes: the 90% solution. In *Proceedings of the Human Language Technology Conference of the NAACL, Companion Volume: Short Papers*, NAACL-Short '06, pages 57–60, Stroudsburg, PA, USA. Association for Computational Linguistics.

Felicia Hurewitz. 1998. A quantitive look at discourse coherence. In Marilyn A. Walker, Aravind K. Joshi, and Ellen F. Prince, editors, *Centering Theory in Discourse*, chapter 14, pages 273–292. Clarendon Press, Oxford.

Richard Johansson and Pierre Nugues. 2007. Extended constituent-to-dependency conversion for english. In *Proc. of the 16th Nordic Conference on Computational Linguistics (NODALIDA)*, pages 105–112.

Megumi Kameyama. 1986. A property-sharing constraint in centering. In *Proceedings of the 24th annual meeting on Association for Computational Linguistics*, pages 200–206. Association for Computational Linguistics.

Nikiforos Karamanis, Chris Mellish, Massimo Poesio, and Jon Oberlander. 2009. Evaluating centering for information ordering using corpora. *Computational Linguistics*.

Rodger Kibble and Richard Power. 2000. An integrated framework for text planning and pronominalisation. In *Proceedings of the first international conference on Natural language generation-Volume 14*, pages 77–84. Association for Computational Linguistics.

Rodger Kibble. 1999. Cb or not Cb? Centering Theory applied to NLP. In *Proceedings of the ACL workshop on Discourse and Reference Structure*.

Rodger Kibble. 2001. A reformulation of Rule 2 of centering theory. *Computational Linguistics*, 27(4):579–587, December.

Alistair Knott, Jon Oberlander, Mick ODonnell, and Chris Mellish. 2001. Beyond elaboration: The interaction of relations and focus in coherent text. *Text representation: linguistic and psycholinguistic aspects*, pages 181–196.

M Lapata and R Barzilay. 2005. Automatic evaluation of text coherence: Models and representations. *International Joint Conference On Artificial* ....

Annie Louis and Ani Nenkova. 2010. Creating local coherence: An empirical assessment. In *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 313–316. Association for Computational Linguistics.

Henk Pander Maat and Ted Sanders. 2009. How grammatical and discourse factors may predict the forward prominence of referents: two corpus studies. *Linguistics*, 47(6):1273–1319.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. 1993. Building a large annotated corpus of English: the Penn TreeBank. *Comput. Linguist.*, 19(2):313–330, June.

Eleni Miltsakaki and Karen Kukich. 2000. The role of centering theory's rough-shift in the teaching and evaluation of writing skills. In *Proceedings of the 38th Annual Meeting on Association for Computational Linguistics*, pages 408–415, Stroudsburg, PA, USA.

Massimo Poesio, Hua Cheng, Renate Henschel, Janet Hitzemann, Rodger Kibble, and Rosemary Stevenson. 2000. Specifying the parameters of Centering Theory: a corpus-based evaluation using text from application-oriented domains. In *Proceedings of the 38th ACL*.

Massimo Poesio, Barbara Di Eugenio, Rosemary Stevenson, and Janet Hitzeman. 2004. Centering: A Parametric Theory and Its Instantiations. *Computational Linguistics*, 30 (3)(2000).

Carlota S Smith. 2003. *Modes of discourse: The local structure of texts*, volume 103. Cambridge University Press.

Michael Strube and Udo Hahn. 1999. Functional centering: Grounding referential coherence in information structure. *Computational Linguistics*, 25(3):309–344.

Maite Taboada. 2008. Deciding on units of analysis within Centering Theory. *Corpus Linguistics and Linguistic Theory*, 4(1):63–108.

David Vadas and James Curran. 2007. Adding noun phrase structure to the penn treebank. In *ANNUAL MEETING-ASSOCIATION FOR COMPUTATIONAL LINGUISTICS*, volume 45, page 240.

Bonnie Webber. 2009. Genre distinctions for discourse in the Penn TreeBank. In *Proceedings of the Joint Conference of the 47th Annual Meeting of the ACL and the 4th International Joint Conference on Natural Language Processing of the AFNLP: Volume 2-Volume 2*, pages 674–682. Association for Computational Linguistics.

Automatic Genre Classification in Web Pages Applied to Web Comments

Melanie Neunerdt, Michael Reyer, Rudolf Mathar
Institute for Theoretical Information Technology
RWTH Aachen University, Germany
{neunerdt, reyer, mathar}@ti.rwth-aachen.de

Abstract

Automatic Web comment detection could significantly facilitate information retrieval systems, e.g., a focused Web crawler. In this paper, we propose a text genre classifier for Web text segments as intermediate step for Web comment detection in Web pages. Different feature types and classifiers are analyzed for this purpose. We compare the two-level approach to state-of-the-art techniques operating on the whole Web page text and show that accuracy can be improved significantly. Finally, we illustrate the applicability for information retrieval systems by evaluating our approach on Web pages achieved by a Web crawler. [1]

1 Introduction

The high amount of social media tools lead to constantly growing user generated content in the Web. Different types of Web sites, e.g., blogs, forums as well as news sites provide functionalities to post Web comments to related articles. Whereas an abundance of Web comments is publicly available, the size of the World Wide Web makes it a challenging task to identify Web pages with Web comments. One solution to build topic specific Web comment corpora, is a focused crawler jointly using a topic classifier and a Web comment classifier. Altogether, automatic Web comment detection has the potential to significantly improve the performance of information retrieval systems and provide corpora, which can efficiently be used, e.g., for marketing studies.

In this paper, we use *Web comment* as a particular text genre. It is fundamental to consider the fact, that a Web page can be a composition of different text genres. For example Web comments posted to a particular article on the same Web page obviously comprise the text genre *article* and *Web comment*. It is to be expected that the associated feature vectors of different text genres show different characteristics which can be identified more easily if they are investigated separately. Therefore, we apply a two-level classification approach. We first classify the text genre of each text segment of the Web page. On the second level we declare a Web page to be relevant, if a Web comment is detected in at least one text segment.

The outline of this paper is as follows. After reviewing related work in Section 2, we propose the two-level classification approach and discuss potential features for detecting Web comments in Web pages in Section 3. Section 4 introduces the corpora used and Section 5 reports experimental results. Finally, we conclude our work and discuss future research directions.

2 Related Work

State-of-the art Web text genre classification approaches differ mainly in the feature set they use and the genre classes they define. Various types of features have been proposed for automatic text genre classification. Web pages are additionally equipped with information such as, formatting information from HTML tags or css-style classes, and meta information given in the URL of a Web page, which further increases the possible feature dimension. Classes for Web genre classification are often related to the type of Web page. They lead from very few classes, e.g., seven in (Lee and Myaeng, 2002), to fine-grained classification with fifteen genres, (Lim et al., 2005). In (Meyer zu Eissen and Stein, 2004) a user study on Web genre class usefulness is performed. As a matter of fact, selected genres influence feature selection for automatic classification.

(Meyer zu Eissen and Stein, 2004; Lim et al., 2005; Qi and Davison, 2009) propose to use style-related HTML features, e.g., HTML tag frequencies, token-related features, e.g., text statistics or digit frequencies and POS features, e.g., noun or verb frequencies. In (Lim et al., 2005) some URL-related features, e.g., depth of URL, are proposed as additional features. (Qi and Davison, 2009) particularly investigate Web-specific features and their usability for different Web page classification tasks, e.g., sentiment classification and subject classification. Beside, on-page features, directly located on the page to be classified, they investigate the usability of features of linked pages. The performance of different POS-related features is particularly studied in (Feldman et al., 2009; Santini, 2004). (Feldman et al., 2009) propose to use POS histograms over a sliding window as features. Compared to the results achieved by a classifier working with POS trigram features a significant performance increase is reported. All these approaches achieve classification accuracies between 70% and 80%.

3 Classification Approach

The goal of our work is to detect Web comments in Web pages. The overall approach is simple, Web pages are split into small text segments, which are represented by feature vectors and thereby classified. As we aim at showing that the classification can significantly be improved by the segmentation approach, we apply a very simple rule based segmentation based on HTML tags: All tags, but links <*a*>, line breaks <*br*> and some font tags, e.g., <*small*> and <*strong*>, are used for splitting. The result is a very fine-grained segmentation, which might result in splitted Web comments or articles. We believe that splitting in too many segments will not effect the performance too much. Though, the parametrization of the classification may be effected, e.g., k.

In the following we mathematically describe the two-level classification problem, classifying Web text segments on first level and Web pages on second level. We define the index set $\mathcal{PM} = \{1, \dots, P\}$ for Web pages. Each Web page p is splitted into a sequence of N_p text segments,

$$\mathbf{S}^p = \left(\mathbf{s}_1^p, \dots, \mathbf{s}_{N_p}^p\right),$$

where each text segment $\mathbf{s}_i^p$, $i = 1, \dots, N_p$ is represented by an n-dimensional feature vector

$$(\mathbf{s}_i^p)^T \in \mathcal{X} = \mathcal{S}_1 \times \mathcal{S}_2 \times \dots \mathcal{S}_n.$$

The aim of the segment classification is to predict the to $\mathbf{S}^p$ associated genre class vector

$$\mathbf{c}^p = \left(c_1^p, \dots, c_{Np}^p\right),$$

with $c_i^p \in \mathcal{C}$ for $i = 1, \dots, N_p$ and $\mathcal{C}$ comprises the set of genre classes.

First of all, we consider a seven-class problem. We differentiate between the seven classes, $WebCOMment$, $ARTicle$, $USEr$, $TITle$, $TIMe$, $METa$ and $OTHer$, represented by

$$\mathcal{C} = \{COM, ART, USE, TIT, TIM, MET, OTH\}\,. \quad (1)$$

Applying this approach, related information like the posting date, the user name or the related article are automatically identified. The Web comment corpus quality benefits from such information. However, for Web comment detection the binary decision if the class is $COMment$ or not is sufficient. In order to solve the sequence labeling task, the optimization problem

$$\hat{\mathbf{c}}^p = \arg\max_{\mathbf{c}^p} \left\{g\left(\mathbf{S}^p, \mathbf{c}^p\right)\right\}$$

where g represents any decision function, is solved. This is a huge optimization problem, which is simplified by two assumptions. First, the genre classes $\hat{c}_i^p$ are predicted independently from predictions $\hat{c}_j^p$ for $j \neq i$ by

$$\hat{c}_i^p = \arg\max_{c_i^p} \left\{g\left(\mathbf{S}^p, c_i^p\right)\right\}.$$

Second, we assume that the text genre class for a given text segment $\mathbf{s}_i^p$ at position i only depends on some - here k - preceeding and succeeding text segments. Hence, the optimization problem is reformulated as

$$\hat{c}_i^p = \arg\max_{c_i^p} \left\{g\left(\mathbf{s}_{i-k}^p, \dots, \mathbf{s}_i^p, \dots, \mathbf{s}_{i+k}^p, c_i^p\right)\right\}.$$

Note that, for the first and last segments of each Web page p, there are not enough predecessor and successor segments available. In such cases the number of considered predecessor and successor segments is reduced to the maximal possible amount. Web pages are considered to be *RELevant*, if at least one segment is classified as $COMment$. Hence, the condition for the second level classification is

$$\sum_{i=1}^{Np} 1_{\{COM\}}\left(\hat{c}_i^p\right) \geq 1, \text{ with } 1_{\mathcal{A}}\left(x\right) = \begin{cases} 1, & x \in \mathcal{A} \\ 0, & \text{else} \end{cases}$$

representing the indicator function for any set $\mathcal{A}$.

In order to compare our approach to existing approaches, we directly solve the classification on Web page level. We represent the whole Web page text by the same feature vector than for later Web page text segments. Hence, we solve the optimization problem for $N_p = 1$.

3.1 Web Page Feature Types

We generally expect that the combination of several features from each level can be used to identify text segments as elements of $\mathcal{C}$, which can then be used to identify relevant Web pages. In our approach we combine some of the features proposed in (Lim et al., 2005; Meyer zu Eissen and Stein, 2004; Kohlschütter et al., 2010) with some new features, which results in 102 features in total. New features are introduced for all three feature types. Such features are motivated by an extensive study of the language in Web comments and the structure of Web pages like blogs and forums.

3.1.1 Token-based Features

Token-based features are easily accessible, without any text preprocessing. However, in order to develop a topic independent solution, token-level features need to be carefully selected. We extend simple frequency count features, e.g., punctuation marks, digits or symbols, by Web comment related features. Emoticons, letter iterations, e.g., Hallllloooo (Helllloooo), multiple punctuations, e.g., !!!, ?!, @ symbols, uncapitalized words, etc. are taken as additional features counting the frequency. Furthermore, some sentiment related features are defined. Adjectives in Web comments are inherently connected with evaluative judgements. Hence, frequency counts of positive and negative orientated adjectives are promising features for differentiating Web comments from other texts. The *SentiWS* word list proposed in (Remus et al., 2010) is used for such frequency counts.

Finally, we complement some features proposed in previous works. (Kohlschütter et al., 2010) propose a text density measure particularly for Web text segment classification. From, e.g., (Lim et al., 2005) we take over frequency counts of *content, function* and *unusual* words. In total 50 token-based features are used for the classification.

3.1.2 POS-based Features

Many approaches introduce features based on Part-of-Speech (POS) information for text classification. Basically, such features are simple frequency counts of single POS tags (1-gram) or ratios between different POS tags, e.g., the verb-noun ratio. For our approach we combine POS-based features proposed in (Lim et al., 2005; Meyer zu Eissen and Stein, 2004) using the STTS tagset (Schiller et al., 1999) with 54 part-of-speech tags. As a tagger we use WebTagger (Neunerdt et al., 2014) particularly developed for social media texts. Previous studies have shown, that due to the dialogic style of Web comments, particularly the sequence of POS tags are different. E.g., in (Neunerdt et al., 2013) POS trigram (3-gram) statistics evaluated on a social media text corpus show significant differences compared to newspaper texts. Hence, POS 3-grams seem to be a good feature to differentiate Web comments from other texts. However, to determine reliable POS tag features requires automatic POS tagging with high accuracies. Common state-of-the art taggers achieve high accuracies on newspaper texts, which significantly drops when applied to unstandardized texts, such as Web comments. For the main classification approach we use 38 POS-based features. Note that, POS-based features, such as sentence length statistics, are also included here, since we use POS tags to detect the end of a sentence.

3.1.3 HTML-based Features

Structural features, based on HTML tags (headline <*h1*>, paragraph <*p*>, etc.) or CSS classes are commonly used for Web page classification approaches. Unfortunately, the usage of CSS classes and HTML tags are mainly Web site specific. The increasing usage of CSS classes and styles makes it even more difficult to infer semantic relations between HTML tags and text segments. However, CSS class names are not chosen arbitrarily and often have a semantic relation to the text elements they are defined for, e.g. the user, the date or the web comment. This allows us to define useful features based on such CSS class names. For example the style of a Web comment is frequently defined by CSS class names like, e.g., *comment, post, message*. Based on a list of common class name strings, we define binary features marking, if one of the strings is contained in the CSS class name of the current segment. In addition, we define another binary feature, marking the presence of HTML tags, which never jointly occur with Web comments due to their functionality, e.g., *h1 option, title, em, button*. Finally, we use further structure related features. Since, the position of the Web segment is often a good hint for the corresponding class, we introduce that as additional feature. For example, Web comments are often located below the article at the end of a Web page. Some other features, e.g., the link density, are taken from (Meyer zu Eissen and Stein, 2004). In total, 14 HTML-based features are used for classification.

4 Corpora

Evaluations are performed on two different corpora, a manually collected Web page corpus for training and testing, and a collection of Web pages accessed with a crawler for validation. Both corpora are selections of German Web pages solely.

4.1 Web Comment Collection

The Web comment collection is created particularly to train a Web comment classifier. It consist of 336 manually assessed Web pages from 237 different Web sites/domains. 71% of the Web pages contain at least one posted Web comment. The remaining 99 Web pages contain Web comments related articles. The Web pages contain forums, blogs and different news sites dealing with different topics. In this paper we call that corpus *Web Comment Train* (*WCTrain*). First we apply the segmentation described in Section 3 to each Web page. Considering the visual representation in a Web browser, plain Web text segments are labeled by four human annotators as either $WebCOMment$, $ARTicle$, $USEr$, $TITle$, $TIMe$ or $METa$ (text, which gives any further meta information to another text/author). Note that, every page is labeled by one annotator, since we do not expect significant inter-annotator disagreement in this context. Unselected text is regarded to $OTHer$ (no content, left over class). The distribution of all classes at token (including non-words)-, word- and segment-level is depicted in Table 1.

Class	# Segments	# Words	# Tokens
Total	45,955	479,483	596,630
$WebCOMment$	5.36 %	36.78%	35.10%
$TITle$	2.94%	1.53%	1.52%
$TIMe$	4.79%	0.62%	1.01%
$METa$	2.16%	0.71%	1.59%
$USEr$	4.43%	0.83%	0.82%
$ARTicle$	1.59%	25.34%	23.89 %
$OTHer$	78.74%	34.19%	36.07%
$NonCOMment$	94.64%	63.22%	64.90%

Table 1: Class distribution in the *WCTrain* Corpus.

4.2 Crawl Collection

A second corpus is introduced, with the goal to evaluate the applicability of the developed Web comment classifiers for information retrieval systems. The Web page corpus is acquired by starting a crawl process from 112 seed pages. We manually have selected the seed pages from 78 different domains, fulfilling one of the two criteria: The Web page is a blog, forum or news site, which contains at least one Web comment. The Web page is a so called hub page, which contains a high number of links to Web pages, which also fulfill the first requirement. The crawl process results in 72,534 Web pages from 1414 different Web domains. For the sample corpus *Web Page Crawl* (*WPCrawl*) 827 Web pages are selected randomly from the basic crawl result. In contrast to the *WCTrain* corpus the annotation is performed on Web page level rather then Web segment level. Two human annotators label each Web page as Web pages are labeled by four human annotators as *RELevant*, if it contains at least one Web comment. All remaining Web pages are regarded to be *non-RELevant*. In total, 57% of such Web pages are *RELevant*. This corpus serves as validation for classification on the second level.

5 Experimental Results

In this section we analyze different classifiers utilizing different feature combinations. In order to build a good information retrieval system, it is important to achieve high precision rates for the particular $COMment$ class. High precision means, high quality Web comment corpora. Therefore, we particularly study the classifiers, considering precision rates P_{COM} on segment level for the COM class. For our experiments we use the WEKA software, (Hall et al., 2009). We analyze three different classifiers, a KNN classifier, a decision tree (J48) and a Support Vector Machine (SVM). For the KNN classifier we used the weighted Manhattan distance as a metric considering $K = 9$ next neighbors, which gave the best result for $K = 1 \dots 15$. Varying the decision tree threshold of the minimum number of objects in a leaf from 2 to 15 and choosing between binary and non-binary split, we used a non-binary tree with a threshold of 6, which gave the best result. For the SVM classifier a Pearson VII function-based universal kernel achieves best results.

5.1 Validation on *WCTrain* Corpus

Cross validation results for the three classifiers using an n-dimensional feature vector are depicted in Table 2. We measure classification accuracy by COM class precision P_{COM}, COM class recall R_{COM}, average F_1-Score, average ROC Area under Curve (AuC) and total accuracy (AC). The upper part of Table 2 depicts classification accuracies achieved with the three different classifiers using all proposed features. In order to analyze the influence of integrating features from predecessor and successor segments for classification in more detail, classification accuracies for different values of parameter k are depicted in addition.

Comparison of the classifiers for $k = 0$ shows not much difference in total accuracies (AC). However, considering P_{COM} class precision results, the KNN classifier significantly outperforms the other approaches. Highest precision rates for the 2-class and 7-class are achieved for $k = 2$ for all classifiers. This confirms our assumption that similar small sequences of text segments occur in Web pages. Hence, considering the text segments close by are useful features for text genre prediction. The KNN classifier solving a 2-class problem achieves the highest precision rate with 0.94. Hence, beside using different classifiers, the values of k allow for further adjustments towards P_{COM} precision rates without decrease of total accuracies.

Considering that KNN achieves the best results for $k = 0$, we exemplarily investigate the performance of KNN classifier, for each feature type separately. Results are depicted in the middle part of Table 2. Using an approach based on token-based features outperforms the POS-based and HTML-based approach. However, classification accuracy drops significantly, compared to the approach, when using all feature types in combination (KNN (k=0)). We further investigate different feature types by calculating the per-feature information gain. Figure 1 shows the features in decreasing order of their information gain for the 2-class and 7-class problem. Information gain values are below 0.17 for the 2-class and below 0.35 for the 7-class problem. Generally, simple token-based features, e.g., capitalized token counts or letter counts, and POS-based features, e.g., POS tags counts or verb noun-ratio, appear to be strong indicators for class membership for the 2- and 7-class problem. Confirming the results achieved with the classifier using HTML-based features solely, HTML-based features are lower ranked. However, assessing features usability by the information gain rates them independently. In order to investigate the combination of different feature considering their redundancy, we apply a greedy correlation-based feature subset selection proposed by (Hall, 1998). Subsets of features that are highly correlated with the class while having low intercorrelation are preferred. Results are depicted in the lower part of Table 2. P_{COM} rates are slightly lower, compared to the classifiers using all 102 features, however the number of features can significantly be reduced by 4/5, which reduces computational classification effort. Analyzing the resulting feature subsets, e.g., for the 7-class problem results in a combination of 36% token-based, 41% POS-based and 23% HTML-based features. We conclude that, the selected subsets of different feature types as well as relatively low per-feature information gain but at the same time high acceptable classification accuracy shows that particularly combining features from

Algorithm	n	$\mathrm{P_{COM}}$		$\mathrm{R_{COM}}$		Average $\mathrm{F_1}$-Score		Average AuC		AC	
		2-class	7-class	2-class	7-class	2-class	7-class	2-class	7-class	2-class	7-class
KNN (k=0)	102	0.88	0.87	0.70	0.73	0.88	0.76	**0.99**	0.97	0.978	0.915
KNN (k=1)	306	0.93	0.91	0.79	0.82	0.92	**0.81**	**0.99**	**0.98**	0.985	0.934
KNN (k=2)	510	**0.94**	**0.93**	0.85	**0.86**	**0.94**	**0.81**	**0.99**	**0.98**	**0.988**	0.938
SVM (k=0)	102	0.87	0.84	0.75	0.80	0.90	0.74	0.87	0.92	0.980	0.911
SVM (k=1)	306	0.88	0.89	0.86	0.80	0.93	0.77	0.93	0.92	0.986	0.926
SVM (k=2)	510	0.90	0.90	**0.88**	0.78	**0.94**	0.69	0.94	0.90	**0.988**	0.908
J48 Tree (k=0)	102	0.80	0.78	0.73	0.73	0.88	0.75	0.93	0.91	0.976	0.912
J48 Tree (k=1)	306	0.82	0.79	0.75	0.75	0.89	0.76	0.93	0.91	0.978	0.935
J48 Tree (k=2)	510	0.83	0.79	0.76	0.78	0.89	0.76	0.93	0.89	0.979	**0.958**
KNN, token features	50	0.83	0.80	0.63	0.67	0.85	0.63	0.98	0.93	0.973	0.878
KNN, POS features	38	0.80	0.78	0.62	0.65	0.85	0.61	0.96	0.92	0.971	0.865
KNN, HTML features	14	0.71	0.64	0.48	0.55	0.78	0.59	0.94	0.90	0.962	0.855
KNN, Subset features	23/22	0.87	0.82	0.62	0.69	0.86	0.60	0.95	0.91	0.975	0.872
SVM, Subset features	23/22	0.80	0.76	0.39	0.70	0.75	0.50	0.69	0.84	0.962	0.863
J48, Subset features	23/22	0.78	0.74	0.67	0.71	0.86	0.62	0.94	0.90	0.973	0.878

Table 2: Cross Validation Results achieved for different classification approaches 2-class/7-class.

2-class problem									
	KNN Classifier			**Decision Tree**			**SVM Classifier**		
k	P_{REL}	R_{REL}	AC_{PAGE}	P_{REL}	R_{REL}	AC_{PAGE}	P_{REL}	R_{REL}	AC_{PAGE}
0	0.83	0.83	0.80	0.68	**0.98**	0.72	0.84	0.94	**0.86**
1	0.83	0.86	0.82	0.75	0.96	0.79	0.73	0.95	0.77
2	**0.87**	0.85	0.84	0.68	0.92	0.71	0.83	0.94	**0.86**
Classification results achieved on total Web page with $N_p = 1$									
	0.78	0.85	0.78	0.73	0.78	0.71	0.75	0.88	0.76

Table 3: Validation on *WPCrawl* corpus.

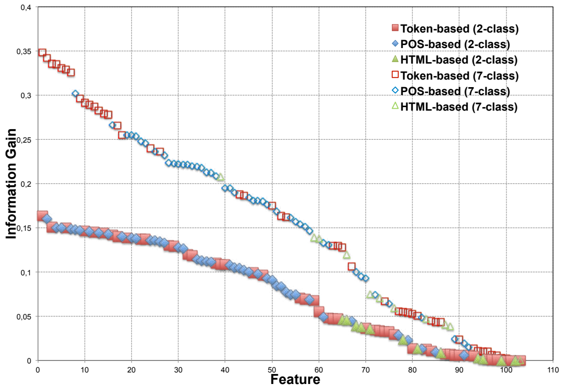


Figure 1: Information gain of different feature types applied to 2- and 7-class problem.

different types is particularly important.

5.2 Validation on *WPCrawl Corpus*

In order to show the usability of our classification approach for information retrieval tasks, we apply our classifier on the *WPCrawl Corpus*. Results are depicted in Table 3. Precision P_{REL}, recall R_{REL} and total accuracy AC_{PAGE} are given on the second classification level for a total Web page rather than a text segment. Hence, e.g., P_{REL} is the number of $RELevant$ Web pages classified as $RELevant$ divided by the total number of $RELevant$ pages. Considering the task of building a Web comment corpus by selecting all $RELevant$ classified pages, high P_{REL} are particularly important. However, the resulting corpus size is even important and hence R_{REL} is not neglectable. Best P_{REL} results are achieved with the KNN classifier with $k = 2$. In this case 463 Web pages would be selected from the original *WPCrawl* corpus, where 87% would be $RELevant$ pages. For comparison, the last column shows results achieved with the classifiers performed without segmentation, on the whole Web page. The highest P_{REL} of 0.78 is achieved with the KNN classifier, which is significantly lower compared to our two-level classification.

6 Conclusion

In this paper, we presented a simple approach for Web comment detection classifying Web text segments as intermediate step. The two level classification particularly improves precision rates compared to a classifier applied to the whole Web page text. Applying our classifier combining token, POS and HTML-based for Web comment corpus refinement to Web pages accessed by a crawler, shows significant improvement in corpus quality. The amount of relevant Web pages containing Web comments, could be improved from 57% to 87% using a KNN classifier.

The presented results raise research in many different directions. Results achieved by feature extensions, motivate to use a Markov model classifier to label Web text sequences. That would allow to model dependencies of predecessor classification results and could further improve classification accuracies. Furthermore, we need to investigate possibilities for feature selection in more detail, to reduce the complexity of the classifier.

References

Sergey Feldman, Marius A. Marin, Mari Ostendorf, and Maya R. Gupta. 2009. Part-of-speech Histograms for Genre Classification of Text. In *ICASSP*, pages 4781–4784. IEEE.

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, and Ian H. Witten. 2009. The weka data mining software: An update. *SIGKDD Explor. Newsl.*, 11(1):10–18, November.

M. A. Hall. 1998. *Correlation-based Feature Subset Selection for Machine Learning*. Ph.D. thesis, University of Waikato, Hamilton, New Zealand.

Christian Kohlschütter, Peter Fankhauser, and Wolfgang Nejdl. 2010. Boilerplate Detection Using Shallow Text Features. In *Proceedings of the Third ACM International Conference on Web Search and Data Mining*, pages 441–450, New York, NY, USA. ACM.

Yong-Bae Lee and Sung Hyon Myaeng. 2002. Text Genre Classification with Genre-revealing and Subject-revealing Features. In *Proceedings of the 25th Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 145–150, New York, NY, USA. ACM.

Chul Su Lim, Kong Joo Lee, and Gil Chang Kim. 2005. Multiple Sets of Features for Automatic Genre Classification of Web Documents. *Inf. Process. Manage.*, 41(5):1263–1276.

Sven Meyer zu Eissen and Benno Stein. 2004. Genre Classification of Web Pages: User Study and Feasibility Analysis. In *KI 2004: Advances in Artificial Intelligence*, pages 256–269. Springer Berlin Heidelberg.

Melanie Neunerdt, Michael Reyer, and Rudolf Mathar. 2013. Part-of-Speech Tagging for Social Media Texts. In *Proceedings of The International Conference of the German Society for Computational Linguistics and Language Technology*.

Melanie Neunerdt, Michael Reyer, and Rudolf Mathar. 2014. Efficient Training Data Enrichment and Unknown Token Handling for POS Tagging of Nonstandardized Texts. In *Proceedings of KONVENS 2014*. In Press.

Xiaoguang Qi and Brian D. Davison. 2009. Web Page Classification: Features and Algorithms. *ACM Comput. Surv.*, 41(2):12:1–12:31.

Robert Remus, Uwe Quasthoff, and Gerhard Heyer. 2010. Sentiws – a publicly available german-language resource for sentiment analysis. In *Proceedings of the 7th International Language Resources and Evaluation (LREC'10)*, pages 1168–1171.

Marina Santini. 2004. A shallow approach to syntactic feature extraction for genre classification, cluk 7: The uk special-interest group for computational linguistics. In *Proceedings of the 7th Annual Colloquium for the UK Special Interest Group for Computational Linguistics*.

Anne Schiller, Simone Teufel, Christine Stöckert, and Christine Thielen. 1999. Guidelines für das Tagging deutscher Textcorpora mit STTS. University of Stuttgart.

TEANLIS - Text Analysis for Literary Scholars

Andreas Müller[1,3]***, Markus John***[2,4]***, Jonas Kuhn***[1,3]
(1) Institut für Maschinelle Sprachverarbeitung Universität Stuttgart
(2) Institut für Visualisierung und Interaktive Systeme (VIS) Universität Stuttgart
(3) `Vorname.Nachname@ims.uni-stuttgart.de`
(4) `Vorname.Nachname@vis.uni-stuttgart.de`

Abstract

Researchers in the (digital) humanities have identified a large potential in the use of automatic text analysis capabilities in their text studies, scaling up the amount of material that can be explored and allowing for new types of questions. To support the scholars' specific research questions, it is important to embed the text analysis capabilities in an appropriate navigation and visualization framework. However, study-specific tailoring may make it hard to migrate analytical components across projects and compare results. To overcome this issue, we present in this paper the first version of TEANLIS (text analysis for literary scholars), a flexible framework designed to include text analysis capabilities for literary scholars.

1 Introduction

Researchers in the (digital) humanities have identified a large potential in the use of automatic text analysis capabilities in their text studies, scaling up the amount of material that can be explored and allowing for new types of questions. To effectively support the humanities scholars' work in a particular project context, it is not unusual to rely on specially tailored tools for feature analysis and visualizations, supporting the specific needs in the project. Support for linking up detailed technical aspects to higher-level research questions is crucial for the success of digital humanities projects, but overly study-specific tailoring limits the usability of the analysis capabilities of the tools in other projects. This effect is also in part due to the potential difficulty of separating the analysis capabilities and the visualizations used to present the results of an analysis[1].

We present the experimental text analysis framework TEANLIS (Text analysis for literary scholars), which is designed to: 1) provide text analysis capabilities which can be applied out-of-the-box and which take advantage of a hierarchical representation of text, 2) integrate functions to load documents which were processed with other text analysis frameworks (e.g. GATE (Cunningham et al., 2013)) and 3) provide standard analysis functions for documents from the recent infrastructure initiatives for the digital humanities such as CLARIN,[2] DARIAH[3] and TextGrid (Hedges et al., 2013). TEANLIS is not in and of itself meant to be a tool for literary analysis, but rather a framework which allows developers to quickly build a tool for literary analysis in particular and text analysis in general.

We work with data visualization experts involved in our project to ensure that TEANLIS can support interactive visualizations of results. Interactive visualizations enable intuitive access to the results of the computational linguistics (CL) methods we provide. TEANLIS is different from established frameworks like GATE (Cunningham et al., 2013) or UIMA (Ferrucci and Lally, 2004) in that it focuses on supporting visualizations and analysis capabilities based on a hierarchical doc-

[2] `http://www.clarin-d.de`

[3] `https://de.dariah.eu/`

ument structure and special analysis capabilities tailored to the needs of literary scholars. A first version of the framework is available online[4]. An example for a tool developed on the basis of TEANLIS is described in Koch et al. (2014).

The remainder of this paper is structured as follows: In section 2, we will review existing frameworks for providing literary scholars with text analysis capabilities. In section 3, we will discuss the model of document structure used for document representation in our framework. Section 4 describes a document navigation scheme implemented in the framework. Section 5 discusses how we load documents which were processed by GATE. In section 6 the different ways in which documents from TextGrid, plain text documents and plain text documents with attached structural markup can be loaded will be discussed. Section 7 describes a baseline for expression attribution, a more general version of quoted speech attribution (Pareti et al., 2013), implemented in our framework. Section 8 summarizes our contributions and discusses future work.

2 Related Work

Most similar to our framework are the widely used frameworks GATE and UIMA. Both frameworks use an offset-based format and provide a large variety of text analysis capabilities in the form of analysis components made by multiple people. To the best of our knowledge neither GATE nor UIMA represent the hierarchical structure of a document explicitly. In our framework, the hierarchical structure of a document is represented explicitly, which allows analysis capabilities based on the hierarchical structure of documents to be implemented in a straightforward manner.

Also similar to our framework are tools for making text analysis capabilities available to humanities researchers who have no background in machine learning and CL. Two of those tools are the eHumanities desktop (Gleim and Mehler, 2010) and the tool developed in Blessing et al. (2013). The eHumanities desktop is specifically designed for the needs of humanities researchers, the tool described in Blessing et al. (2013) for researchers in political science. To the best of our knowledge neither one contains facilities to represent a hierarchical representation of document structure explicitly.

The WebLicht environment (Hinrichs et al., 2010), a webservice-based orchestration facility for linguistic text processing tools, is largely orthogonal to the approach presented here, since it is centered around a classical linguistic processing chain and does not put emphasis on higher-level document navigation.

3 Model of document structure

We inherently view and analyze literary documents as having a minimal hierarchical structure. This structure consists of a linguistic and an organizational structure and forms the basis for the visualization of document structure. The smallest unit of the minimal structure is a character. Every other unit is then defined by its start and end offset in the text. For example, a token, the next largest linguistic unit in a document, is defined by its start and end offset and additional properties like its part-of-speech tag, its lemma or its relation to other tokens. Tokens are contained in sentences. Therefore characters, tokens and sentences form the minimal linguistic structure of a document.

The minimal organizational structure of a document are textual lines. Lines are the smallest unit of organizational structure. Other common units are paragraphs, pages, sub-chapters and chapters. In most cases, lines are to the organizational structure what characters are to the linguistic structure: The smallest units of organizational structure which all other units are composed of.

The hierarchical representation of documents allows for the generic implementation of a document navigation scheme which will be discussed in the next section.

4 Document navigation

The following concept for document navigation is also used in the tool based on TEANLIS mentioned earlier (Koch et al., 2014). The following example discussed with reference to figure 1 was also presented in this paper.

[4] https://github.com/AndreasMuellerAtStuttgart/TEANLIS

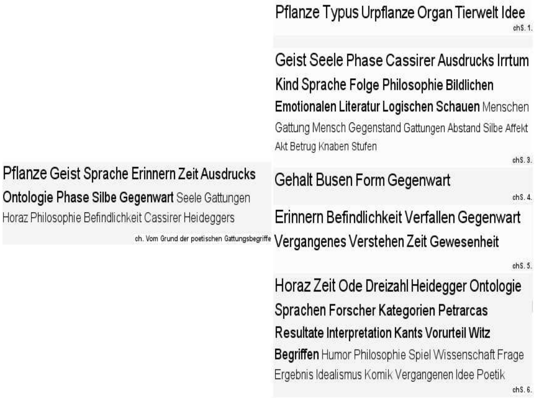

Pflanze Typus Urpflanze Organ Tierwelt Idee
chS. 1.
Geist Seele Phase Cassirer Ausdrucks Irrtum Kind Sprache Folge Philosophie Bildlichen Emotionalen Literatur Logischen Schauen Menschen Gattung Mensch Gegenstand Gattungen Abstand Silbe Affekt Akt Betrug Knaben Stufen
chS. 3.
Pflanze Geist Sprache Erinnern Zeit Ausdrucks Ontologie Phase Silbe Gegenwart Seele Gattungen Horaz Philosophie Befindlichkeit Cassirer Heideggers
ch. Vom Grund der poetischen Gattungsbegriffe
Gehalt Busen Form Gegenwart
chS. 4.
Erinnern Befindlichkeit Verfallen Gegenwart Vergangenes Verstehen Zeit Gewesenheit
chS. 5.
Horaz Zeit Ode Dreizahl Heidegger Ontologie Sprachen Forscher Kategorien Petrarcas Resultate Interpretation Kants Vorurteil Witz Begriffen Humor Philosophie Spiel Wissenschaft Frage Ergebnis Idealismus Komik Vergangenen Idee Poetik
chS. 6.

Figure 1: Example for a segmentation computed by the document navigation scheme (derived from Staiger (1946), graphic taken from Koch et al. (2014))

Word clouds are often used to give an overview over the topics occurring in a text. We provide a generic functionality for computing word clouds from the nouns occurring in the elements of a hierarchical level. For example, we can compute word clouds for every chapter in a book. This is accomplished by using each chapter as a document in a Lucene[5] index. We view each noun as a distinct term in a document and use every noun which occurs in at least one chapter as a search term. The keywords in a word cloud for a given chapter are then the nouns which scored highest when they were used as a search term for that chapter. This mechanism can be used for every level in the hierarchy. By using the lucene standard scoring formula to rank terms, we take the idf (inverse document frequency) of a term into account. Thus, the top keywords for an element on a given level of the hierarchy are not only the nouns with the highest frequency in the element. A noun which has a low frequency in the element but appears in very few or no other elements besides the given element is also likely to appear as a keyword. This follows the intuition behind the TFxIDF formula from information retrieval (Manning et al., 2008).

Some elements, like chapters, usually have multiple topics occurring in them. Those topics can sometimes be seen in a word cloud by looking for keywords which are semantically related. Ideally we would like to know where in the chapter a topic like this is discussed. Therefore, we implemented a generic functionality for splitting a hierarchy element into topic segments using the implementation of the TextTiling algorithm (Hearst, 1997) of the morphadorner library (Burns, 2013). This allows us to compute word clouds for the topic segments. In those word clouds the topics which can be seen in the chapter word cloud can be much more recognizable. An example is shown in figure 1.

Here, the words ”Erinnern” (remembering), ”Zeit” (time) and ”Gegenwart” (present) in the chapter on the left indicate a topic occurring in this chapter. The segments on the right are automatically computed and further segment the chapter. The fourth segment on the right contains those three words and also other words which are related, like ”Vergangenes” (roughly translates to ”that which was” in English). This indicates that the fourth segment on the right discusses the subtopic indicated by the three words in the segment on the left. This is an example for how the topic segmentation of arbitrary hierarchy elements can assist in document navigation.

An example for a concrete research question which can be addressed with a tool based on TEANLIS is discussed in Koch et al. (2014). This example discusses how the document navigation scheme presented in this section can help a literary scholar answer questions about which works and authors associated with different literary styles are discussed in a poetic.

5 GATE Converter

Since our framework is also offset-based converting documents in our format to GATE documents is straightforward as far as the offsets are concerned. For mapping the properties of, for example, tokens, we define a mapping from property names in our framework to the corresponding feature names in GATE. This ensures that features have the names the processing resources which are used to further process a document in GATE expect. A similar mapping is used to convert documents in GATE format to documents in our format.

A particularly useful feature of converting documents in our format to GATE documents is

[5] http://lucene.apache.org/

that it allows the generic use of the JAPE[6] system. JAPE allows the definition of regular expression grammars over annotations and their features. JAPE is easy to use, even for people who have no background in computer science. Therefore, developers can provide access to JAPE (via GATE Developer[7], the graphical interface to a lot of GATE analysis and annotation capabilities) in a simple manner, which could be very useful for literary scholars who are familiar with the JAPE system or are willing to learn it.

Note that by using the Graph Annotation Framework (GrAF) described in Ide and Suderman (2009) we could in principle also convert documents in our format to UIMA documents, because the GrAF framework supports conversion from GATE to UIMA format and back.

6 Generic xml and plain text loader

To access the documents in TextGrid in a generic way, we implemented a loader for documents in TEI (Unsworth, 2011) format and plain text documents with pre-defined structural markup. There are two types of loaders: A minimal loader and a structure-aware plain text loader.

6.1 Minimal loader

The minimal loader works for all documents in xml format which have a tag containing the text of the document. This tag has to be specified. The method simply reads the text content of the xml-element corresponding to the tag and stores it as the text of the document in our format. If possible, the language of the text is extracted from meta-data and a suitable sentence splitter and tokenizer are used to pre-process the text, giving the document at least the minimal linguistic structure.

We support six languages by integrating the OpenNLP tools[8] for sentence splitting and tokenization. The languages are: Danish, German, English, Dutch, Portugese and Swedish[9]. In the generic loader, if no line delimiting characters are given we represent the text in one-sentence-per-line format. If a document is from TextGrid we search for paragraphs by searching for <p>tags. Note that even though the minimal loader does not recognize document structure, the document navigation scheme explained before can automatically compute structure and an overview of the topics contained in the structural elements. This can be done by using topic segmentation on the whole text. Then, the resulting segments can be segmented themselves and so forth.

Note that even though the minimal loader does not recognize document structure, the document navigation scheme explained before can automatically compute structure and an overview of the topics contained in the structural elements.

6.2 Structure aware plain text loader

For text files we also provide the option of inserting structural tags to mark the standard structure our framework recognizes. To get the textual units constituting the structural elements on a given hierarchical level, we simply split the text on the structural tags provided by the user. For example, if PARAGRAPH tags are given we would simply regard all text between two PARAGRAPH tags as one paragraph. If the plain text files have structural tags which do not correspond to our tags but delimit the same units (for example, a PARAGRAPH tag given as a P tag), the user can specify a mapping between the text in the files and our tags (for example, mapping P to PARAGRAPH). This represents a simple way to attach structural markup to a text.

7 A baseline for expression attribution

We already mentioned that TEANLIS is designed to support the development of tools for literary analysis. To this end, we implemented baselines for tasks which are relevant for literary analysis. One of those tasks is expression attribution. Expression attribution is a more general type of quotation extraction (Pareti et al., 2013). For example, expression attribution includes a sentence like: ”It is, as Husserl showed, paradoxical to say they could vary.” which includes an expression of an author (Husserl). This expression is an abstract representation of the expression of Husserl, not something he actually said or wrote exactly as expressed in the sentence.

The baseline is described in detail in a paper

[6] http://gate.ac.uk/sale/tao/splitch8.html#x12-2190008

[7] http://gate.ac.uk/sale/tao/splitch3.html#x6-420003

[8] https://opennlp.apache.org/

[9] http://opennlp.sourceforge.net/models-1.5/

we submitted to STRiX 2014[10]. The paper is currently under review. Part of the following description is taken from that paper. Essentially, our technique extracts triples of the form (person, verb-cue, sentence-id). Person is the utterer of an expression, verb-cue is the verb used to detect the expression (if a verb is used to detect the expression) and sentence-id is the id of the sentence containing the expression. The baseline extracts those triples by detecting sentences which either contain the name of an author and a quotation or the name of an author and a verb indicating the presence of an expression (the verb ”express” or the verb ”say”).

We evaluated the baseline by extracting triples from Staiger (1946). The system identified 64 instances of attributed expressions within the text. We then manually classified the instances into three classes, in order to gain insight into the behavior of the algorithm and to guide future work. If the sentence contained an attributed expression and the utterer was identified correctly, we considered the instance to be annotated fully correct. If the sentence contained an attributed expression but the utterer was not correctly identified, we considered the item to be partially correct. All other instances were considered an error. The first author of the current paper and a colleague from the same institute annotated these classes in parallel, with an initial F_1-agreement of 0.67. Differences have been adjudicated after discussion with a domain expert.

Our baseline identifies 62.5% of the utterances correctly, and for 51.6% the correct utterer was also identified.

8 Future Work

We presented a framework for developing tools to support the analysis of texts with a hierarchical structure in general and literary texts in particular. Our framework is different from the established frameworks GATE and UIMA in that it provides analysis capabilities based on the recognition of the hierarchical structure of a text. It also provides facilities for computing the hierarchical structure of arbitrary texts in a semi-automatic manner. Also, our documents can be converted to GATE documents and conversely. This allows the integration of the analysis capabilities provided by GATE. Through the GrAF framework we can theoretically convert our documents to UIMA documents, which is something we want to investigate in the future.

We plan to integrate other topic segmentation algorithms, especially algorithms for hierarchical topic segmentation like the one described in Eisenstein (2009). We are also in the process of writing genre-specific converters to convert documents from the TextGrid repository to our documents and take their existing structure into account. This can be done by recognizing predefined structural elements, like recognizing page breaks by searching for the <pb>tag. Documents from different genres are then distinguished by which of those tags they use to mark structure. This allows developers to access a large repository of literary documents without having to write converters of their own.

We are also in the process of implementing baselines for CL tasks which can benefit literary analysis. One example is the baseline for expression attribution discussed in the last section. Another type of tasks are text classification tasks like classifying paragraphs with respect to what theme they talk about. In a first baseline, themes are ”aesthetic” and ”poetic”. Paragraphs are classified based on typical words for the themes provided by the literary scholar in our project. Although we have not formally evaluated those baselines yet we observed that they work quite well on the text in our corpus we tested them on. However, the point of implementing those baselines is to provide developers with a starting point for quickly getting results for those tasks. This enables them to see how well obvious baselines perform on their data and to assess the specific problems of the task with respect to their data.

Acknowledgments

This work has been funded by the German Federal Ministry of Education and Research (BMBF) as part of the ”ePoetics” project.

[10] http://spraakbanken.gu.se/eng/strix2014

References

Andre Blessing, Jonathan Sonntag, Fritz Kliche, Ulrich Heid, Jonas Kuhn, and Manfred Stede. 2013. Towards a tool for interactive concept building for large scale analysis in the humanities. In *Proceedings of the 7th Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, pages 55–64, Sofia, Bulgaria, August. Association for Computational Linguistics.

Philip R. Burns. 2013. Morphadorner v2: A java library for the morphological adornment of english language texts. October.

Hamish Cunningham, Valentin Tablan, Angus Roberts, and Kalina Bontcheva. 2013. Getting more out of biomedical documents with gate's full lifecycle open source text analytics. *PLOS Computational Biology*.

Jacob Eisenstein. 2009. Hierarchical text segmentation from multi-scale lexical cohesion. In *Proceedings of Human Language Technologies: The 2009 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, NAACL '09, pages 353–361, Stroudsburg, PA, USA. Association for Computational Linguistics.

David Ferrucci and Adam Lally. 2004. Uima: An architectural approach to unstructured information processing in the corporate research environment. *Nat. Lang. Eng.*, 10(3-4):327–348, September.

Rüdiger Gleim and Alexander Mehler. 2010. Computational linguistics for mere mortals — powerful but easy-to-use linguistic processing for scientists in the humanities. In *Proceedings of LREC 2010*, Malta. ELDA.

Marti A. Hearst. 1997. Texttiling: Segmenting text into multi-paragraph subtopic passages. *Comput. Linguist.*, 23(1):33–64, March.

Mark Hedges, Heike Neuroth, Kathleen M. Smith, Tobias Blanke, Laurent Romary, Marc Kster, and Malcolm Illingworth. 2013. Textgrid, textvre, and dariah: Sustainability of infrastructures for textual scholarship. *Journal of the Text Encoding Initiative [Online]*, June.

Erhard W. Hinrichs, Marie Hinrichs, and Thomas Zastrow. 2010. WebLicht: Web-Based LRT Services for German. In *Proceedings of the ACL 2010 System Demonstrations*, pages 25–29.

Nancy Ide and Keith Suderman. 2009. Bridging the gaps: Interoperability for graf, gate, and uima. In *Proceedings of the Third Linguistic Annotation Workshop*, ACL-IJCNLP '09, pages 27–34, Stroudsburg, PA, USA. Association for Computational Linguistics.

Steffen Koch, Markus John, Michael Wörner, Andreas Müller, and Thomas Ertl. 2014. Varifocalreader - in-depth visual analysis of large text documents. In *To appear in: IEEE Transactions on Visualization and Computer Graphics (TVCG)*.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to Information Retrieval*. Cambridge University Press, New York, NY, USA.

Silvia Pareti, Timothy O'Keefe, Ioannis Konstas, James R. Curran, and Irena Koprinska. 2013. Automatically detecting and attributing indirect quotations. In *EMNLP*, pages 989–999. ACL.

Emil Staiger. 1946. *Emil Staiger: Grundbegriffe der Poetik*. Atlantis Verlag Zürich.

John Unsworth. 2011. Computational work with very large text collections. *Journal of the Text Encoding Initiative [Online]*.

Labelling Business Entities in a Canonical Data Model

Nathali Ortiz Suarez
SAP SE
Dietmar-Hopp-Allee 16
69190 Walldorf
nathali.ortiz.suarez@sap.com

Jens Lemcke
SAP SE
Vincenz-Prienitz-Str. 1
76131 Karlsruhe
jens.lemcke@sap.com

Ulrike Pado
HFT Stuttgart
Schellingstr. 24
70174 Stuttgart
ulrike.pado@hft-stuttgart.de

Abstract

Enterprises express the concepts of their electronic business-to-business (B2B) communication in individual ontology-like schemas. Collaborations require merging schemas' common concepts into Business Entities (BEs) in a Canonical Data Model (CDM). Although consistent, automatic schema merging is state of the art, the task of labeling the BEs with descriptive, yet short and unique names, remains. Our approach first derives a heuristically ranked list of candidate labels for each BE locally from the names and descriptions of the underlying concepts. Second, we use constraint satisfaction to assign a semantically unique name to each BE that optimally distinguishes it from the other BEs.

Our system's labels outperform previous work in their description of BE content and in their discrimination between similar BEs. In a task-based evaluation, business experts estimate that our approach can save about 12% of B2B integration effort compared to previous work and about 49% in total.

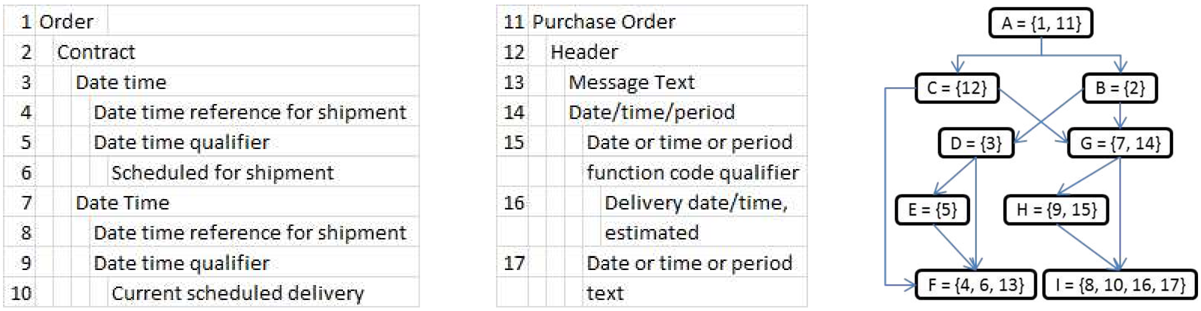
1 Introduction

Businesses often exchange electronic messages like Purchase Orders, which contain compatible concepts (e.g., shipment dates and delivery address) that are however arranged and named differently in each company's ontology-like messaging standards (schemas). For instance, the two exemplary schemas shown on the left-hand side of Fig. 1 both speak about the delivery date, but use different phrases – "Current Scheduled Delivery" (node 10) and "Delivery Date/Time, estimated" (node 16). Misinterpretation is likely and may lead to delays and other financial losses.

The solution is to align the participating enterprises' schemas and find new, unique and appropriate (natural-language) names for the contained concepts, for all participants to use. A solution for the alignment task has been proposed in Lemcke et al. (2012): They create a CDM made up of BEs which can be visualised as clusters of equivalent nodes of the original schemas as visualized on the right-hand side of Fig. 1. This is similar to Ontology Merging (Shvaiko and Euzenat, 2013) except that the relation between the nodes is "part-of" and has to be maintained consistently.

As described in Lemcke et al. (2012), the only reliable source for correspondences between the schema nodes are the mappings business experts create when integrating two systems. Analysing the mappings shows that, for example, the delivery date is expressed in schema 1 by the value of node 8 in the "Date time" structure, together with the "Current scheduled delivery" qualifier (node 10). In schema 2, this corresponds to the combination of nodes 16 and 17. Therefore, BE I containing nodes 8, 10, 16 and 17 is created.

In this paper, we tackle the problem of automatically finding short, descriptive and unique natural-language labels for each of the BEs to replace the symbolic names F or I, based on the names and descriptions provided for each of the original nodes (see Table 1). The desired result are labels like



1	Order
2	Contract
3	Date time
4	Date time reference for shipment
5	Date time qualifier
6	Scheduled for shipment
7	Date Time
8	Date time reference for shipment
9	Date time qualifier
10	Current scheduled delivery

11	Purchase Order
12	Header
13	Message Text
14	Date/time/period
15	Date or time or period
	function code qualifier
16	Delivery date/time,
	estimated
17	Date or time or period
	text

Figure 1: Two exemplary input schemas and corresponding Canonical Data Model (CDM)

BE	Node	Name of node	Description of node
F	4	Date time reference for shipment	To specify pertinent dates and times.
	6	Scheduled for shipment	
	13	Message Text	To provide a free-form format that allows the transmission of text information
I	8	Date time reference for shipment	To specify pertinent dates and times.
	10	Current scheduled delivery	
	16	Delivery date/time, estimated	Date and/or time when the shipper of the goods expects delivery will take place.
	17	Date or time or period text	To specify date, and/or time, or period.

Table 1: Exemplary BEs and nodes. Texts taken from B2B standards UN/EDIFACT (http://www.unece.org/cefact/edifact/welcome.html) and ASC X.12 (http://www.x12.org/).

Shipment Date and *Delivery Date*.

The labelling task is complicated by the limited vocabulary of the description data, since controlled terms from a strictly defined domain are used. For example, both BE description sets in Table 1 contain the words *date*, *shipment* or *scheduled*. Since we see fewer distinct content words than BEs, labels must be phrases. Also, we have to balance the need for short labels with specificity and discrimination amongst semantically similar BEs.

Further, reusing the same node defined by some schema template in different contexts is very common in B2B integration. For example, the date and time structures of node 4 and 8 in Table 1 can be interpreted either as a shipment or a delivery date, depending on whether they appear in conjunction with the qualifier node 6 (in BE F) or 10 (in BE I). This means that words and concepts introduced by different usage contexts of nodes are commonly used in BE descriptions.

Also, free text nodes like node 13 are commonly misused to store e.g. the shipment date. Both factors result in noise in the form of misleading words in the accumulated descriptions of a BE.

We clarify our assumptions about what defines a good label in Section 2. Based on these rules, our approach for labelling the CDM is described in Section 3. Note that the approach is completely domain- and mostly task-agnostic and could be used in other settings where short texts are involved. We present evaluation results with respect to label quality and time saved in Section 4.

2 Desiderata for Labels

An optimal labelling is reached when the following assumptions are true: Labels are natural language words or phrases that are:

Descriptive The label should state the concept of the BEs. Therefore, the concepts which are most frequently present in the names and descriptions of a BE are good label candidates.

Discriminative The label should state the distinguishing property of the BE. Therefore, the best candidates for labelling a BE are concepts which are frequently present in its names and descriptions, but not in the overall CDM.

Short The label should balance shortness (by Occam's razor) and specificity (to achieve uniqueness and discriminate between BEs).

Semantically Unique Two BEs must have non-synonymous labels. As the CDM has reference character for business experts, it is necessary to assign unique labels for unique BEs.

3 Labelling Business Entities

We use the approach developed in-house by (Dietrich et al., 2010). They introduce the tool pipeline shown in Fig. 2 to solve the labelling problem for the CDM. The Dietrich et al. approach generates label candidates from the node names and descriptions for each BE and validates them against a domain lexicon and search results in three search engines. However, due to data sparseness in both types of resources, correct label candidates are often erroneously rejected. Further, the approach conflates different senses of the same word. We address both of these issues below.

We also use a new strategy for labelling the CDM: First, we generate plausible label candidates for each BE and rank them heuristically. Second, we optimize globally, picking the set of labels for the CDM with the best overall ranks. This is similar to the global inference strategy, which recently has become increasingly popular (cf. work starting with Roth and Yih (2004)).

We now describe how we use and extend the tools from Fig. 2 to create labels with the properties defined in Section 2. Note that for both BE names and (possibly noisy) descriptions, processing is the same. We do, however, give more weight to the candidates extracted from the (cleaner) BE names. From here on, we use d_x as a placeholder to refer interchangeably to the names or the descriptions of the specific BE be_x.

Descriptive labels For descriptive labels, we need to find the most representative concept in a BE be_x. One strategy could be to look for domain terms which can be assumed to be relevant, but the Dietrich et al. results indicate that existing resources are too sparse for this. Therefore, we consider every term in the BE names and descriptions. To be agnostic of synonyms, our adapted synonym finder first extracts all possible meanings of each term t by retrieving the synsets $S_t = \{s_1, s_2, \ldots, s_n\}$ from WordNet (Fellbaum, 1998). Further, S_t is extended by the synsets of derivationally related forms of t as returned by WordNet. To increases the possibility of overlaps of the synsets of different, related terms, especially when used as different POS. The frequency of the synset s among the synsets of all terms of the names and descriptions d_x of the BE be_x, denoted as $f(s, d_x)$, indicates the relevance of s for describing be_x. We normalize the frequency over all be_x's synsets $S_{d_x} = \bigcup_{t \in d_x} S_t$ as in the term frequency (TF) approach by

$$tf(s, d_x) = \frac{f(s, d_x)}{\max\{f(s, d_x) : s \in S_{d_x}\}}.$$

In contrast to solely TF, the full TF/IDF approach did not yield satisfactory results: We found that since a BE's core concept may frequently appear in other BEs' descriptions due to re-use of nodes in different contexts and the misuse of free-text nodes, the IDF term was commonly very small and erroneously filtered out the true core concept.

For the final creation of labels, we express a synset s by the most frequent term t from d_x with $s \in S_t$ to adapt to the common technical terms of the domain.

Discriminative labels As there are fewer interesting words than BEs, word selection by TF does not produce unique labels, and phrases are needed. We use the description *The field represents the contract date representing the current scheduled delivery* to demonstrate how these are generated. First, nouns, verbs, adjectives and adverbs are identified as interesting words to build phrases. As an alternative design decision, each interesting term is then represented by its most frequent WordNet synset as described before and illustrated in Table 2. However, another alternative could be for example representing each interesting term by its first common hyperonym.

Second, our adapted phrase generator passes a sliding window over the text and considers all synset sequences in the window as possible candidates. With this window which was chosen heuristically, we both ensure some local coherence between the candidates and limit the numbers of possible combinations. For our running example we use a sliding window of size 4. We compute the relative distance of the synsets based on their position in the sentence. (E.g., *delivery* at position 11 and *current* at position 9 are two units apart.)

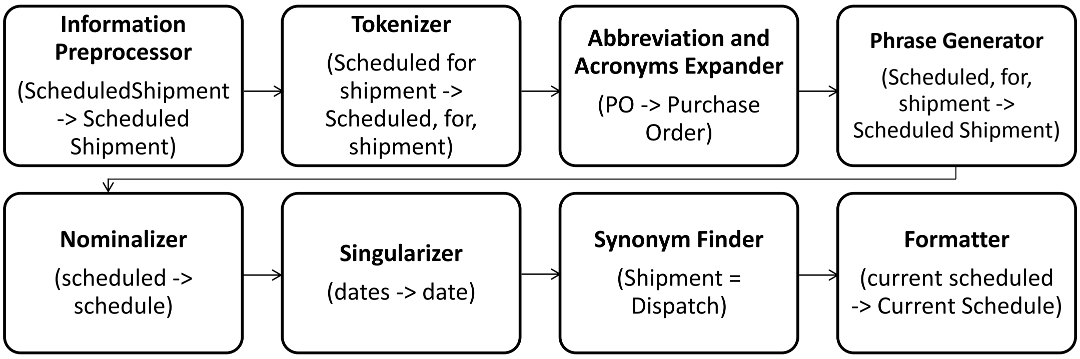

Information Preprocessor
(ScheduledShipment -> Scheduled Shipment)
Tokenizer
(Scheduled for shipment -> Scheduled, for, shipment)
Abbreviation and Acronyms Expander
(PO -> Purchase Order)
Phrase Generator
(Scheduled, for, shipment -> Scheduled Shipment)
Nominalizer
(scheduled -> schedule)
Singularizer
(dates -> date)
Synonym Finder
(Shipment = Dispatch)
Formatter
(current scheduled -> Current Schedule)

Figure 2: Tool pipeline for generating BE label candidates

Token	field	represents	contract	date	representing	current	scheduled	delivery
Position	2	3	5	6	7	9	10	11
POS	N	V	N	N	V	A	A	N
Synset	S1	S2	S3	S4	S2	S5	S6	S7

Table 2: Representation of each term as position in sentence, POS, and most frequent synset

Tag Pattern	**Example**
AN	Scheduled Delivery
NN	Reference Shipment
AAN	Current Scheduled Delivery
ANN	Added Tax Delivery
NAN	Reference Scheduled Delivery
NNN	Date Time Shipment
NPN	Reference for Shipment

Table 3: Justeson and Katz (1995) phrase patterns

The lower the relative distance, the more likely the phrase is to be useful, because it is present (almost) verbatim in the input. Third, for avoiding redundancy, we filter out synset sequences that contain duplicate synsets. Fourth, our adapted formatter chooses the most relevant word from the input sequences for each combination of synset and POS tag. The resulting phrase has to correspond to the POS tag sequences proposed in Justeson and Katz (1995) shown in Table 3. The input sequence in Table 2 yields phrases like *field representation*, *contract date representation*, *scheduled delivery*, *current scheduled delivery* and *current scheduled*. *Current scheduled* matches no pattern in Table 3, so it is changed to *current schedule*.

We estimate the quality of the phrases heuristically instead of checking against lexical resources. We use the length $le = |\mathbf{p}|$ of the phrase $\mathbf{p}$ to rank more specific phrases higher. We also consider the average frequency

$$\overline{wf} = \frac{\sum_{t \in \mathbf{p}} tf(t, d_x)}{le}$$

of the words of the phrase $\mathbf{p}$ in the names or descriptions of the BE be_x, favouring labels with more descriptive terms.

Short labels The previous step prefers relevant, but longer labels. We balance this preference with two measures that discourage long phrases: We consider the reciprocal of the average distance $\overline{di}^T = \frac{le-1}{di}$ of the words in a phrase, where di is the distance between the first and the last word of the compound in the original text, favouring short phrases taken literally from the text. The frequency $pf = tf(\mathbf{p}, d_x)$ of the phrase in the names or descriptions of be_x has a similar effect because longer phrases tend to be less frequent.

The final ranking of label candidates uses all four measures (length, average word frequency, inverse average word distance, and phrase frequency), each normalized over all candidates.

All weights are equal, except that we weight the measures for extracting phrases from the BE names twice as high as the measures for extracting from BE descriptions, since the names are defined by experts and contain less noise than was observed in the BE descriptions. This decision was supported by our analysis of results on the development set.

Semantically unique labels Finally, one of the locally generated phrases needs to be assigned to each BE, but not two BEs can get synonym labels. To solve this problem in a globally optimal way, we formulate the constraints and variables of a Constraint Satisfaction Problem (CSP). The CSP is solved by Choco 2.1.3 (choco Team, 2010), a very general constraint satisfaction framework.

Each BE be_x is represented by the variables $label$ (candidate phrases), $synsets$ (synset sequence for each phrase) and $rank$ (rank in terms of our heuristics).

A set of feasible tuples constraints ensures that $label$, $synsets$ and $rank$ are internally consistent for each BE be_x. Another two sets of all-different constraints ensure uniqueness among the values assigned to the $label$ and respectively to the $synsets$ variables, i.e., labels have to be unique both in terms of tokens and of concept. The system maximizes the formula $\sum_x rank_x$.

The complexity of the CSP depends most strongly on the size of the CDM, i.e., number b of BEs, and the window size w when generating phrases. The number of phrases, which make up the domains of the $label$ variables, depends exponentially on the window size and linearly on the length of names and descriptions. The CSP itself has exponentially many solutions depending on the number of BEs. So, the total worst-case complexity is $\mathcal{O}(2^{wb})$. In our case, with a $w = 5$ and $b = 25$ the computational time is approximately 3 hours and with the same w but $b = 38$ it is approximately 6.45 hours.

4 Evaluation

For evaluation, we compare to the baseline approach by Dietrich et al. (2010). We use 38 BEs that were unseen during the development of the tool pipeline. This data has the disadvantage of being proprietary, but there is not, to our knowledge, a comparable freely-available data set.

Our first objective is to establish the need for enforcing **unique labels**. Recall that our approach is designed to never assign the same label to different BEs. We automatically analysed the names proposed by the baseline approach, which assigns non-unique labels to 21% of the BEs. This is not acceptable in practice, since the point of the CDM is to allow unambiguous communication.

	Our	**BL**
Correct	70.3%	60.2%
Incorrect	29.7%	39.8%

Table 4: Descriptive and discriminative labels: Percentage of correct label-description pairings for our and the Baseline (BL) approach

The second part of our evaluation focuses on the **descriptive** and **discriminative** properties of our labels. This evaluation was done by ten novice users (due to the limited availability of experts). They assessed whether the label assigned to a BE correctly reflects its distinguishing features. In the survey, the participants answered 20 questions (ten for each approach). The participants saw the top-ranked BE label as generated by one of the systems, as well as the description of the input BE and the descriptions of semantically similar distractor BEs. If the participant chose the input description as best matching the label, we took that to mean that the label correctly distinguishes the semantics of the BE from the others.

The results of this survey are shown in Table 4. For the baseline approach, the participants chose the correct description for the label in 60.2% of the time, as opposed to 70.3% of the time for our approach. A X^2-test with a null hypothesis of chance assignment of correct and incorrect labels is significant at the 0.05 level; we conclude that our labels are more discriminative among BEs and describe BE content better than the labels returned by the baseline approach.

Finally, we present a task-based evaluation that was carried out with the help of B2B experts. Our objective here is to show that our system is useful in a real-world setting to the very group of people who are its intended users. Nine B2B experts estimated how much time and effort they would have saved creating the labels with the help of the output data of the approaches. The survey used five BEs and had three kinds of questions:

First, the participants were asked to create a label for a BE by hand, based on the names and descriptions available for it. These names and descriptions were also input to the systems.

Second, based on their manually created label, the participants estimated how much effort they could have saved in step 1 if they had had available

Effort Saved	Our	BL
$\geq$ 90%	8	4
90-75%	5	2
75-50%	5	5
50-30%	13	12
$\leq$ 30%	14	22
Avg (%)	49.2	37.1

Rank	Our	BL
1	36	9
2	26	19
3	21	24
4	18	27
5	19	26
6	15	31
Avg	3.02	3.99

Table 5: Task-based evaluation of label usefulness to experts: Result for evaluating effort saved (left) and label rank according to usefulness (right) of our and the baseline (BL) approach

the label candidates by one of the approaches. The participants chose one of five levels: more than 90% (when the label in step 1 is almost equal to the proposed candidates), between 90 and 75%, between 75 and 50%, between 50 and 30% and less than 30% (when the label is completely different).

Third, six model labels, three from each approach, had to be ranked in order of their usefulness for creating their label.

Table 5 shows the result for the effort-saved estimation on the left-hand side. We computed the average amount of effort saved by using the mid-point for each of the categories, e.g. 82.5 for the 90-75% category. Our approach saved 12.1 percentage points more expert effort than the baseline, and 49.2% of total effort. This corresponds to about four working hours (out of an eight-hour day). The baseline approach would allow the experts to save about three working hours, so using our approach saves an additional hour of (highly-qualified and highly-compensated) expert times.

The right-hand side of Table 5 shows the summarized results from the ranking task. Numerically, the experts ranked our proposals on average one rank higher than the baseline proposals. X^2-tests with the null hypothesis of an equal number of total observations in each rank found that the numerical differences for rank 1 and 6 are statistically significant at the 0.05 level. Overall, these results again illustrate that proposals given by our approach will be more useful for the experts in label creation than the baseline system.

5 Related Work

This paper is concerned with labelling a merged ontology in an unsupervised way given the node names and descriptions from the source ontologies. To our knowledge, this task is not commonly treated in the ontology merging literature.

In computational linguistics, our task is most comparable to the problem of assigning keywords or index terms that best describe a document's content (see, e.g., Kim et al. (2010)). However, our data is shorter, more repetitive and more ambiguous than running text from scientific publications or newspapers, and we have to obey the additional constraint of finding unique labels.

The labelling task is also somewhat reminiscent of the task of finding appropriate names for FrameNet framesets in the SemFinder system (Green and Dorr, 2004). Green and Dorr use WordNet synsets and glosses as their input data and rely heavily on WordNet's tree structure. This strategy is however infeasible for highly domain-specific texts like ours.

6 Conclusions

This paper proposed a method for labelling the BEs of a CDM by analysing the aggregated names and descriptions underlying the BEs, assuming that appropriate labels should be descriptive, discriminative, short and semantically unique.

Our strategy is very general and can be applied to other tasks inside and outside the ontology labelling domain. Several properties of the B2B domain challenged our implementation: Re-use and misuse of structural elements caused notable noise in the input data and the limited vocabulary of controlled terms means that the same relevant terms and concepts appear in multiple BEs.

We therefore applied the strategy of generating phrases as label candidates locally and then picking globally optimal label candidates. This strategy ensures unique labels, which are a core requirement in our domain.

Our evaluation showed that our labels are more descriptive of BE content and discriminate better among similar BEs than the baseline. A task-based evaluation with B2B experts, who are the intended users of the system, suggests potential effort savings in this crucial task of B2B integration of almost 50%, corresponding to four working hours out of an eight-hour work day.

References

choco Team. 2010. choco: an Open Source Java Constraint Programming Library. Research report 10-02-INFO, École des Mines de Nantes.

Michael Dietrich, Dirk Weissmann, Jörg Rech, and Gunther Stuhec. 2010. Multilingual extraction and mapping of dictionary entry names in business schema integration. In *Proceedings of the 12th International Conference on Information Integration and Web-based Applications & Services*, pages 863–866.

Christiane Fellbaum. 1998. *WordNet: An Electronic Lexical Database*. Bradford Books.

Rebecca Green and Bonnie J Dorr. 2004. Inducing a semantic frame lexicon from wordnet data. In *Proceedings of the 2nd Workshop on Text Meaning and Interpretation*, pages 65 – 72.

John S. Justeson and Slava M. Katz. 1995. Technical terminology: Some linguistic properties and an algorithm for identification in text. *Natural Language Engineering*, 1(1):9–27.

Su Nam Kim, Olena Medelyan, Min-Yen Kan, and Timothy Baldwin. 2010. Semeval-2010 task 5 : Automatic keyphrase extraction from scientific articles. In *Proceedings of the 5th International Workshop on Semantic Evaluation*, pages 21–26.

Jens Lemcke, Gunther Stuhec, and Michael Dietrich. 2012. Computing a canonical hierarchical schema. In *Proceedings of the I-ESA Conferences Volume 5*, pages 305–315.

Dan Roth and Wen-tau Yih. 2004. A linear programming formulation for global inference in natural language tasks. In *Proceedings of the Conference on Natural Language Learning*, pages 1–8.

Pavel Shvaiko and Jérôme Euzenat. 2013. Ontology matching: State of the art and future challenges. *IEEE Trans. Knowl. Data Eng.*, 25(1):158–176.

Detecting Comparative Sentiment Expressions – A Case Study in Annotation Design Decisions

Wiltrud Kessler and **Jonas Kuhn**
Institute for Natural Language Processing
Universität Stuttgart
Pfaffenwaldring 5b, 70569 Stuttgart
wiltrud.kessler@ims.uni-stuttgart.de

Abstract

A common way to express sentiment about some product is by comparing it to a different product. The anchor for the comparison is a comparative predicate like "better". In this work we concentrate on the annotation of multiword predicates like "more powerful". In the single-token-based approaches which are mostly used for the automatic detection of comparisons, one of the words has to be selected as the comparative predicate. In our first experiment, we investigate the influence of this decision on the classification performance of a machine learning system and show that annotating the modifier gives better results. In the annotation conventions adopted in standard datasets for sentiment analysis, the modified adjective is annotated as the aspect of the comparison. We discuss problems with this type of annotation and propose the introduction of an additional argument type which solves the problems. In our second experiment we show that there is only a small drop in performance when adding this new argument type. [1]

1 Introduction

Sentiment analysis is an area in Natural Language Processing that deals with the task of determining the polarity (positive, negative, neutral) of an opinionated document, sentence or other text unit. In product reviews, sentiment is usually determined with regard to some target product, e.g., the sentence "X has a good lens" expresses positive sentiment towards X. A common way to express sentiment about some product is by comparing it to a different product. As many standard approaches assume one polarity to be assigned to one target entity, they cannot deal with comparisons which involve more than one target entity and may involve more than one polarity. It is thus necessary to analyze comparisons separately.

For our purposes we define a comparison to be any statement about the similarity or difference of two entities. Comparative sentences in the linguistic sense ("X is better than Y" or "X is the best") are included in this definition and indeed many comparisons are of this form, but user generated texts also contain many more diverse statements, e.g., "X blows away all others".

In most popular sentiment corpora to date, comparisons are anchored on one word that "expresses the comparison" (*comparative predicate*) which has three arguments: the two *entities* that are compared and the *aspect* they are compared in (Jindal and Liu, 2006b; Kessler et al., 2010). Most comparative predicates are single words like "better" or "best", but English grammar rules systematically introduce multiword predicates. Consider the following variations of a sentence (predicates in bold, arguments in brackets):

(1) a. "[It]$_{\text{entity1}}$ had a **sturdier** [feel]$_{\text{aspect}}$."
b. "[It]$_{\text{entity1}}$ had a **less sturdy** [feel]$_{\text{aspect}}$."

Sentence 1a compares the aspect "feel" of a camera to some other camera with the comparative predicate "sturdier". If we change the direc-

[1] This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Page numbers and proceedings footer are added by the organizers. License details: http://creativecommons.org/licenses/by/4.0/

tion of the comparison, we get a multiword predicate with the modifier "less" added (sentence 1b). In the following we will refer to all such modifiers as *function words* and to the modified adjective heads as *content words*.

In the literature to date, most approaches to automatically detect comparative predicates are single-token-based. For multiword predicates these approaches require one word to be chosen as the comparative predicate (either the function word or the content word, respectively). The first question we want to address in this case study is how this design decision influences the classification performance of a machine learning system trained to detect comparisons.

In many available corpora, the function word is annotated as the comparative predicate and the content word is annotated as the aspect. This creates the counterintuitive situation that changing the direction of the comparison may introduce a new aspect. Also, annotation schemes that define only one aspect slot force a decision whenever a content word and a real aspect are present. This may lead to loss of information or annotation inconsistencies. We propose to solve these problems by introducing a new argument type and as a second question in this case study investigate the effect on performance.

2 Related Work

The syntax and semantics of comparatives have been the topic of research in linguistics for quite some time (Bresnan, 1973; von Stechow, 1984). In the context of sentiment analysis, Jindal and Liu (2006a) are the first to propose an approach for the identification of sentences that contain comparisons. Their system uses class sequential rules based on keywords as features for a Naive Bayes classifier. In this work we assume that we are given a set of such sentences and aim at identifying the components of the comparisons.

Several approaches have been presented for the detection of comparative predicates and arguments. In follow-up work on their sentence identification, Jindal and Liu (2006b) detect comparison arguments with label sequential rules and in a second step identify the preferred entity in a ranked comparison (Ganapathibhotla and Liu, 2008). Semantic Role Labeling has inspired approaches that detect predicates and subsequently their arguments, those have been applied to Chinese (Hou and Li, 2008) and English (Kessler and Kuhn, 2013). Xu et al. (2011) use Conditional Random Fields to extract relations between two entities, an attribute and a predicate phrase.

All these studies assume a specific way of annotating comparative predicates and arguments and do not investigate the impact this design decision has on actual classification results.

3 Multiword Predicates

Multiword predicates account for about 10-20% of comparative predicates in our data. Some are expressions like "X has the edge over Y" or "X is on par with Y" which we will not discuss in this work. The focus of this study are multiword predicates like "less sturdy" which are systematically introduced by English grammar rules for expressing comparisons. These constitute the majority of multiword predicates and are composed of a modifying function word and a content word. Besides the modifiers "less" / "more" for comparative forms, and "most" / "least" for the superlative, the list of function words includes "as" which is used to introduce an equative comparison like "X is as good as Y".[2]

In the literature to date, single-token-based approaches are mostly used for the automatic detection of comparative predicates. A strong argument can be made to select the function word as the token anchor for the comparative predicate. There will be more training instances to use in machine learning for a given function word than for the individual content words, so sparseness is reduced. On the other hand, choosing the content word may be more informative for end users.

The first question we want to investigate in this study is whether the different annotation decisions translate into a difference in classification performance. In our first experiment we identify all occurrences of multiword predicates. In one setting (*function predicates*), we annotate the modifying function word as the comparative predicate. In the second setting (*content predicates*), we annotate the modified content word.

[2] Note that not all occurrences of the keywords indicate multiword predicates, e.g., in "X has less noise" the word "noise" is not part of the predicate but the compared aspect.

The following illustrates the different annotations for an example sentence:

(2) a. "…had a **less** [sturdy]$_{\text{aspect}}$ [feel]$_{\text{aspect}}$ …" *(function predicates)*

b. "…had a less [**sturdy**]$_{\text{aspect}}$ [feel]$_{\text{aspect}}$ …" *(content predicates)*

In both cases we have the same number of comparative predicates, only the annotations differ. Argument annotations are identical.

The second question deals with the annotation of the content word when we use function predicates. Most corpora annotate the content word as an aspect. We will illustrate some problems with this approach in the following examples:

(3) a. "…a **sturdier** [feel]$_{\text{aspect}}$ …"

b. "…a **less** [sturdy]$_{\text{aspect}}$ [feel]$_{\text{aspect}}$ …"

c. "…a **less** [sturdy]$_{\text{aspect}}$ feel …"

d. "…a **less** sturdy [feel]$_{\text{aspect}}$ …"

e. "…a **less** flimsy [feel]$_{\text{aspect}}$ …"

If we compare sentences 3a and 3b we see that changing the direction of the comparison introduces a new aspect. This is counterintuitive because what is compared (i.e., the aspect) should not depend on the introduced ranking. Additionally, if there is only one slot for the aspect, as is the case in one of the corpora we use, annotators will need to decide between annotations 3c and 3d. Annotation 3c is inconsistent when compared to annotation 3a as both compare the same property of the product but have different annotations for aspect. With annotation 3d we lose information about the actual sentiment polarity that is expressed as we are not able to distinguish it from the annotation in sentence 3e.

To solve these issues, we propose to introduce a separate argument with the sole purpose of modeling the content word in a multiword predicate. In our second experiment we use function words as predicates and change the label of the content word from aspect (used in *function predicates*) to this new argument we will call scale (*function preds. w. scale*) to determine the influence on argument classification. This results in the following annotations being compared:

(4) a. "…had a **less** [sturdy]$_{\text{aspect}}$ [feel]$_{\text{aspect}}$ …" *(function predicates)*

b. "…had a **less** [sturdy]$_{\text{scale}}$ [feel]$_{\text{aspect}}$ …" *(function predicates with scale)*

	J&L	J-C	J-A	IMS
total preds.	668	642	1327	2108
multiword preds.	36	71	127	245
– more	*13*	*26*	*68*	*123*
– less	*4*	*6*	*12*	*18*
– most	*2*	*1*	*4*	*12*
– least	*0*	*0*	*1*	*1*
– as	*17*	*38*	*42*	*91*

Table 1: Multiword predicates in the data.

The tasks of predicate and argument identification are independent of argument labels, so the only change will be in argument classification. We expect a drop in classification performance due to the increased number of classes, but hope that the drop is not significant as the new argument class is well-defined and should be relatively easy to distinguish from real aspects.

4 Data

We use four datasets in our experiments: the J&L data[3] (Jindal and Liu, 2006b), the camera (J-C) and car (J-A) parts of the JDPA corpus[4] (Kessler et al., 2010), and our own set of camera reviews (IMS)[5] (Kessler and Kuhn, 2014).

We extract all sentences where we find at least one comparative predicate. Table 1 contains some statistics about the number of multiword predicates in these datasets.

In the JDPA data the function word is annotated as the comparative predicate and the content word as the aspect. For every annotated predicate that matches our function word keywords, we check if the token directly following the predicate is annotated as the aspect. If the predicate is "as", we take the aspect as the content word. For the other function words we use the word only if it is an adjective (as determined by the Stanford POS Tagger). This serves to distinguish "less sturdy" which we want to include in our experiments from "less noise" where the noun "noise" should be the aspect, not part of the predicate.

[3] http://www.cs.uic.edu/~liub/FBS/data.tar.gz

[4] http://verbs.colorado.edu/jdpacorpus/

[5] http://www.ims.uni-stuttgart.de/forschung/ressourcen/korpora/reviewcomparisons/

		P	R	F1	Δ
J&L	function preds.	**76.4**	**66.8**	**71.3**	
	content preds.	75.2	60.9	67.3	-4.0
J-C	function preds.	74.3	**59.3**	**66.0**	
	content preds.	**75.6**	55.0	63.7	-2.3
J-A	function preds.	**74.6**	**59.5**	**66.2**	
	content preds.	74.5	53.2	62.1	-4.1
IMS	function preds.	84.4	**76.4**	**80.2**	
	content preds.	**84.5**	72.6	78.1	-2.1

Table 2: Results predicate identification.

In the J&L data the complete phrase "as X as" is annotated as the predicate. We check if the first and last word of a predicate is "as", and take the words in between as content words. For the other function words annotation is like in the JDPA corpus, so we proceed the same way.

In our IMS data, the function word is always annotated as the predicate. The content word is annotated as a separate argument scale which we can use directly. For the first experiment we map the scale annotations to aspect.

The resulting annotations for JDPA and J&L are a bit noisy, but manual inspection shows that nearly all of the content words are correctly identified. We miss some occurrences of multiword predicates in cases where some other aspect is present and has been annotated instead of the content word (cf. example 3d).

5 Experiments

Setup. We use the MATE Semantic Role Labeling system (Björkelund et al., 2009)[6] with default settings and without the re-ranker. We re-train the system on our datasets to identify comparative predicates and arguments. We perform three classification steps: predicate identification, argument identification and argument classification. The classification uses features based on the output of a dependency parser. Features are extracted for predicates and arguments as well as the predicate head, predicate dependents and the path between argument and predicate. We use the same features for all experiments and identify predicates and arguments of all parts of speech. This setup is equivalent to (Kessler and Kuhn, 2013).

We evaluate on each dataset separately using 10-fold cross-validation. We report precision (P), recall (R), and F1-measure (F1). All results are calculated on all predicates and arguments annotated in the data. Bold numbers denote the best result in each column and dataset.

We cannot calculate significance because annotations change between experiments, but we report the absolute differences in F1-measure to the function predicate setting (Δ).

Function predicates vs. content predicates. Table 2 shows the results for predicate identification. We can see that annotating the content word decreases performance in all datasets. This fits our expectation as lexical features have a big weight in the model and by choosing a number of different adjectives over few function words we make the data more sparse. The decrease is quite large compared to the relatively small number of changes we are making.

Table 3 and the first two lines for each dataset in Table 4 show the results of argument identification resp. classification. With system predicates, due to the decreased performance in predicate identification, performance on arguments suffers to a similar degree. With gold (annotated) predicates, performance still suffers for J&L and IMS, but the JDPA datasets are not as much affected or even gain. Part of this is due to the fact that *content predicates* over-generates aspects that are the same token as the predicate even for single word predicates like "faster". Such annotations never occur in the other datasets but are common in the JDPA datasets. The increased recall for aspects balances the loss on the other arguments.

Aspect annotations vs. scale annotations. The second experiment influences only argument classification, compare lines 1 and 3 for every dataset in Table 4. As we introduce more classes, we expect overall performance to drop. Indeed there is a drop, but the difference between the two configurations is small. When we look at the confusion matrices for all datasets, we see that there are nearly no confusions of the scale with an entity and only few of scale and aspect.

We have analyzed some cases where the scale has been confused with the aspect in the IMS data.

[6] http://code.google.com/p/mate-tools/

		with system predicates				with gold predicates			
		P	R	F1	Δ	P	R	F1	Δ
J&L	function predicates	**57.6**	**31.9**	**41.1**		69.4	**46.3**	**55.5**	
	content predicates	56.8	28.1	37.6	-3.5	**69.6**	45.2	54.9	-0.6
J-C	function predicates	**57.4**	**25.7**	**35.5**		**67.9**	37.2	**48.1**	
	content predicates	56.7	24.9	34.6	-0.9	67.6	**37.3**	**48.1**	-0.0
J-A	function predicates	**57.2**	**27.5**	**37.2**		**70.4**	41.7	52.4	
	content predicates	56.8	25.8	35.5	-1.7	**70.4**	**42.1**	**52.7**	+0.3
IMS	function predicates	**70.7**	**44.1**	**54.3**		**78.9**	**57.4**	**66.4**	
	content predicates	70.4	41.9	52.5	-1.8	77.9	56.5	65.5	-0.9

Table 3: Results argument identification.

		with system predicates				with gold predicates			
		P	R	F1	Δ	P	R	F1	Δ
J&L	function predicates	**50.2**	**27.8**	**35.8**		**59.9**	**40.0**	**48.0**	
	content predicates	48.9	24.2	32.4	-3.4	59.7	38.8	47.0	-1.0
	function preds. w. scale	49.6	27.5	35.4	*-0.4*	59.2	39.5	47.4	*-0.6*
J-C	function predicates	**49.5**	**22.2**	**30.7**		**55.5**	**30.4**	**39.3**	
	content predicates	47.0	20.6	28.7	-2.0	54.5	30.1	38.8	-0.5
	function preds. w. scale	49.4	22.1	30.6	*-0.1*	55.2	30.2	39.1	*-0.2*
J-A	function predicates	**43.8**	**21.1**	**28.5**		50.2	29.7	37.3	
	content predicates	**43.8**	20.0	27.4	-1.1	**51.0**	**30.5**	**38.2**	+0.9
	function preds. w. scale	43.3	20.8	28.1	*-0.4*	49.7	29.4	37.0	*-0.3*
IMS	function predicates	**63.0**	**39.3**	**48.4**		**69.0**	**50.2**	**58.1**	
	content predicates	62.4	37.1	46.5	-1.9	67.7	49.1	56.9	-1.2
	function preds. w. scale	62.4	38.9	47.9	*-0.5*	68.4	49.8	57.6	*-0.5*

Table 4: Results argument classification (micro-average over all classes).

Confusions occur mostly with untypical scale arguments like “more feature rich” or “more pro” where the system predicts an aspect because the content word is tagged as a noun. We have also found a few annotation errors where annotators mistakenly annotated an aspect instead of a scale.

6 Conclusions

In this short paper we present experiments on how different annotations of multiword comparative predicates (“more powerful”, “as good as”, …) affect the classification performance of a machine learning system that identifies comparative predicates and arguments. Our experiments indicate that it is more helpful to annotate function words than content words as predicates. In the annotation conventions adopted in standard datasets for sentiment analysis, the modified adjective is annotated as the aspect of the comparison. We discuss problems with this type of annotation and propose the introduction of an additional argument type which solves the problems. In our second experiment we show that there is only a small drop in performance when adding this new argument type. For future work we plan to look more closely at the annotation of other (non-systematic) multiword predicates such as “on par with”.

Acknowledgments

We thank the reviewers for their helpful comments. The work reported in this paper was supported by a Nuance Foundation grant.

References

Anders Björkelund, Love Hafdell, and Pierre Nugues. 2009. Multilingual Semantic Role Labeling. In *Proceedings of CoNLL '09 Shared Task*, pages 43–48.

Joan W. Bresnan. 1973. Syntax of the comparative clause construction in English. *Linguistic Inquiry*, 4(3):275–343.

Murthy Ganapathibhotla and Bing Liu. 2008. Mining opinions in comparative sentences. In *Proceedings of COLING '08*, pages 241–248.

Feng Hou and Guo-hui Li. 2008. Mining Chinese comparative sentences by semantic role labeling. In *Proceedings of ICMLC '08*, pages 2563–2568.

Nitin Jindal and Bing Liu. 2006a. Identifying comparative sentences in text documents. In *Proceedings of SIGIR '06*, pages 244–251.

Nitin Jindal and Bing Liu. 2006b. Mining comparative sentences and relations. In *Proceedings of AAAI '06*, pages 1331–1336.

Wiltrud Kessler and Jonas Kuhn. 2013. Detection of product comparisons - How far does an out-of-the-box semantic role labeling system take you? In *Proceedings of EMNLP '13*, pages 1892–1897.

Wiltrud Kessler and Jonas Kuhn. 2014. A corpus of comparisons in product reviews. In *Proceedings of LREC '14*.

Jason S. Kessler, Miriam Eckert, Lyndsay Clark, and Nicolas Nicolov. 2010. The 2010 ICWSM JDPA Sentiment Corpus for the Automotive Domain. In *Proceedings of ICWSM-DWC '10*.

Arnim von Stechow. 1984. Comparing semantic theories of comparison. *Journal of semantics*, 3:1–77.

Kaiquan Xu, Stephen Shaoyi Liao, Jiexun Li, and Yuxia Song. 2011. Mining comparative opinions from customer reviews for competitive intelligence. *Decis. Support Syst.*, 50(4):743–754, March.

Improving the Performance of Standard Part-of-Speech Taggers for Computer-Mediated Communication

Andrea Horbach, Diana Steffen, Stefan Thater, Manfred Pinkal
Department of Computational Linguistics, Saarland University, Saarbrücken, Germany
(andrea|dsteffen|stth|pinkal)@coli.uni-saarland.de

Abstract

We assess the performance of off-the-shelve POS taggers when applied to two types of Internet texts in German, and investigate easy-to-implement methods to improve tagger performance. Our main findings are that extending a standard training set with small amounts of manually annotated data for Internet texts leads to a substantial improvement of tagger performance, which can be further improved by using a previously proposed method to automatically acquire training data. As a prerequisite for the evaluation, we create a manually annotated corpus of Internet forum and chat texts.

1 Introduction

Around the turn of the century, the Internet made huge amounts of natural-language text easily accessible, and thus enabled a hitherto inconceivable success story of data-driven, statistical methods in computational linguistics. But the Internet also created a new challenge for language processing because it substantially changed the object of investigation. In computer-mediated communication (CMC), a wide variety of new text genres and discourse types such as e-mail, twitter, blogs, and chat rooms have emerged, which differ from standard texts in various ways and to different degrees. Differences include tolerance against typing errors and spelling rules, inclusion of colloquial, spoken-language elements in lexicon, syntax, and style (e.g., contractions like *gibt es* to *gibts*); intended use of non-standard-language components, like systematic "misspelling" and non-standard lexical items (e.g., neologisms or acronyms), to mention just a few. Statistical NLP tools are usually trained on and optimized for standard texts like newspaper articles. Reliable high-performance off-the-shelf tools show a dramatic performance drop, when applied to substantially differing linguistic material. This holds also for basic tasks such as POS tagging, which is particularly detrimental because the basic information is needed for all kinds of more advanced analysis tasks.

In this paper, we report work on POS tagging of two different CMC text types in German. We assess the performance of POS taggers trained on standard newspaper texts when applied to CMC texts and explore easy-to-implement and low-resource methods to adapt these taggers to CMC texts. We test the performance of three state-of-the-art taggers and explore two adaptation methods: First, we generate additional training material from automatically annotated data using a method that has been proposed recently by Kübler and Baucom (2011) for a different domain adaptation task. Second, we use small amounts of manually annotated CMC data as additional training data.

The main result of this paper is that even small amounts of manually annotated CMC training data substantially improve tagger performance on CMC texts; a combination of manually annotated and automatically acquired training data leads to a further improvement of tagger performance to up to 91% on texts from an Internet forum. A further major contribution is the POS-tagged CMC gold standard corpus consisting of about 24 000 tokens, which we created as a prerequisite for our evaluation and which will be made publicly available.

2 Related work

The growing interest in CMC language can be seen from a number of recently established collaborative activities like the scientific network *Empirical Research on Internet-based Communication*[1], the recently launched European network *Building and Annotating CMC Corpora*[2], and the Special Interest Group *Computer-mediated Communication* within the Text Encoding Initiative[3] (TEI).

Specific work for POS-tagging of non-standard texts include work by Ritter et al. (2011), Derczynski et al. (2013), Gimpel et al. (2011) and Owoputi et al. (2013), who report about POS tagsets and optimization of linguistic tools for annotating English Twitter data.

Kübler and Baucom (2011) investigate domain adaptation for POS taggers using the consent of three different taggers on unannotated sentences to create a new training set. They reach a moderate increase in accuracy from 85.8% to 86.1% on dialogue data but are still far below the performance on standard newspaper texts. We adopt their approach of tagger consent as one way of training set expansion in our experiments.

Work for German has been done by Giesbrecht and Evert (2009), who compare the performance of five different statistical POS tagger on different types of Internet texts, showing that the accuracy of approx. 97% on standard newspaper texts drops below 93%s when tagging web corpora. They mostly investigate texts that are close to standard language such as online news texts. Forum texts deviate most from the standard and the performance for forum texts matches our observations. Chat corpora are not covered in their study.

Bartz et al. (2014) suggest an extension of the widely used STTS tagset for POS tagging of web corpora, which we also use.

While our approach tries improves the performance of existing POS taggers on CMC texts, Rehbein (2013) develops a new POS tagger for German twitter data, which is trained using word clusters with features from an automatically created dictionary and out-of-domain training data.

3 Gold standard annotation

This section describes the annotation of computer-mediated discourse with POS information to be used as gold standard data in the experiments reported in Section 4 below.

3.1 Data sources

We select two complementary types of Internet text – forum posts from the Internet cooking community *www.chefkoch.de* and the *Dortmund Chat Corpus* (Beißwenger, 2013) – to cover a range of phenomena characteristic of Internet-based communication.

Forum. We use forum articles from the Internet cooking community *www.chefkoch.de*, which we downloaded in Feb. 2014, resulting in a large corpus of about 500 million tokens. Although the website primarily offers cooking-related services, forum articles address a wide range of everyday life topics and only a minor part of them – less than 1% as indicated by a case study – has the form of actual cooking recipes. In comparison to chats, we expect a higher agreement with standard language.

Chat. We complement the forum dataset with the *Dortmund Chat Corpus*, which is the standard corpus for German chat data; it consists of chat logs of various degrees of formality, ranging from very informal contexts to moderated expert chats. Since the focus of our research are phenomena typical for computer-mediated discourse, we select our gold standard data only from informal chats, which we assume to contain a larger number of interesting CMC phenomena.

3.2 Tagset

CMC data contain some language phenomena that are not properly covered by the standard STTS tagset, such as emoticons, so called “action words” in inflective form (e.g., *rumsitz*), URLs and various kinds of contractions. In order to account for the most frequent of those phenomena we use an extended version of STTS proposed by Bartz et al. (2014) containing additional tags for these categories.

We add two tags to capture errors made by the writers unaware of German spelling rules. ERRAW is assigned when a token should be part of the following token, i.e. if the writer inserted an erroneous whitespace; ERRTOK is a tag for the opposite case when the writer joined two words

[1] http://www.empirikom.net/

[2] https://wiki.itmc.tu-dortmund.de/cmc/

[3] http://www.tei-c.org/Activities/SIG/CMC/

tag	description	example	freq. forum	freq. chat
VVPPER	full verb + personal pronoun	versuchs, gehts, gibbet, kuckste	0.10	0.26
VMPPER	modal + personal pronoun	kanns, willste	0.02	0.05
VAPPER	auxiliary + personal pronoun	isses, hassu, wirste	0.06	0.13
KOUSPPER	conjunction + personal pronoun	wenns	0.01	0.00
PPERPPER	2 personal pronouns	[wenn] ses [frisst]	0.01	0.01
ADVART	adverb + article	son, sone	0.00	0.03
ADR		@nudelsupperstern, Sebastian	0.38	2.20
URL		www.uni-hildesheim.de	0.00	0.05
ONO	onomatopoeia	hehe, Mmmmmm	0.02	0.50
EMO	emoticons	:-), ⟨img src="smileys/wink.gif"⟩	1.72	1.40
AW	a verb in inflective form	ächz, rumsitz, knuddel	0.15	2.30
AWIND*	marks AW boundaries	*	0.24	4.01
ERRAW*	incorrectly separated word	[meine Kinder da] anzu [melden]	0.20	0.11
ERRTOK*	tokenization error	gehtso, garnicht	0.07	0.15
all new tags			3.02	11.18
all standard STTS tags			97.98	88.82

Table 1: Additional STTS tags, descriptions, examples and tag frequencies (%) in the goldstandard corpora. A * marks those tags that were not included in the extension by (Bartz et al., 2014)

that should be separated. Table 1 shows all non-standard tags we use together with examples.

3.3 Annotation

We manually annotated 11 658 tokens from the *Dortmund Chat Corpus* and 12 335 tokens from randomly chosen posts from the *chefkoch* corpus with POS information. Prior to annotation, the data has been automatically tokenized. The tokenizer sometimes tears apart strings that should form one token, such as several subsequent punctuation marks (e.g., *!!!*) or ASCII emoticons. Those systematic errors have been cleaned up manually. To simplify the annotation process, we also corrected few tokenisation errors made by the user in cases where it was an obvious typing error; for instance, *wennman* was corrected to *wenn man*.

Each file in both subcorpora has been annotated by two annotators. For the forum subcorpus, annotators were able to see the first post in the respective thread in order to provide them with potentially helpful context. For the chat data, they annotated continuous portions of approx. 550 tokens of chat conversations.

Annotators were asked to ignore token-level errors like typos or grammatical errors whenever possible, i.e. to annotate as if the error was not there. For instance, when the conjunction *dass* was erroneously written *das*, they should annotate KOUS even though *das* as a correct form can only occur as ART, PRELS or PDS.

After the annotations, annotators were shown where their annotation differed from the one of their co-annotator (without showing them the other annotation) in order to self-correct obvious mistakes. Cases of disagreement after that initial error correction have been resolved by a third annotator. The pairwise inter-annotator agreement (κ coefficient) ranges between 0.92 and 0.95 after the initial annotation and between 0.96 and 0.97 after self-correction.

Split into Training and Test Data. For our experiments in the next section, we split the gold standard into one third that is used as additional training material and two thirds for testing, making sure that equal portions of the *chat* and *forum* datasets are used in the resulting test and training dataset.

3.4 Corpus Analysis

The two subcorpora vary considerably not only in general linguistic properties like average sentence length (10.5 tokens for *forum*, 5.9 for *chat*) but even more so in the frequency with which POS tags, especially the non-standard tags occur. Table 1 shows the relative frequency of the new tags in both corpora. These numbers confirm our initial hypothesis about the degree of deviation from the standard in the two subcorpora: While the *forum* data only contain 3% of nonstandard tags, *chat* contains 11.2% of those new tags, thus clearly calling for adapted processing tools. 78.3% of all sentences in *forum* do not contain any non-standard tag, while in *chat* only 60.0% of all sentences are covered by the traditional STTS tagset.

4 Experiments

This section compares and combines two ways to re-train statistical POS taggers to improve their performance on CMC texts: (a) We extend a standard newspaper-based training corpus with data drawn from automatically tagged CMC texts applying a technique proposed by Kübler and Baucom (2011). (b) We extend the training corpus with small portions of manually annotated CMC texts. Results show that while the first approach leads to minor improvements of tagger performance, it is outperformed by a large margin by the second approach – even if only very few additional training sentences are added to the training corpus. A small further improvement can be obtained by combining the two approaches.

4.1 Methods

The key idea behind the approach of Kübler and Baucom (2011) is to parse raw text using different taggers, and to extend the training data for the taggers with automatically annotated sentences for which all taggers produce identical results. In our experiment, we use the following three taggers: TreeTagger (Schmid, 1994), Stanford Tagger (Toutanova et al., 2003) and TnT (Brants, 2000).

Baseline training corpus. As a starting point for our re-training experiments, we train our taggers using the Tiger corpus (Brants et al., 2004), which is a widely used German newspaper corpus providing POS annotations for roughly 900 000 tokens (50 000 sentences). The Tiger corpus consists of 20-year-old newspaper articles using the old German orthography. Since many words in our datasets are written according to the new spelling rules introduced in 1996, we automatically convert the original Tiger corpus to the new German orthography using *Corrigo* (Kurzidim, 2004) and replace approx. 11 000 tokens (1.2%) by their new spelling. We combine both variants of the corpus (original and converted) into a single new training corpus, referred to as “Tiger New” (*tn*) below.

Experiment 1: Corpus expansion by using multiple taggers. We apply each of the three taggers to the complete *Chefkoch* and *Dortmund Chat* datasets, resulting in an annotated corpus consisting of around 36 000 000 sentences.[4] For around 2 700 000 sentences ($<$ 8%) all three taggers agree completely. From those sentences we randomly select 50 000 sentences (561 000 tokens) from *Chefkoch* and 10 000 sentences (102 000 tokens) from *Dortmund Chat* and add them to our baseline corpus; we refer to the resulting training corpus as *tn+auto*.

Experiment 2: Adding manually annotated CMC data. In a second experiment, we use one third of the annotated gold standard data (around 7 800 tokens) as additional training material. Because this added data amounts to less than 1% of the number of tokens in the Tiger New corpus, we boost it by adding it several times, arbitrarily setting the boosting factor to 5 (*tn+gold*).

Experiment 3: Combining the two methods. In a third experiment, we combine the two approaches and generate a second set of automatically created gold-standard sentences by randomly selecting new training sentences automatically tagged with the *tn+gold* models (of the same amount as before). We call this dataset *tn+auto2*. The full dataset (*tn+gold+auto2*) consists of the Tiger corpus extended by gold standard data and additional automatically tagged data, tagged with the help of the same gold-standard data.

4.2 Results

The left part (“all sentences”) of Table 2 shows the performance of the three taggers using different training datasets. Unsurprisingly, the original Tiger model (*tn*) performs very poorly when applied to non-standard CMC texts. Adding automatically annotated new training data (*tn+auto*) gives us a moderate and consistent positive effect across all corpora and taggers, improving tagger performance on average by 1.3% on the “All” test set. A much larger gain in performance can be obtained

[4] In order to avoid problems resulting from different tokenizations of the input texts when tagger results are compared (see below), we do not use the built-in tokenizers of the three taggers but use Stefanie Dipper’s tokenizer (http://www.linguistics.ruhr-uni-bochum.de/˜dipper/token izer.html) for all three taggers.

		all sentences			standard sentences only		
Tagger	trained on	Chat	Forum	*All*	Chat	Forum	*All*
TreeTagger	Tiger new (tn)	0.714	0.845	*0.784*	0.800	0.874	*0.842*
	+auto	0.727	0.855	*0.796*	0.816	0.885	*0.854*
	+gold	0.826	0.881	*0.855*	0.861	0.909	*0.888*
	+gold+auto2	**0.835**	0.888	*0.863*	**0.873**	0.917	**0.898**
Stanford	tn	0.702	0.840	*0.776*	0.789	0.869	*0.834*
	+auto	0.715	0.851	*0.788*	0.803	0.880	*0.847*
	+gold	0.816	0.897	*0.860*	0.849	0.910	*0.884*
	+gold+auto2	0.826	0.903	*0.867*	0.863	0.918	*0.894*
TnT	tn	0.691	0.846	*0.774*	0.777	0.876	*0.832*
	+auto	0.708	0.857	*0.788*	0.796	0.889	*0.848*
	+gold	0.827	0.906	*0.870*	0.852	0.918	*0.889*
	+gold+auto2	**0.835**	**0.912**	**0.877**	0.863	**0.923**	*0.897*

Table 2: Accuracy of various models on both gold standard datasets, evaluated on the complete test set (*all sentences*) and on the subset that contains only sentences with tags from the original STTS (*standard only*). All differences in model performance are pairwise statistically significant (for each tagger and sub-corpus) according to a McNemar test ($p < 0.005$).

by adding small amounts of manually annotated CMC data (*tn+gold*); the performance gain is especially large for the *chat* subcorpus where it leads to an improvement of 13.4% for the best-performing TnT tagger, compared to the baseline. For forum data with a higher degree of standard language the improvement is less pronounced but still much larger compared to the *tn+auto* models. Adding both gold-standard data and automatically tagged data (*auto2*) leads to the best performing models with an accuracy of up to 91% (TnT) on forum data. We also tried to combine *auto* with *gold*, but found no positive effect.

Standard tags. The poor performance of the original tagger models and the large performance improvement obtained by adding additional training data from the gold standard is to some extent unsurprising, since the test data contains many tokens annotated with new POS tags which the original taggers cannot predict. We should note, however, that the performance gain cannot be explained by new POS tags only: The right part of Table 2 shows the performance of the taggers when applied to sentences from the gold standard in which new POS tags are not used. The performance of the original taggers is still quite low on this test set (between 83% and 84%) and is improved to 90% (TreeTagger) by using additional training data.

New tags. We also investigated the performance of the three taggers wrt. those words in the gold standard that received a new POS tag from the STTS extension by our overall best-performing model. TreeTagger achieves only 42% accuracy on such words, while Stanford Tagger and TnT achieve 58% and 67%, respectively. The low results are not surprising, given the small amount of training data. Stanford and TnT perform better than TreeTagger since they are able to generalize to unseen words, while TreeTagger assigns new tags only to known words and obviously needs larger amounts of training data to adapt to new texts or tags.

Performance on unknown words. The three taggers also show different behavior when evaluated only on unknown lexical material, i.e. words that do not occur in the training data. The best-performing model (*tn+gold+auto2*) for each tagger reaches performances of 41% (Stanford), 49% (TreeTagger) and 74% (TnT), showing again that TreeTagger and to some extent the Stanford Tagger seem to rely much more than TnT on lexical information.

Performance on specific new classes. Additionally we looked at the individual performance wrt. the new tags, for the best-performing models for all three taggers, and observe wide variation both across taggers and POS tags. Infrequent tags, especially the rare contractions are generally not learned well. Some tags with higher frequencies are learned with F-Scores higher than 0.95: EMO and AWIND for TnT, while TreeTagger (0.44) and Stanford (0.87) perform worse for EMO. Unsurprisingly AWIND (almost always a *) is learned well by all taggers. ADRs, although frequent, seem to be generally hard: the best-performing TnT tagger reaches an F-score of 0.18.

If we consider only unknown words within new tags we see a similar picture as in the general analysis of unknown words: While TnT can assign the new tags to the frequent classes (ADR, AW, EMO) although with some performance loss, Stanford and TreeTagger only successfully recognize some instances of unknown ADR, AW and EMO (but all with very low recall rates).

We also experimented with simple hand-crafted pattern matching rules to extend the accuracy for the most frequent new tags, e.g. tagging all words containing an @ in the beginning as ADR. However as the @ is left out in many ADRs and the syntactically integrated ADRs are tagged in the gold-standard as NE, we could not improve the performance by such additional rules. This shows again, that tagging of those new STTS categories is not a simple task and dependent from both word information and distribution.

4.3 Varying the amount of gold-standard data.

One potential disadvantage of using manually annotated gold-standard data to (re-)train taggers is that annotation is time-consuming and expensive. We should stress, however, that even a very small amount of manually annotated training data leads to a large improvement of tagger performance: We split the training part of the gold-standard into three equal parts and train models on corpora where we add (boosted 5 times) one part (*gold1*), two parts (*gold2*) and all three parts (*gold3*) to the training set. The results are presented in figure 1 exemplarily for the TnT tagger. We see that already a very limited time investment – around 20 hours of work for double annotations of approx. 2 600 tokens – leads to a vital improvement of tagging performance and adding more gold data improves the performance further, but not to the same extent.

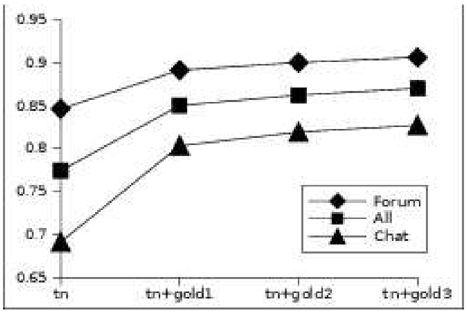


Figure 1: Accuracy of TnT when adding different amounts of gold standard data to the training data

5 Conclusions and Future Work

We have tested the performance of three state-of-the-art POS taggers and explored two low-resource and easy-to-implement adaptation methods to increase tagger performance on computer mediated communication (CMC) texts. A previously proposed method of using automatically annotated data to extend the training set leads to small improvement of tagger performance. A much higher improvement of tagger performance can be obtained by using small amounts of manually annotated CMC data as additional training data. A further improvement can be obtained by combining the two approaches, leading to up to 91% tagger performance on internet forum texts.

In future work, we will investigate the effects of training on a particular genre instead of CMC texts in general: While both forum and chat data deviate from standard texts, they each have their own particularities the taggers have to account for. The token *g* for example is used in in the *chefkoch* forum almost exclusively as abbreviation for *Gramm* (*gram*), whereas in chat corpora it usually indicates an action word as in **g** standing for *grin*.

We will also explore the effects that the choice of the tagging algorithm has and how the taggers can be used in a way that combines their individual strengths better.

Acknowledgements. This work is part of the BMBF-funded project “Analyse und Instrumentarien zur Beobachtung des Schreibgebrauchs im Deutschen.” We would like to thank our annotators Ali Abbas, Jana Ott, Jakob Prange and Maximilian Wolf. We also thank Dietrich Klakow for support concerning the forum dataset.

References

Thomas Bartz, Michael Beißwenger, and Angelika Storrer. 2014. Optimierung des Stuttgart-Tübingen-Tagset für die linguistische Annotation von Korpora zur internet-basierten Kommunikation: Phänomene, Herausforderungen, Erweiterungsvorschläge. *Zeitschrift für germanistische Linguistik*, 28(1):157–198.

Michael Beißwenger. 2013. Das Dortmunder Chatkorpus. *Zeitschrift für germanistische Linguistik*, 41(1):161–164.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen, Esther Knig, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. Tiger: Linguistic interpretation of a german corpus. *Journal of Language and Computation, Special Issue*, 2(4):597–620.

Thorsten Brants. 2000. TnT – a statistical part-of-speech tagger. In *Proceedings of the Sixth Conference on Applied Natural Language Processing*, pages 224–231, Seattle, Washington, USA, April. Association for Computational Linguistics.

Leon Derczynski, Alan Ritter, Sam Clark, and Kalina Bontcheva. 2013. Twitter part-of-speech tagging for all: Overcoming sparse and noisy data. In *Proceedings of the International Conference on Recent Advances in Natural Language Processing*.

Eugenie Giesbrecht and Stefan Evert. 2009. Is part-of-speech tagging a solved task? an evaluation of pos taggers for the german web as corpus. In *Proceedings of the Fifth Web as Corpus Workshop*, pages 27–35.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A Smith. 2011. Part-of-speech tagging for twitter: Annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: short papers-Volume 2*, pages 42–47. Association for Computational Linguistics.

Sandra Kübler and Eric Baucom. 2011. Fast domain adaptation for part of speech tagging for dialogues. In Galia Angelova, Kalina Bontcheva, Ruslan Mitkov, and Nicolas Nicolov, editors, *RANLP*, pages 41–48. RANLP 2011 Organising Committee.

Michael Kurzidim. 2004. Fehlerpolizei - Wie gut sind Rechtschreibkorrektur-Programme? *c't*, (2):110–117.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT*, pages 380–390.

Ines Rehbein. 2013. Fine-grained pos tagging of german tweets. In *Language Processing and Knowledge in the Web*, pages 162–175. Springer.

Alan Ritter, Sam Clark, Oren Etzioni, et al. 2011. Named entity recognition in tweets: an experimental study. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 1524–1534. Association for Computational Linguistics.

Helmut Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of International Conference on New Methods in Language Processing*, Manchester, UK.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of the 2003 Human Language Technology Conference of the North American Chapter of the Association for Computational Linguistics (HLT-NAACL 2003)*, pages 252–259, Edmonton, Canada.

Unsupervised Text Segmentation for Automated Error Reduction

Lenz Furrer
University of Zurich
Binzmühlestr. 14, CH-8050 Zürich
lenz.furrer@uzh.ch

Abstract

Challenging the assumption that traditional whitespace/punctuation-based tokenisation is the best solution for any NLP application, I propose an alternative approach to segmenting text into processable units. The proposed approach is nearly knowledge-free, in that it does not rely on language-dependent, man-made resources. The text segmentation approach is applied to the task of automated error reduction in texts with high noise. The results are compared to conventional tokenisation.

1 Introduction

Dividing written text into small units is one of the most basic and fundamental steps in Natural Language Processing (NLP). Generally, this task does not attract much attention, as it is most often carried out by relying on a language's orthography for marking word boundaries. Whitespace/punctuation-based tokenisation – which is applicable to most mainstream languages covered in NLP literature – is not necessarily the optimal starting point for every NLP application. In this work, I investigate the use of an alternative segmentation approach by applying it to the task of automatically reducing errors in documents of amendable text quality.

The experiments reported in this work were carried out with electronic documents created by Optical Character Recognition (OCR) software. Modern OCR tools provide a high recognition accuracy. However, it is often desirable to improve the text quality of OCR-generated documents in a post-processing step, especially if they are in a challenging format such as historical books.

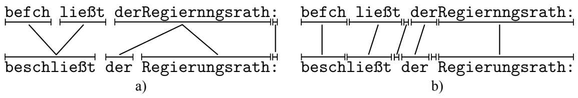

befch ließt derRegiernngsrath:
beschließt der Regierungsrath:
a)
befch ließt derRegiernngsrath:
beschließt der Regierungsrath:
b)

Figure 1: A line of OCRed text (upper line) and its correction (lower line), segmented with traditional, whitespace-based tokenisation (a) and with a non-standard segmentation method (b). Note that the space character is often, but not always, treated as a separate token in (b). The line can approximately be translated as 'the executive council decides'.

Most post-correction attempts operate on the level of words, i. e. tokenised text. However, recognition errors can lead to erroneous tokenisation if word separators are omitted or falsely inserted, as shown in figure 1. Segmentation errors need special attention in word-based correction approaches. As illustrated in example (a), tokenisation is misled by missing or falsely inserted space characters, producing disrupted and run-on tokens which cannot be corrected by comparing words one by one. An attempt to extensively correct segmentation errors, thus, massively increases the search space for correcting. If, however, tokenisation was not carried out solely on the basis of how a token is delimited, but the internal structure of a token was taken into account instead, one might expect a tokenisation scheme that is less sensitive to segmentation errors produced by the recognition system. Example (b) shows how a more fine-grained tokenisation can simplify the correction procedure, in that the search space is reduced. The resulting segments do not necessarily correspond to any linguistic categories; they might, however, help localise and fix segmentation errors. In the example given, all errors can be addressed within the scope of a single token, no 1:n correspondences have to be considered.

2 Related Work

2.1 Automated OCR Error Correction

The field of automated OCR error correction has gained attention over the last years, keeping up with a growing number of large-scale digitisation projects.[1] Most of the work is concerned with cleaning up the output of an off-the-shelf OCR system, i. e. post-processing, although there are attempts at tweaking the performance of the system itself, such as Heliński et al. (2012). While purely dictionary-based correction attempts for OCR errors have not been very popular lately, there is still, work on efficient dictionary lookup, designed for the use with OCRed search terms and similar scenarios (Mihov and Schulz, 2004; Mihov et al., 2007). Lund and Ringger (2009) and Volk et al. (2010) achieve text improvements by combining the output of multiple OCR systems, following the idea that different recognition techniques lead to different errors.

Corpus-based post-correction of OCR errors has found more and more proponents in the past, thus according with a general trend in many NLP areas. Among other advantages, corpus-based correction – as opposed to a dictionary-driven approach – enables exploitation of context information, which allows for addressing *real-word errors*, i. e. OCR misrecognitions that result in another existing word (e. g. *Negierung* 'negation' instead of *Regierung* 'government'), as opposed to *non-word errors*, which are misspelt words (e. g. *Rcgiernng*). As an early example, Tong and Evans (1996) investigate the use of a bigram language model for correcting OCR errors. The corrections are performed on word level, using a lexicon derived from a training set of error-free texts. The authors tackle both non-word and real-word errors and report error reduction rates up to 60 % for plain alphabetic tokens. Bassil and Alwani (2012) perform corpus-based corrections with the Google Web 1T 5-Gram Data Set. Their approach performs very well on noisy OCRed text, although the small size of the test set (less than 300 running tokens) gives it only limited evidence. More of a bootstrapping approach is followed by Reynaert (2008). He pursues the idea that OCR errors are unsystematic noise that can be filtered out without the use of clean text.

Some work also addresses segmentation errors. For example, Reynaert (2004) builds a corpus-derived lexicon containing word bigrams, wich enables a chance of correcting run-on tokens. Interestingly, he later states that he "do[es] not here attempt to resolve run-ons" (Reynaert, 2006, p. 90). Kolak et al. (2003) explicitly tackle both kinds of segmentation errors by allowing splits in the lexicon words and the OCR tokens.

2.2 Unsupervised Text Segmentation

Most NLP tasks operate at the level of words. The task of splitting text into words needs some attention in the case of continuous sequences, as with speech recognition or in the case of orthographies lacking word boundaries such as Chinese, see e. g. Chung and Gildea (2009), or when dealing with phonemical transcriptions (Goldwater et al., 2006). It is usually referred to as *word segmentation*. In contrast to this, *tokenisation* is the task of achieving the same goal for texts containing word dividers (mostly blank spaces). Although tokenisation seems to be quite straightforward a task, there are still innovations in the field, like the proposal by Barrett and Weber-Jahnke (2011), who aim at performing tokenisation and part-of-speech tagging simultaneously.

Tokenisation may also be difficult in the case of untrusted input. Wrenn et al. (2007) attempt the segmentation of texts produced by clinicians, which have a high spelling-error rate (including segmentation errors), causing troubles to a standard tokeniser. The authors introduce word boundaries using a technique borrowed from unsupervised morphological segmentation,[2] which was originally applied to single words and short

[1] See e. g. Holley (2009) for a large Australian newspaper digitisation program, Jisc (`http://www.jisc.ac.uk/`) for a list of ongoing projects on the British Isles, or the IMPACT initiative (Tumulla, 2008).

[2] See the comprehensive work by Hammarström and Borin (2011) for an overview of the field.

		1		1		2		1		1		2		20		5		13		25		←
	d		i		s		t		u		r		b		a		n		c		e	
→		15		24		24		8		2		2		4		2		1		1		

Figure 2: LPV (first row) and LSV (last) counts for the word *disturbance*, based on a list of dictionary entries (Harris, 1967, p. 69). The general tendency of decreasing numbers is interrupted by "peaks" at positions with higher variability, e. g. between *disturb* and *ance* both in left-to-right and right-to-left reading.

utterances. Wrenn et al. adapt the method to work with running text of arbitrary length.

Golcher (2006) proposes a comprehensive approach at segmenting text in an unsupervised manner, addressing text segmentation, morphological decomposition, multiword unit detection, and compound analysis at the same time. He uses a combination of different statistical measures to segment continuous text into useful units. Unfortunately, the author did not perform an extrinsic evaluation, such as applying the segmented text to an information retrieval or machine translation system, by which means the advantages of the proposed segmentation could have been shown.

3 Methods

In a series of experiments in this and earlier work (Furrer, 2013), I examined the use of an alternative text segmentation scheme, as opposed to traditional tokenisation. The effects of the segmentation are measured by the performance of an automated error correction system.

3.1 Text Segmentation

Before being processed by the correction module, all text needs to be segmented into basic units. The text segmentation method presented here induces segment boundaries from the distribution of characters. It is based on the work of Wrenn et al. (2007) and goes back to the *letter successor variety* method (LSV), which was introduced by Harris (1955) and given its name by Hafer and Weiss (1974). The intuition behind LSV is that morpheme boundaries can be inferred from statistical properties of the characters found in a list of words or short phrases. For a set of words that share a common prefix x, LSV(x) is defined as the number of distinct characters that succeed x in these words. In figure 2, the bottom row lists LSV counts for every character transition in a test word, based on counts for all entries of an English dictionary. For example, if a word begins with the letter *d*, the second letter can be chosen from 15 possibilities, one of which is *i*. In all words beginning with *distu*, the sixth position is occupied by one of 2 distinct letters. Analogously, LSV can be counted backwards, i. e. counting the distinct letters preceding a shared suffix (see the top row in figure 2). This is called *letter predecessor variety* (LPV). In order to manage continuous text of arbitrary length, the context for computing LSV/LPV is limited to a *window* of k characters (Markov assumption). The windowed subsequences are stored in a character trie as is illustrated in figure 3.

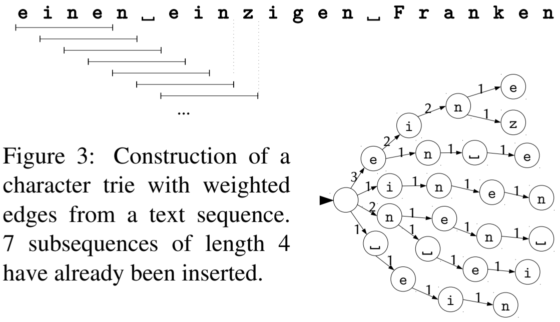


Figure 3: Construction of a character trie with weighted edges from a text sequence. 7 subsequences of length 4 have already been inserted.

During segmentation, the entire text collection is read twice: In a learning step, the character distribution is gathered from all texts and stored in two character tries – one for the forward, and one for the backward reading. Subsequently, this global information is used to find good split points locally. Given an input sequence s, a pair of character tries, and a minimal peak threshold, the segmentation routine splits s *exhaustively* into adjacent subsequences. Whitespace characters are thus not removed, but retained in the segments.

For every character transition i in s, a fragility score f is computed on the basis of LSV. The algorithm aims at finding peaks in the sequence of LSV values, i. e. values that are greater than their immediate neighbours. If there is a peak at transition i, then the LSV drop to both its neighbours is summed:

$$f_s(i) = \begin{cases} \Delta_s^<(i) + \Delta_s^>(i) \\ \qquad \text{if } \Delta_s^<(i) > 0 \wedge \Delta_s^>(i) > 0 \\ 0 \quad \text{otherwise} \end{cases} \tag{1}$$

where $\Delta^<$ and $\Delta^>$ are the increasing and decreasing side of the peak, respectively. For a character window of length k, the definitions of $\Delta^<$ and $\Delta^>$

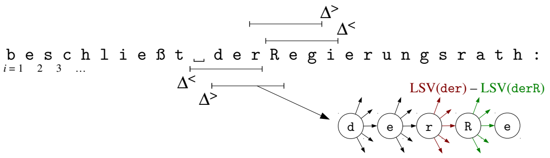


Figure 4: Illustration of a segmentation example.

are narrowed down to the following:

$$\begin{aligned}\Delta^{<}_{s,k}(i) &= \mathrm{LSV}(s^{i}_{i-k+2}) - \mathrm{LSV}(s^{i-1}_{i-k+2})\\ \Delta^{>}_{s,k}(i) &= \mathrm{LSV}(s^{i}_{i-k+3}) - \mathrm{LSV}(s^{i+1}_{i-k+3})\end{aligned} \tag{2}$$

By looking separately at each side of i, $\Delta^{<}$ and $\Delta^{>}$ capture a maximum of context each. For example, with a window size $k = 5$ and $s = \texttt{beschließt␣derRegierungsrath :}$, $\Delta^{<}_{s}(i = 14)$ is calculated using the following subsequences (see also figure 4):

$$\begin{aligned}\Delta^{<}_{s,5}(14) &= \mathrm{LSV}(\texttt{␣der}) - \mathrm{LSV}(\texttt{␣de})\\ \Delta^{>}_{s,5}(14) &= \mathrm{LSV}(\texttt{der}) - \mathrm{LSV}(\texttt{derR})\end{aligned} \tag{3}$$

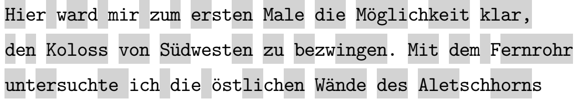
Analogously, the fragility score is computed for the backward reading of s. If the summed scores at transition i reach or exceed the peak threshold, a segment boundary is inserted. Figure 5 shows an example of segmented text.

3.2 Error Correction

I modelled the automated error correction framework closely after the Hidden Markov Model (HMM) proposed by Tong and Evans (1996), which is theoretically well founded and easily adaptable to new data. By realising the digitisation chain as a Markov model, one assumes that the original text can be cleaned from the transmission noise by means of statistical properties.

In the experiments by Tong and Evans, these properties are estimated from a collection of clean texts. The observations are the output produced by the OCR system, cut into processable units (henceforth *segments*) in the preceding segmentation routine. The hidden states are the correct segments that have to be predicted by the correction system. The correction is performed by finding the best path through all possible hidden states, which is done with the Viterbi algorithm.

An important thing to note is that the distinction between error *detection* and error *correction* is absent from this approach. In a HMM, correcting a text segment means predicting a hidden label that looks different from the observed token; not correcting – which is the most frequent operation – means predicting a label that happens to equal the observation. This behaviour is controlled by the emission probability.



```
Hier ward mir zum ersten Male die Möglichkeit klar,
den Koloss von Südwesten zu bezwingen. Mit dem Fernrohr
untersuchte ich die östlichen Wände des Aletschhorns
```

Figure 5: Example of segmented text. Segment boundaries are indicated by change of shading. The character distribution properties were learnt with a window size of 5 and the segmentations were carried out with a peak threshold of 10.

The best sequence of correct segments W^{best} is determined by the conditional probability $P(W|O)$. Using Bayes' theorem, this is equivalent to:

$$W^{\text{best}} = \arg\max_{W}(P(W) \times P(O|W)) \tag{4}$$

$P(W)$ and $P(O|W)$ are the products of the transition and emission probabilities, respectively, for a sequence of observations O and all possible correction sequences W. $P(W)$ is estimated with a statistical language model using the tool `KenLM` (Heafield, 2011), which implements *modified Kneser-Ney* smoothing (Kneser and Ney, 1995; Chen and Goodman, 1998). The language model was trained on a normalised version of the text collection.[3] $P(O|W)$ is estimated from the frequencies of observed human corrections by maximum likelihood estmation (MLE). It expresses the distribution of noise that has been introduced by the OCR process. In order to reduce the search space for potential corrections, I used `simstring` (Okazaki and Tsujii, 2010) for fast retrieval of candidate segments with a minimal cosine similarity of 0.6.

The probability of an OCRed segment o being produced from a true segment w is based on the *minimum edit distance* between the two segments. This means that the cost of transforming w into o is described in terms of edit operations, i. e. inserting, deleting, or substituting a character. By finding the minimal set of edit operations, the characters of w and o are aligned. Based on the confusion probability of each character alignment pair, $P(o|w)$ can be estimated:

$$P(o|w) = \prod_{i,j \in \mathcal{A}_{w,o}} P_{\text{conf}}(o_i|w_j) \tag{5}$$

[3] All letters were converted to lower-case and sequences of digits were replaced by '0'. Please note that this normalisation is only applied when estimating and querying the transition probability, but not for the emission probability.

where $o_1, \ldots, o_n$ and $w_1, \ldots, w_m$ are the characters in o and w, respectively, $\mathcal{A} \subseteq \{0, \ldots, n\} \times \{0, \ldots, m\}$ is the set of alignments between o and w, and $P_{\text{conf}}(x|y)$ is the confusion probability of the aligned characters x and y, estimated with MLE. In order to account for unseen character pairs, the estimations are smoothed with a discount factor α in the range $[0, 1]$. If there is no empirical estimation for a character pair o_i, w_j, the following back-off model is applied:

$$P_{\text{backoff}}(o_i|w_j) = \begin{cases} \alpha & \text{if } o_i = w_j \\ \frac{1-\alpha}{|A|-1} & \text{otherwise} \end{cases} \qquad (6)$$

where $|A|$ is the size of the training data alphabet.

4 Evaluation

In the present work, the outcome of "unorthodox" text segmentation is investigated by measuring its impact on a subsequent NLP application, namely the automatic correction of OCR errors. In an extrinsic evaluation like this, the usefulness of the intermediate step – the segmented text – is only determined by the improvements of the complete system. It does not matter if the resulting segments do or do not agree with human annotations, correlate with usual tokenisation, or correspond to linguistically approved entities.

For the present experiments, I worked with a collection of alpine texts from the 19th century. The text collection consists of 35 volumes of the yearly publications of the Swiss Alpine Club. The books were digitised using OCR in the *Text+Berg digital* project,[4] located at the University of Zurich. Most of the texts are written in German (approximately 90 % of the sentences) and French (10 %), with some portions in English and Italian (0.1 % each). They were published between 1864 and 1899 and sum up to a total of more than 21 000 pages with approximately 560 000 word tokens. The text quality in terms of OCR accuracy is acceptable, but far from perfect. The OCR process was challenged by many factors, such as historical spelling, a high rate of special vocabulary (place names, scientific terms, regional language variations), mixed paper quality, and complex layout (tables, equations).

The text versions used in this work are taken from an offspring project called *SAC-Kokos*,[5] which runs an online platform for publishing and improving the OCRed texts. The project is built on the idea of crowd-sourcing: users may read the texts online and edit them in an easy click-and-type manner whenever an error is encountered. Over the last months, the text quality has been considerably improved by enthusiasts who read through the texts and correct OCR errors.

Throughout the experiments, I used two snapshots of the text collection: one taken at an early stage of the project, when the texts still were close to the raw output of the OCR process (henceforth "*oldest*"), and a very recent one, reflecting many improvements by human editing ("*newest*"). These two versions of the same text collection are used to measure how well the correction system is able to imitate human text correction.

4.1 Evaluation Setup

For creating a test set, I collected a subset of pages with a minimal length of 100 words. I further excluded pages with a character edit ratio below 0.2 % or above 2 %, as it is very likely that pages with a very low change rate have not been thoroughly reviewed yet, while pages with too many edits probably reflect problems outside the reach of an automated correction attempt, such as rearrangements of longer text parts. From this subset, which comprises about 60 % of all pages, I sampled 1000 pages as a test set, and another 1000 pages as a tuning set. The remainder of the (unfiltered) collection was used for training.

The evaluation setup is modelled towards a realistic scenario, where a collection of noisy texts is available, but only a limited amount of clean data. Thus, I used the *oldest* version of the data for training the transition probabilities of the correction HMM. The emission probabilities were estimated from the differences in the *newest* and *oldest* versions of the tuning set pages.

Both training and test data were segmented with the described method. Based on experience with prior experiments, I set the character-trie window and the peak threshold to values of 5 and 10, respectively. Additionally, I ran a separate instance of this experiment with tokenised data, created with a simple regular-expression based tokeniser. For each of the two instances, I carried out multiple runs by varying the value of the discounting

[4] http://textberg.ch/

[5] http://kokos.cl.uzh.ch/

	segmented		tokenised	
	$\alpha = .97$	$\alpha = .9$	$\alpha = .97$	$\alpha = .9$
Δ_E	-16	-12	-11	-12
mod.	30	35	13	15
TP	23	23.5	12	13.5
FP	7	11.5	1	1.5
Prec.	76.67 %	67.14 %	92.31 %	90.00 %
Rec.	0.21 %	0.21 %	0.11 %	0.12 %

Table 1: Error reduction for segmented and tokenised data.

parameter α, which affects the emission probability of the HMM.

After estimating the HMM weights from the training data, the correction system processed the *oldest* version of the test data. Using the *ISRI OCR-Evaluation Frontiers Toolkit* (Rice, 1996), I measured the OCR quality before and after applying the corrections, by assuming that the *newest* test data version is a reasonable approximation of ground truth data.

4.2 Results

Evaluation results are given in table 1. Each column represents a different experimental configuration. The first row (Δ_E) shows the respective error reduction rate (in characters). The total number of modifications made by the system is given in the second row (number of modified segments/tokens, but usually only one character is affected). The following rows give figures for true and false positives (TP, FP) as well as precision and recall. A modification by the system was counted as TP if it reduced the error rate, but as FP if it performed an over-correction (i. e. introduced a new error, at least from the perspective of the *newest* data). In some cases, the system spotted an error correctly, but failed at correcting it (e. g. when deleting a misrecognised character rather than replacing it), which is a neutral modification with respect to the error rate; I counted these cases as half TP, half FP. The recall figures are based on the total number of character errors in the test set, which is 11 100.

5 Discussion

All systems show a moderate error reduction. However, the 60.2 % error reduction reported by Tong and Evans (1996, p. 96) cannot be reproduced, for the systems are very conservative (one would like to say “shy”). The reason for the low recall seems not to lie in the non-standard segmentation, since the tokenisation-based systems are just as cautious. Rather, it seems that the correction model does not adapt well to the present-day requirements of error correction. In fact, one of the key differences between the same task in the 1990s and in the 2010s is the initial accuracy of the noisy documents: While the overall word-error rate in Tong and Evans’ texts is 22.9 %, it is as low as 1.7 % in my test data.[6] This means that the task of finding these well-hidden errors is now harder by an order of magnitude.

Comparing the results of the different systems, the segmentation-based approach generates more TP and leads to a greater error reduction, while tokenisation yields a higher precision. The effect of the discounting factor α is small. It seems to make the systems slightly more audacious when reduced (which gives more probability mass to unseen character substitutions); this comes at a cost in precision, however.

A qualitative analysis of the data by inspecting the performed modifications exemplifies the advantages of the unsupervised segmentation approach. Besides generally good corrections like *Glär-nisch* → *Glärnisch* (a mountain) or *HUtten* → *Hütten* (‘huts’), which could also be detected in a word-based correction attempt, there are cases that clearly profit from the non-standard segmentation. The French phrase *II y a* → *Il y a* (‘there is’) and the spaced abbreviation *8. A. C.* → *S. A. C.* were each treated as a single segment and could be safely corrected, while the single tokens *II* and *8.* are not that easily identified as errors. Furthermore, *tobeis* → *tobels* corrects the last part of the compound place name *Welschtobel* (in genitive case), which is presumably supported by occurrences of this segment in different compounds.

While the overall performance of the presented error reduction system is not overwhelming, the unsupervised segmentation scheme is able to address certain kinds of errors that are harder to find by word-based correction systems. In future work, I intend to test the unsupervised segmentation with a more sophisticated correction algo-

[6] Measured by the *newest* data, which might be too low an estimate, since there might still be uncorrected errors.

rithm.

References

Neil Barrett and Jens H. Weber-Jahnke. 2011. Building a biomedical tokenizer using the token lattice design pattern and the adapted Viterbi algorithm. *BMC Bioinformatics*, 12(S-3).

Youssef Bassil and Mohammad Alwani. 2012. OCR context-sensitive error correction based on Google Web 1T 5-Gram data set. *American Journal of Scientific Research*, 50(50), February.

Stanley F. Chen and Joshua T. Goodman. 1998. An empirical study of smoothing techniques for language modeling. Technical report tr-10-98, Harvard University.

Tagyoung Chung and Daniel Gildea. 2009. Unsupervised tokenization for machine translation. In *Proceedings of the 2009 Conference on Empirical Methods in Natural Language Processing*, EMNLP '09, pages 718–726. ACL.

Lenz Furrer. 2013. Unsupervised text segmentation for correcting OCR errors. Master's thesis, University of Zurich.

Felix Golcher. 2006. Statistical text segmentation with partial structure analysis. In *Proceedings of KONVENS*, pages 44–51.

Sharon Goldwater, Thomas L. Griffiths, and Mark Johnson. 2006. Contextual dependencies in unsupervised word segmentation. In *Proceedings of the Annual Meeting of the Association for Computational Linguistics*, ACL '06, pages 673–680.

Margaret A. Hafer and Stephen F. Weiss. 1974. Word segmentation by letter successor varieties. *Information Storage and Retrieval*, 10(11-12):371–385.

Harald Hammarström and Lars Borin. 2011. Unsupervised learning of morphology. *Computational Linguistics*, 37(2):309–350.

Zellig S. Harris. 1955. From phoneme to morpheme. *Language*, 31(2):190–222.

Zellig S. Harris. 1967. Morpheme boundaries within words: Report on a computer test. In *Transformations and Discourse Analysis Papers*, pages 68–77. Department of Linguistics, University of Pennsylvania.

Kenneth Heafield. 2011. KenLM: Faster and smaller language model queries. In *Proceedings of the Sixth Workshop on Statistical Machine Translation*, Edinburgh, UK, July. Association for Computational Linguistics.

Marcin Heliński, Miłosz Kmieciak, and Tomasz Parkoła. 2012. Report on the comparison of Tesseract and ABBYY FineReader OCR engines. Technical report, Poznań Supercomputing and Networking Center (PCSS), Poznań.

Rose Holley. 2009. How good can it get? analysing and improving OCR accuracy in large scale historic newspaper digitisation programs. *D-Lib Magazine*, 15(3/4).

Reinhard Kneser and Hermann Ney. 1995. Improved backing-off for m-gram language modeling. In *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing*, volume I, pages 181–184, Detroit, Michigan, May.

Okan Kolak, William Byrne, and Philip Resnik. 2003. A generative probabilistic OCR model for NLP applications. In *Proceedings of the 2003 Conference of the North American Chapter of the Association for Computational Linguistics on Human Language Technology*, volume 1 of *NAACL '03*, pages 55–62, Stroudsburg, PA, USA. Association for Computational Linguistics.

William B. Lund and Eric K. Ringger. 2009. Improving optical character recognition through efficient multiple system alignment. In *Proceedings of the 9th ACM/IEEE-CS joint conference on Digital libraries*, JCDL '09, pages 231–240, New York, NY, USA.

Stoyan Mihov and Klaus U. Schulz. 2004. Fast approximate search in large dictionaries. *Computational Linguistics*, 30:451–477.

Stoyan Mihov, Petar Mitankin, Annette Gotscharek, Ulrich Reffle, Klaus U. Schulz, and Christoph Ringlstetter. 2007. Tuning the selection of correction candidates for garbled tokens using error dictionaries. In *Finite State Techniques and Approximate Search: Proceedings of the First Workshop on Finite-State Techniques and Approximate Search*, pages 25–30, Borovets, Bulgaria.

Naoaki Okazaki and Jun'ichi Tsujii. 2010. Simple and efficient algorithm for approximate dictionary matching. In *Proceedings of the 23rd International Conference on Computational Linguistics*, COLING '10, pages 851–859, Beijing, China, August.

Martin Reynaert. 2004. Text induced spelling correction. In *Proceedings of the 20th International Conference on Computational Linguistics*, COLING '04, pages 834–840, Geneva, Switzerland, Aug 23–Aug 27. ICCL.

Martin Reynaert. 2006. Corpus-induced corpus cleanup. In *Proceedings of the 5th International Conference on Language Resources and Evaluation*, LREC 2006, pages 87–92. European Language Resources Association (ELRA).

Martin Reynaert. 2008. Non-interactive OCR post-correction for giga-scale digitization projects. In *Proceedings of the 9th International Conference on Computational Linguistics and Intelligent text pro-*

cessing, CICLing'08, pages 617–630, Berlin, Heidelberg. Springer-Verlag.

Stephen V. Rice. 1996. *Measuring the Accuracy of Page-Reading Systems*. Ph.D. thesis, University of Nevada, Las Vegas.

Xian Tong and David A. Evans. 1996. A statistical approach to automatic OCR error correction in context. In *Proceedings of the Fourth Workshop on Very Large Corpora*, WVLC-4, pages 88–100.

Martina Tumulla. 2008. IMPACT: Improving access to text. *Dialog mit Bibliotheken*, 20(2):39–41, July. (German article).

Martin Volk, Torsten Marek, and Rico Sennrich. 2010. Reducing OCR errors by combining two OCR systems. In *ECAI 2010 Workshop on Language Technology for Cultural Heritage, Social Sciences, and Humanities*, LaTeCH 2010, pages 61–65, August.

Jesse O. Wrenn, Peter D. Stetson, and Stephen B. Johnson. 2007. An unsupervised machine learning approach to segmentation of clinician-entered free text. In *Proceedings of the AMIA 2007 Annual Symposium*, pages 811–815.

Efficient Training Data Enrichment and Unknown Token Handling for POS Tagging of Non-standardized Texts

Melanie Neunerdt, Michael Reyer, Rudolf Mathar
Institute for Theoretical Information Technology
RWTH Aachen University, Germany
{neunerdt, reyer, mathar}@ti.rwth-aachen.de

Abstract

In this work we consider the problem of social media text Part-of-Speech tagging as fundamental task for Natural Language Processing. We present improvements to a social media Markov model tagger, by adapting parameter estimation methods for unknown tokens. In addition, we propose to enrich the social media text corpus by a linear combination with a newspaper training corpus. Applying our tagger to a social media text corpus results in accuracies of around 94.8%, which comes close to accuracies for standardized texts. [1]

1 Introduction

Part-of-Speech (POS) tag information can be achieved by automatic taggers with accuracies up to 98% for standardized texts. However, when applying state-of-the-art taggers to non-standardized texts such as social media texts or spoken language, tagging accuracies drop significantly. Social media texts suffer from informal writing style such as misspelled or shortened words, which leads to a high number of unknown (out-of-vocabulary) tokens. Thus, some special challenges are given for developing methods for automatic social media text POS taggers. In this work we propose some adapted parameter estimation methods to our social media Markov model tagger, *WebTagger* (Neunerdt et al., 2013a). We improve the parameter estimation for unknown tokens in several ways. Beside different combination methods for tokens' prefix and suffix tag distributions, we propose a semi-supervised verb auxiliary lexicon. Furthermore, we consider the different grammatical structure of social media and newspaper texts leading to diverse distributions of POS tag sequences. In contrast to existing POS tagging approaches, we propose a linear combination of a social media training corpus and a newspaper corpus by an efficient oversampling of the in-domain training data. We experimentally evaluate the proposed methods for a German social media text corpus and different social media text types. Results are compared to the underlying *WebTagger* and state-of-the art widely used POS taggers. We show that by applying our adapted Markov model tagger to an existing social media text corpus we are able to obtain accuracies close to 95%.

The paper is organized as follows: Section 2 summarizes the related work to provide an overview of POS tagging, particularly for non-standardized texts. In Section 3 and 4 we introduce the basic tagger model and propose our adapted parameter estimation methods. Section 5 reports experimental results. In Section 6 we conclude our work.

2 Related Work

Performance investigations of state-of-the art taggers (Toutanova et al., 2003; Schmid, 1995) show that automatic POS tagging of non-standardized social media texts results in significant accuracy drops, see (Giesbrecht and Evert, 2009; Neunerdt et al., 2013b) . Therefore, recent publications (Gadde et al., 2011; Owoputi et al., 2012; Owoputi et al., 2013; Rehbein, 2013; Neunerdt et al., 2013a) particularly deal with the task of tagging non-standardized texts, such as twitter messages or Web comments. (Gadde et al., 2011) introduce feature adaptions to the Stanford maximum entropy tagger (Toutanova et al., 2003), to handle noisy English text. Results are evaluated based on an SMS dataset. In (Gimpel et al., 2011) a twitter tagger based on a conditional random field (CRF) with features adapted to twitter characteristics is proposed. They propose some additional word clustering and further improvement to their method in (Owoputi et al., 2013) and evaluate their approach on different English twitter data, where a maximal accuracy of 92.8% is achieved. (Rehbein, 2013; Neunerdt et al., 2013a) propose POS taggers for German social media texts. In (Rehbein, 2013) a CRF POS tagger for German Twitter microtexts is presented. Applying word clustering with features extracted from an automatically created dictionary leads, to 89% accuracy, which is slightly lower then results achieved for English twitter data. In (Neunerdt et al., 2013a) a Markov model tagger, called *WebTagger*, for the application to Web comments is proposed. Improvements are particularly achieved by the mapping of unknown tokens to known training tokens or some regular expressions. Furthermore, a semi-supervised auxiliary lexicon is proposed. Tagging accuracies of about 94% are achieved on a Web comment corpus. The proposed *WebTagger* serves as a basis for the methods introduced in this work.

3 Tagger Model

As a basic tagger model we use the Markov model proposed in (Neunerdt et al., 2013a). In this section we shortly explain this basic model. The aim of the tagger is to predict the associated POS tag sequence $t_1, \ldots, t_n, \ldots, t_N$ with $t_n \in \mathcal{T}$ (STTS) for a given sequence of tokens $w_1, \ldots, w_n, \ldots, w_N$ with $w_n \in \mathcal{W}$, where $\mathcal{W}$ contains all possible tokens. The optimization task is given as

$$\hat{t}_1^N = \arg\max_{t_1^N} P(t_1^N, w_1^N)$$

with a sequence of POS tags t_l^n

$$t_l^n = \begin{cases} (t_l, \ldots, t_n) & 1 \leq l \leq n \leq N \\ (t_1, \ldots, t_n) & l \leq 0 \end{cases}$$

where $l \in \mathbb{Z}$, $n \in \mathbb{N}$, and $l \leq n \leq N$. The sequence of tokens w_l^n is defined analogously. By applying the probability chain rule and some simplifying assumptions the optimization problem is solved by:

$$\hat{t}_1^N = \arg\max_{t_1^N} \prod_{n=1}^{N} \overbrace{\frac{P(t_n \mid w_n)}{P(t_n)}}^{LexicalProb.} \overbrace{P(t_n \mid t_{n-k}^{n-1})}^{TransitionProb.}$$

where $k \in \mathbb{N}$ describes the dependency depth of transition probabilities. Before the tagger can be used to predict the associated POS tag sequence $\hat{t}_1^N$, lexical und transition probabilities have to be estimated. Estimation of transition probabilities are inherited from (Neunerdt et al., 2013a). Lexical probability estimation methods are adapted and complemented, by our proposed methods described in the following section.

4 Lexical Probability Estimation

Lexical probability estimation differs significantly depending on wether a token is known or unknown from the training corpus. Whereas for known tokens the empirical distributions is accessible from the training, in the unknown case it is a more challenging task. However, we still know some characteristics of the word, e.g. the prefix and suffix of a word or some knowledge from an unsupervised or semi-supervised corpus.

In the following section we propose adaptions to a social media text tagger based on such characteristics and knowledge. In order to describe our estimation methods we first introduce a manually annotated social media text corpus

$$\mathcal{TR}_{ID} = \{(\hat{w}_i, \hat{t}_i) \mid 1 \leq i \leq I\} \qquad (1)$$

which is used for training. For each word $\hat{w}_n$ the correct tag $\hat{t}_n$ is known. Furthermore, we treat lexical probabilities as position independent and hence replace $P(t_n \mid w_n) = P(t \mid w)$ in the following notation.

4.1 Prefix/Suffix Combination

Previous work has shown that a words' prefix and suffix can successfully be used to determine the words' POS tag. Based on the set of training tokens $\mathcal{W}$ we determine all prefixes $p \in P$ and suffixes $s \in S$ of maximal length five. We assess the lexical probabilities for a given word w with its prefix $p(w)$ by:

$$\hat{P}_p(t \mid w) = \frac{|\{i|\hat{t}_i=t \wedge p(\hat{w}_i)=p(w)\}|}{|\{i|p(\hat{w}_i)=p(w)\}|}$$

Lexical probabilities $\hat{P}_s(t \mid w)$ are defined equivalently. The open question is, how to combine prefix and suffix tag distributions. In our approach we propose four different combination methods and discuss and compare them in Section 5. First, we assume prefix and suffix tag distributions to be independent and hence use the joint probability distribution

$$\hat{P}^g_{ps}(t \mid w) = \frac{\hat{P}_p(t|w)\hat{P}_s(t|w)}{\sum_t \hat{P}_p(t|w)\hat{P}_s(t|w)}$$

later referred as *geometric mean*. Combining prefix and suffix distributions in that way has been successfully be applied to POS tagging performed on newspaper texts in (Schmid, 1995). However, the characteristics of unknown tokens in social media texts differ from those appearing in newspaper texts. A more robust method for uncommon prefix and/or suffix, which arise from informal writing style characteristics, e.g. word shortenings or typing errors is needed. Therefore, in a second step we combine prefix and suffix tag distributions by building the *arithmetic mean* value for each tag probability, as proposed in our previous work, (Neunerdt et al., 2013a):

$$\hat{P}^a_{ps}(t \mid w) = \frac{\hat{P}_p(t|w)+\hat{P}_s(t|w)}{\sum_t \left(\hat{P}_p(t|w)+\hat{P}_s(t|w)\right)} \qquad (2)$$

In a third step, we define an approach aiming at choosing the most reliable tag distribution between $\hat{P}_p(t \mid w), \hat{P}_s(t \mid w)$. Therefore the entropy of prefix and suffix tag distributions is used as a criteria. We introduce random variables $T_{p(w)} \sim \left(\hat{P}_p\left(t \mid w\right)\right)_{t \in \mathcal{T}}$ and $T_{s(w)}$ analogously. The idea is to minimize the conditional entropy and hence chose the tag distributions, which contains less uncertainty about the tag t to predict:

$$\hat{X} = \arg\min_{X \in \{T_{p(w)}, T_{s(w)}\}} H(X) \qquad (3)$$

with

$$H(T_{p(w)}) = -\sum_{t \in \mathcal{T}} \hat{P}_p(t \mid w) \log \hat{P}_p(t \mid w)$$

and $H(T_{s(w)})$ analogously. However, the significance of the empirical prefix/suffix POS tag distribution, strongly depends on the frequency of prefixes/suffixes. A prefix, which has been seen once, leeds to zero uncertainty about the tag and will fulfill the minimum criteria. Hence, we apply some simple tests on the frequencies before applying the minimum entropy approach (3). The first test checks, if the frequencies of both prefix and suffix exceed a predefined threshold α, i.e.,

$$\hat{P}_{p(w)} > \alpha \wedge \hat{P}_{s(w)} > \alpha \qquad (4)$$

In that case the distribution given by $\hat{X}$ in (3) is used. As optional tests we check if exactly one of the thresholds is exceeded and use the corresponding probability distribution. If all these tests fail the distribution from (2) is taken. We will evaluate this strategy later on, with and without the optional tests, referred as *Rule-based-2-case* and *Rule-based-4-case*.

4.2 Semi-supervised Verb Auxiliary Lexicon

Investigating tagging results of state-of-the art newspaper taggers applied to social media texts, exhibit a frequent number of unknown verbs. This can be explained by the different dialogic style of social media texts, where different verb conjugations occur. Even a tagger trained on social media data, only contains a small part of such verbs, due to the small corpus size. Furthermore, lexical probabilities can not reliably be estimated from prefix and suffix tag distributions for such verbs. However, preparing a fully-supervised social media training text with adequate corpus size is extremely time-consuming and demands expert knowledge from the annotator. We propose an alternativ approach, which reduces annotation effort significantly.

The basic idea is to create a verb auxiliary lexicon with corresponding tag sets for each token. For approximately 14,000 verbs, a conjugation table including indicative and subjunctive for different tenses as well as the imperative, participle and infinitive is extracted from *www.verbformen.de*. For an exemplary conjugation table, the corresponding POS tag is assigned manually to each verb form. Corresponding POS tags are automatically transferred to all other conjugation tables. Based on that conjugation tables all possible tokens with their corresponding tags denoted by $\mathcal{T}_{w_m}$ are combined in a verb auxiliary lexicon $\mathcal{V}^+$ containing 115,000 entries. If there is more than one possible tag, an adequate tag distribution needs to be assigned. Therefore, two approaches are utilized. First, all words $\hat{w}_k$ of the manually annotated training corpus with the same POS tag set $\mathcal{T}_{w_m}$ are determined and the cumulated tag distribution of those words is taken. Hence, the lexical probability is refined as

$$\hat{P}_{\mathcal{V}^+}(t \mid w_m) = \frac{|\{k|\hat{t}_k{=}t{\wedge}\mathcal{T}_{\hat{w}_k}{=}\mathcal{T}_{w_m})\}|}{|\{k|\mathcal{T}_{\hat{w}_k}{=}\mathcal{T}_{w_m}\}|},$$

where $\mathcal{T}_{\hat{w}_k} = \{\hat{t}_l \mid \hat{w}_l = \hat{w}_k\}$. We assume all $t \in \mathcal{T}_{w_m}$ to be equally distributed, if no word with the same POS tag set $\mathcal{T}_{w_m}$ exists. If a token is not known from training or the verb auxiliary lexicon, prefix-/suffix estimations described in the previous section is performed.

4.3 Joint-Domain Training

In this section, the term *domain* is associated with a text corpus characterized by a particular style characteristic. A social media text corpus is mentioned as in-domain corpus, whereas all text with different characterization are out-domain texts. We define the combination of in- and out-domain training data as joint-domain training. Different experimental studies have shown that out-domain training data can improve tagging accuracies, e.g., (Rehbein, 2013; Neunerdt et al., 2013a). This particularly holds, if the available in-domain corpus of small size only. A typical approach is to stepwise increase the amount of out-domain training and retrain the tagger on such data. Then the amount of out-domain training data achieving best results is determined.

In contrast to existing approaches, we suggest an alternative method for combining in- and out-domain training data. The basic idea is a weighted joint-domain training. A manually annotated newspaper training corpus

$$\mathcal{TR}_{OD} = \{(\dot{w}_n, \dot{t}_n) \mid 1 \leq n \leq O\}$$

is added to our *WebTrain* corpus (1). In contrast to other approaches information from the whole available out-domain training corpus is used, no matter about corpus size. To cope with the different corpora sizes , we apply oversampling to the in-domain social media text corpus. Therefore, we multiply the *WebTrain* corpus $\beta \in \mathbb{N}$ times, while combining it with the newspaper corpus. We use a set of combined training pairs

with $\mathcal{TR} = \{(\tilde{w}_n, \tilde{t}_n) \mid 1 \leq n \leq \tilde{N} = O + \beta I\}$

$$(\tilde{w}_n, \tilde{t}_n) = \begin{cases} (\dot{w}_n, \dot{t}_n) & 1 \leq n \leq O \\ (\hat{w}_i, \hat{t}_i) & n > O\,, i = (n - O - 1 \mod I) + 1. \end{cases}$$

Table 1: Tagger evaluation for different estimation methods based on prefix and suffix information.

	Mean Precision		**Mean Recall**		**Mean Accuracy**	
	Pref/Suf	Total	Pref/Suf	Total	Pref/Suf	Total
***WebTrain* Test**						
Geometric	**61.43**	**84.96**	43.16	85.66	71.37	94.66
Arithmetic	53.06	84.43	51.03	85.82	**73.97**	**94.79**
Rule-base 2-case	51.58	84.70	50.86	85.76	73.50	94.77
Rule-base 4-case	41.88	84.65	**51.58**	**86.00**	71.90	94.68
***WebTypes* Test**						
Geometric	**37.96**	**80.67**	26.64	80.13	57.08	90.42
Arithmetic	35.02	78.67	**35.24**	80.47	58.02	90.63
Rule-base 2-case	35.84	78.73	34.31	**80.68**	**58.09**	**90.66**
Rule-base 4-case	29.96	78.29	34.07	80.42	56.28	90.48

The method of oversampling, see ,e.g., (Pelayo and Dick, 2007), has originally been proposed to handle the class imbalance problem in a sample corpus. Combining imbalanced in- and out-domain training data corpora has not yet been performed to the problem of POS tagging.

5 Experimental Results

We first evaluate the treatment of unknown words with different prefix/suffix estimation methods and with the semi-supervised verb auxiliary lexicon. After comparing the proposed WebTagger with two state-of-the art taggers, the performance increase by weighted joint-domain training is pointed out in more detail in 5.2.

For the purpose of training two corpora, an in-domain social media corpus and out-domain newspaper text corpus are used. As social media texts, we use the *WebTrain* corpus with Web comments containing 36,000 tokens, introduced by (Neunerdt et al., 2013b). A detailed description and further corpus statistics can be found in (Neunerdt et al., 2013b; Neunerdt et al., 2013a). Annotation rules, particularly for social media text characteristics, and inter-annotator agreement results are given in (Trevisan et al., 2012). As a newspaper corpus we use the *TIGER* treebank (Brants et al., 2004) text corpus, containing 890,000 tokens. In order to test the tagger with different parameter settings on different social media text types, we use the *WebTypes* corpus (Neunerdt et al., 2013a) as additional test data. All corpora are annotated with manually validated POS tags according to the STTS annotation guideline.

5.1 Unknown Word Treatment Analysis

For all evaluations in this section, we perform ten 10-fold cross validations on the *WebTrain* corpus.

WebTrain subsets are created by randomly selecting sentences. The following results are mean values over the resulting 100 training and test sample pairs. Note that for all cross validations the taggers are trained in a combination with 90% of the *TIGER* corpus. The remaining *TIGER* subset is used for testing.

First, we discuss different prefix/suffix combinations methods. All cross validation results for the different methods are depicted in the upper part of Table 1. On the average each *WebTrain* test set contains about 4.22% tokens, where prefix/suffix estimation is applied. We calculate mean class precision and recall rates and the total accuracies for the whole test text (Total) and for the tokens, where the prefix/suffix estimation is applied (Pref/Suf). Experimentally we determine $\alpha = 50$ to be the best threshold for the *Rule-based-2-case (R-b2c)* and *Rule-based-4-case (R-b4c)* method and depict results for that value. In order to investigate the influence on different social media text types, we additionally apply all taggers to the *WebTypes* corpus, where prefix/suffix estimation is applied to 8.44% tokens on average. Results are depicted in the lower part of Table 1. The arithmetic mean method results in the best overall *WebTrain* accuracies. However, considering the mean class precision, the geometric mean method significantly outperforms the other methods with 61.43% accuracy achieved on prefix/suffix tokens. The R-b4c approach reaches slightly better mean class recall results compared to the arithmetic mean. Hence, depending on the later application, requiring POS tag information, one might be rather interested in a high per class accuracy in contrast to the total accuracy and rather prefer one of the later mentioned methods. Results achieved on the *WebTypes* data basically confirm these cross validation results. However, the R-b2c method slightly increases mean accuracies and total recall rates.

In the following, we evaluate the performance of the semi-supervised verb auxiliary lexicon and decide to use prefix/suffix combination by the *arithmetic mean* method for all following evaluations. Cross validation accuracies achieved for *WebTrain*, *WebTypes* and *TIGER* are depicted in Table 2 without (*) and with (-) verb lexicon. In addition to total accuracies, unknown word accuracies are depicted. The introduction of the verb lexicon increases the unknown word accuracy about 1 percentage point, whereas the verb lexicon achieves about 80% accuracy, which is significantly higher compared to prefix/suffix methods. Noteable is that the performance of the verb lexicon drops about 20 percentage points, when applied to *WebTypes*. This can be explained by a high number of verbs, where no known word with the same POS tagset T_{w_m} exists and hence estimates are less reliable, due to the equal tag distribution. Furthermore, it has to be considered that accuracies are averaged over 100 different trainings but the *WebTypes* test set is fixed and hence not exactly comparable.

Finally, we compare the adapted WebTagger to two state-of-the art taggers, TreeTagger (Schmid, 1995) and Stanford (Toutanova et al., 2003), see Table 2. Both taggers are trained and tested on the same 100 samples using their standard parameters. Influence of linear combined joint-domain training leads to 0.36 and 0.51 percentage points improvement for *WebTrain* and *WebTypes* (forth column). Joint-domain training methods are studied in more detail in Section 5.2. The adapted WebTagger significantly outperforms both state-of-the art taggers, when applied to social media texts. Differences between the taggers are statistically significant according to a corrected resampled paired t-test (Nadeau and Bengio, 2001) applied to all cross validation with a significance level of $p = 0.001$. All results achieved with the adapted WebTagger on the newspaper test drop slightly. This is due to the β factorization towards the *WebTrain* corpus. However, the tagger is developed for social media texts.

5.2 Influence of Joint-domain Training

In this section we investigate the influence of out-domain training data in more detail. We particularly compare our proposed linear combination of joint-domain training to existing approaches, where the ratio between in- and out-domain training is adjusted by the out-domain corpus size. First we stepwise increase the amount of *TIGER* training data. Starting with a size equal to *WebTrain corpus* size, we randomly choose sentences in each step. This is performed 100 times and data is added to the data selected in the previous

Table 2: Tagger evaluation for different text types trained on joint-domain data.

	#Tokens	WebTagger (Neunerdt et al., 2013a) (*)		WebTagger (*) +Verblexicon $\mathcal{V}^+$ (-)			WebTagger (-) +$\beta = 10$ fact.	TreeTagger	Stanford
		Unknown	Total	Unknown	Verb	Total	Total	Total	Total
WebTrain test	3,628	76.06	94.38 ± 0.46	77.05	80.72	94.43 ± 0.47	$\mathbf{94.79 \pm 0.45}$	93.84 ± 0.55	93.50 ± 0.56
WebTypes	4,006	62.53	90.11 ± 0.11	62.89	60.96	90.12 ± 0.12	$\mathbf{90.63 \pm 0.12}$	88.02 ± 0.13	86.27 ± 0.10
TIGER test	88,910	88.87	97.22 ± 0.01	90.07	90.58	97.28 ± 0.01	$\mathbf{97.13 \pm 0.01}$	97.98 ± 0.01	98.69 ± 0.01

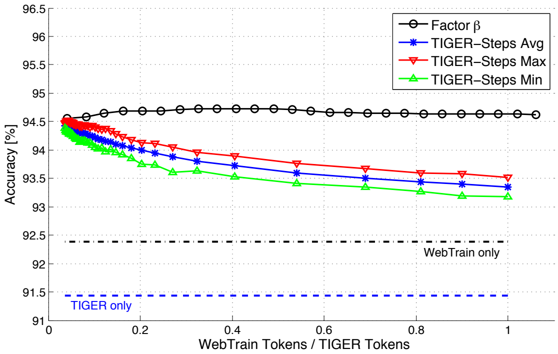


Figure 1: Influence of different joint-domain trainings evaluated on *WebTrain*.

step. Each of these out-domain training samples is combined with each training of a 10-fold *WebTrain* cross validation (3,600 tokens each part). Mean accuracies of cross validation tagging over all 1000 training samples are depicted for different in-/out domain ratios in the blue curve ($\star$) in Figure 1. Additionally the minimum and maximum accuracy of the 100 *TIGER* training samples is depicted in the green ($\triangle$) and red curve (∇). In order to give some reference values, we train our tagger exclusively on the *TIGER*/*WebTrain* corpus. Accuracies are depicted by the blue and black dotted line in both figures. Second we apply our linear combination approach and combine the *TIGER* and *WebTrain* corpus in the same cross validation for different β values. Cross validation results and test results achieved are depicted in the black curve ($\circ$). First, we compare the accuracies achieved with our approach (black curve, $\circ$) to those achieved with the best *TIGER* training part (red curve, ∇). The black curve ($\circ$) stays above the red curve (∇) over all in-/out-domain ratios. The red curve (∇) represents the optimum result for the given number of out-domain tokens. The plot indicates that exploiting this degree of freedom the performance of our approach is hardly reached. Determining the optimum training corpus results in a huge evaluation effort, which is very time consuming. If the *TIGER* training part is not determined properly and, e.g., chosen randomly, tagging accuracies can be significantly lower. In the worst case minimum accuracies depicted in the green curve ($\triangle$) are achieved. Applying our method with $\beta = 10$ results in a maximum cross validation accuracy of 94.79%. Determining the best β is considerably faster compared to identifying the best *TIGER* training part. Even if no effort is spent on determining the best β, accuracies are only slightly lower than optimum. Considering these evaluations it is obvious that our approach is robust in the sense that the performance slightly changes, if the ratio of tokens is changed. The result depicted in Figure 1 show the robustness of our method, no matter what β values we choose. Finally, we compare the results achieved for exclusively trained taggers on *TIGER*/*WebTrain* corpus. All combination methods significantly exceed accuracies achieved for single training over all in-/out domain ratios. This states that a joint-domain training approach is always reasonable.

6 Conclusion

We have compared state-of-the art taggers with our adapted WebTagger. It outperforms the others considerably with an average accuracy of 94.8% applied to a German social media text corpus. Additionally, it yields a minimum improvement compared to state-of-the art taggers of 2.6% percentage points for a social media text type corpus different from the training corpus type. In our approach we have improved the following two items of the original WebTagger. First we have amended the estimation of lexical probabilities for unknown tokens by introducing tag distributions derived from prefix and suffix information and a semi-supervised verb auxiliary lexicon. Second we have enriched the social media text corpus by a linear combination following an oversampling technique with a newspaper training corpus.

References

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language & Computation*, pages 597–620.

Phani Gadde, L. V. Subramaniam, and Tanveer A. Faruquie. 2011. Adapting a WSJ Trained Part-of-Speech Tagger to Noisy Text: Preliminary Results. In *Proceedings of the 2011 Joint Workshop on Multilingual OCR and Analytics for Noisy Unstructured Text Data*, pages 5:1–5:8.

Eugenie Giesbrecht and Stefan Evert. 2009. Is Part-of-Speech Tagging a Solved Task? An Evaluation of POS Taggers for the German Web as Corpus. In *Proceedings of the Fifth Web as Corpus Workshop*, pages 27–35.

Kevin Gimpel, Nathan Schneider, Brendan O'Connor, Dipanjan Das, Daniel Mills, Jacob Eisenstein, Michael Heilman, Dani Yogatama, Jeffrey Flanigan, and Noah A. Smith. 2011. Part-of-Speech tagging for Twitter: annotation, features, and experiments. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 42–47.

Claude Nadeau and Yoshua Bengio. 2001. Inference for the generalization error. *Maschine Learning*.

Melanie Neunerdt, Michael Reyer, and Rudolf Mathar. 2013a. A POS Tagger for Social Media Texts trained on Web Comments. *Polibits*, 48:59–66.

Melanie Neunerdt, Michael Reyer, and Rudolf Mathar. 2013b. Part-of-Speech Tagging for Social Media Texts. In *Proceedings of The International Conference of the German Society for Computational Linguistics and Language Technology*.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, and Nathan Schneider. 2012. Part-of-Speech Tagging for Twitter: Word Clusters and Other Advances. Technical report, School of Computer Science, Carnegie Mellon University.

Olutobi Owoputi, Brendan OConnor, Chris Dyer, Kevin Gimpel, Nathan Schneider, and Noah A Smith. 2013. Improved part-of-speech tagging for online conversational text with word clusters. In *Proceedings of NAACL-HLT*, pages 380–390.

L. Pelayo and S. Dick. 2007. Applying novel resampling strategies to software defect prediction. In *Fuzzy Information Processing Society, 2007. NAFIPS '07. Annual Meeting of the North American*, pages 69–72, June.

Ines Rehbein. 2013. Fine-grained pos tagging of german tweets. In *Language Processing and Knowledge in the Web*, volume 8105 of *Lecture Notes in Computer Science*, pages 162–175. Springer Berlin Heidelberg.

Helmut Schmid. 1995. Improvements in Part-of-Speech Tagging With an Application to German. In *Proceedings of the ACL SIGDAT-Workshop*, pages 47–50.

Kristina Toutanova, Dan Klein, Christopher D. Manning, and Yoram Singer. 2003. Feature-rich Part-of-Speech Tagging With a Cyclic Dependency Network. In *Proceedings of Human Language Technology Conference*, pages 173–180.

Bianka Trevisan, Melanie Neunerdt, and Eva-Maria Jakobs. 2012. A Multi-level Annotation Model for Fine-grained Opinion Detection in German Blog Comments. In *Proceedings of KONVENS 2012*, pages 179–188.

Biased Parsing for Syntactic Evaluation of Machine Translation

Melania Duma
Department Informatik
University of Hamburg
Vogt-Kölln-Strasse 30, 22527 Hamburg
duma@informatik.uni-hamburg.de

Wolfgang Menzel
Department Informatik
University of Hamburg
Vogt-Kölln-Strasse 30, 22527 Hamburg
menzel@informatik.uni-hamburg.de

Abstract

Syntactic Machine Translation evaluation is often based on the parsing of Machine Translation output. However, this is a challenging task, mainly due to the ungrammaticality of the candidate translations. To overcome this problem we present a method of guiding the parsing of the candidate translation by taking into consideration syntactic information extracted from the parse of the reference translation. For this purpose, we modified a Weighted Constraint Dependency Grammar parser by integrating a predictor component in the parsing process. The evaluation was performed using a syntax based Machine Translation evaluation metric and the results show that the method brings improvement to the correlation scores.

1 Introduction

Machine Translation (MT) evaluation aims at providing a set of automatic methods for assessing the quality of MT output. These automatic methods can be grouped into different categories based on the level where the matching between a reference translation and the candidate is performed. On the lexical level, most of the metrics are based on matching word forms between the candidate translation and the reference translations. However, this form of matching cannot detect all the similarity aspects between sentences and therefore syntactic metrics have been introduced (e.g. Liu and Gildea, 2005; Owczarzak et al., 2007). On the other hand, syntactic metrics often require additional external processing tools, for example part-of-speech taggers or parsers, and therefore their results are influenced by the quality of the existing tools. In the case of parsers, the task of analyzing candidate translations proves to be particularly difficult due to the ungrammaticality of MT output. This leads to a decrease of the parsing accuracy, which in turn influences the final score of the syntactic metrics. The aim of the research presented in this paper was to decide whether it would be possible to guide the parsing of MT output with structural expectations in order to increase the accuracy of syntactic evaluation metrics for MT. To investigate this, we used a Weighted Constraint Dependency Grammar parser (Menzel and Schröder, 1998), which can easily be modified to bias the parsing process. This technique has already been successfully exploited to modulate parsing using external semantic information in limited domains (McCrae, 2007). The rest of this paper is organized as follows. In Section 2 the related work to our research is presented. WCDG is introduced in Section 3, followed by a description of modifications made to WCDG in order to bias the parsing of MT output. In Section 5 the experimental setup and the evaluation is reported, while in the last section conclusions and future work are presented.

2 Related work

There are a number of tools in the literature, which explore the idea of aligning a reference to a candidate translation. However, in contrast to our approach, these tools are mainly used to perform automatic error analysis of MT output (e.g. Popović , 2011; Zeman et al., 2011) or for the automatic evaluation of MT output (e.g. Denkowski and Lavie, 2014). Projection of dependencies is another research idea connected to our work. In Jiang and Liu (2010), dependency structures are projected through an alignment from a sentence in the source language to a sentence in the target language. Other work related to dependency projection is presented by Ganchev et al. (2009) or Huang et al. (2009). However, in contrast to our approach, all these methods aim at parsing grammatically correct input. Moreover, our work focuses on performing dependency projection for sentences in the same language by using a rule based parser.

Another line of research related to our work concentrates on improving the accuracy of MT output parsing. Owczarzak et al. (2007) propose a method to decrease parser noise when parsing MT output by using the n best parses for the syntactic evaluation of the candidate translation. In Rosa et al. (2012) a method for biasing the parsing is described. It is achieved by exploring different monolingual and bilingual features. The experiments were performed on English to Czech translations, using a reimplementation of the MST parser (McDonald et al., 2006). Artificial grammatical errors were inserted in the training data for the target language parser in order to increase its robustness. Evaluation was performed indirectly through the use of a post-processing tool for MT output, and showed improvement of the results for the manual and the automatic evaluation of MT outputs. While Rosa et al. (2012) applied a data driven parser, our approach uses a rule based one. Moreover we aim at biasing the parsing by extracting syntactic information from the reference translation, while in the case of Rosa et al. (2012) the biasing is achieved based on information extracted from the source sentence.

3 WCDG

Parsing using the Weighted Constraint Dependency Grammar (WCDG) (Menzel and Schröder, 1998) is achieved by means of graded constraints. Each constraint in WCDG has been manually assigned a weight, which is a value between 0 and 1. A constraint with the weight of 0 is called a hard constraint and represents a rule that always has to be imposed. The result of parsing with WCDG is a dependency structure, which is made up of a set of vertices and a set of edges. The constraints are used in order to decide between alternative structures and to determine which dependency structure is the optimal representation for the input sentence. Each vertex of the structure represents a token of the input phrase, while each edge connects a head to a dependent and is annotated with a label for the dependency relation. If an edge or a pair of edges does not comply with a constraint, then a constraint violation is raised. The score of a parse is calculated as the product of the weights pertaining to all the instances of violated constraints.

Based on the toleration of constraint violations, parsing with WCDG shows robustness against ungrammaticality. This is an important feature especially in the context of parsing MT output, which is often ungrammatical. Even in this case, WCDG always succeeds with a result, although the final dependency structure may receive a low score. Another advantage of WCDG is its ability to determine constraint violations, which can be exploited for error analysis. Moreover, the score of the parse could be seen as a means of differentiating between the qualities of the candidate translations. A further benefit of using WCDG is that it allows the integration of external predictors by involving them in the constraint evaluation process. This way, modules like a POS tagger, a chunker or a data driven parser have been successfully used to increase the parsing accuracy (Foth and Menzel, 2006; Khmylko et al., 2009). For our research we used jwcdg[1], which is a weighted constraint dependency grammar parser that features a grammar for German.

4 Biased parsing of MT output using WCDG

In order to guide the parsing of candidate translations, we have designed a predictor component, which suggests dependency relations likely to occur in the translation. The integration of the predictor into WCDG is shown in Figure 1.

The predictions are constructed based on the alignment between the reference and the candidate translations and on the dependency relations

[1] https://nats-www.informatik.uni-hamburg.de/CDG/WebHome

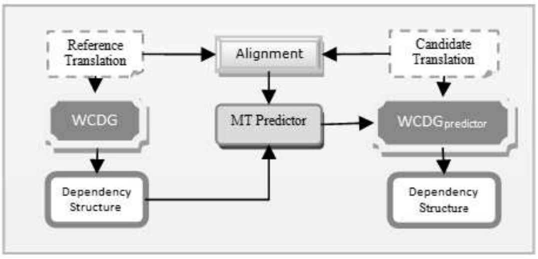


Figure 1: Integration of the predictor into WCDG.

without labels extracted from the parse tree of the reference translation. More precisely, as depicted in Figure 2, if:

- a token *m* from the candidate translation is aligned to a token *i* from the reference translation
- token *i* is subordinated to token *j* in the reference translation
- token *j* is aligned to token *n* from the candidate translation

then a prediction that token m is subordinated to token n is made. The predictions were integrated into WCDG through a constraint which was added to the grammar. This constraint is validated for every vertex of the dependency structure. It verifies the equality between the head identified by the predictor and the current head identified by WCDG. If they differ then a constraint violation is raised. A score of 0.5 was empirically determined to be the optimum weight for the new constraint. Experiments have shown that further weakening of the weight, led WCDG to overlook many high quality predictions in favor of other dependency structures which were preferred by the parser.

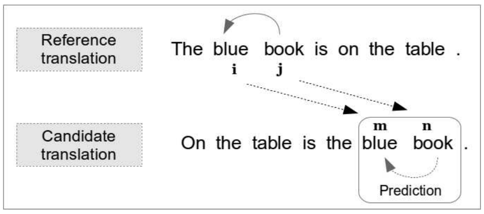

Reference translation
The blue book is on the table .
i
j
Candidate translation
On the table is the blue book .
m
n
Prediction

Figure 2: Example of prediction.

In order to generate accurate predictions, the quality of the alignment is critical. Therefore, the METEOR monolingual aligner (Banerjee and Lavie, 2005; Denkowski and Lavie, 2014) was used, as it was reported to achieve high precision and good recall in comparison with other alignment tools (Zeman et al., 2011). The METEOR aligner creates a mapping based on matching unigrams. The matching can be of multiple types: exact, stem, synonym or paraphrase. For our experiments, we took into consideration all the matching types, with the exception of synonym matching.

5 Evaluation

In this section the syntactic metric used for evaluation, the description of the experimental setup and the results are presented.

5.1 The Headword Chain based Metric

Among the existing syntactic metrics, Headword Chain based Metric (HWCM) (Liu and Gildea, 2005) was chosen, as a means to evaluate the parser, because of its high performance reported in Liu and Gildea (2005) and because it is based on the comparison of dependency parse trees. A headword dependency chain is defined as the sequence of words which forms a path in a dependency tree. HWCM computes the number of matching headword chains of different lengths extracted from the dependency trees of the reference and that of the candidate translations. The HWCM score is defined as:

$$HWCM = \frac{1}{D}\sum_{n=1}^{D}\frac{\sum_{g\in chain_n(hyp)} count_{clip}(g)}{\sum_{g\in chain_n(hyp)} count(g)}$$

where D is the maximum length of the chain, and $count(g)$ and $count_{clip}(g)$ represent the number of times that chain g appears in the dependency tree of the hypothesis. However, the latter one cannot exceed the maximum number of times the chain occurs in the parse of the reference translation.

5.2 The experimental setup

The evaluation was conducted based on data selected from the Workshop on Statistical Machine Translation (WMT) 2013 (Bojar et al., 2013). Because, at the moment, parsing with jwcdg is restricted to German, the experiments were performed on the English-German language pair. For this language pair 15 system outputs and their corresponding human judgments were made available. From each system test set we selected the first 1600 sentences. Therefore, our final test set consisted of 24000 candidate translations. In order to evaluate the predictor, each candidate translation from the test set was parsed using the original WCDG and the modified version of WCDG, which we will refer to as WCDG$_{\text{predictor}}$. Two HWCM scores, each corresponding to a version of the parser, were computed and their correlation with human judgments was determined. The evaluation was performed at system level and at segment level by using two nonparametric rank correlation coefficients.

5.3 System level evaluation

For the system level evaluation, the Spearman rank correlation coefficient was computed. It compares the ranking based on human judgments with the one based on HWCM system scores, which are calculated as the average of all the individual segment scores. We used the formula presented in (Macháček and Bojar, 2013):

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2-1)}$$

where d_i represents the difference between the human rank and the metric rank for system i and n is the number of systems. The result is a number between -1 and 1, where -1 shows that there is inverse correlation between the two rankings, while a result of 1 indicates absolute correlation between the two rankings.

Metric	WCDG	WCDG$_{\text{predictor}}$
HWCM-1	0.66	0.66
HWCM-2	0.74	**0.76**
HWCM-3	0.77	**0.79**
HWCM-4	0.82	0.82
HWCM-5	**0.83**	0.81
Average ρ	0.76	**0.77**
BLEU	0.85	
METEOR	0.83	

Table 1: Spearman rank correlation coefficient results.

The results for system level correlation according to WCDG and WCDG$_{\text{predictor}}$ are summarized in Table 1. Several HWCM versions were tested, with the length of the headword chain ranging between 1 and 5. In addition, the correlation for BLEU (Papineni et al., 2002) and METEOR[2] (Denkowski and Lavie, 2014) with human scores is also presented in this table. For the experiments we used the BLEU implementation included in the Phrasal[3] toolkit (Green et al., 2014), which computes sentence level smoothed scores based on Lin and Och (2004). The coefficients presented in Table 1, denote an increase in the correlation when WCDG$_{\text{predictor}}$ is used. In the case of HWCM-1 the equality in correlations can be explained by the fact that only unigrams are taken into account and the influence of WCDG$_{\text{predictor}}$ cannot be detected. The table also shows the average of the correlation scores over all the lengths of headword chains and it indicates that biased parsing leads to a better correlation of HWCM with the human judgments at system level.

5.4 Segment level evaluation

For the segment level evaluation of HWCM, the Kendall τ rank correlation coefficient was used. For its computation, we utilized formulas from (Macháček and Bojar, 2013), which are reproduced below:

$$Pairs := \{(a,b) | r(a) < r(b)\}$$

$$Con := \{(a,b) \in Pairs | m(a) > m(b)\}$$

$$Dis := \{(a,b) \in Pairs | m(a) < m(b)\}$$

$$\tau = \frac{|Con| - |Dis|}{|Con| + |Dis|}$$

where $Pairs$ is a set consisting of pairs of translations of the same segment where one translation was scored higher than the other translation by the human judges. The Con set represents the concordant pairwise comparisons, while the Dis set represents the discordant pairwise comparisons. A pair of translations is judged to be concordant if the order imposed by the human ranks is respected by the automatic scores, and is judged as discordant otherwise. The pairs of translations, with a tie occurring between the HWCM scores or the human ranks, are not taken into account in the calculation of the coefficient. The result of Kendall τ is a number between -1 and 1, where -1 denotes the fact that no concordant pairs were registered, while 1 shows that all pairs were judged as being concordant.

[2] www.cd.cmu.edu/ alavie/METEOR/

[3] nlp.stanford.edu/phrasal/

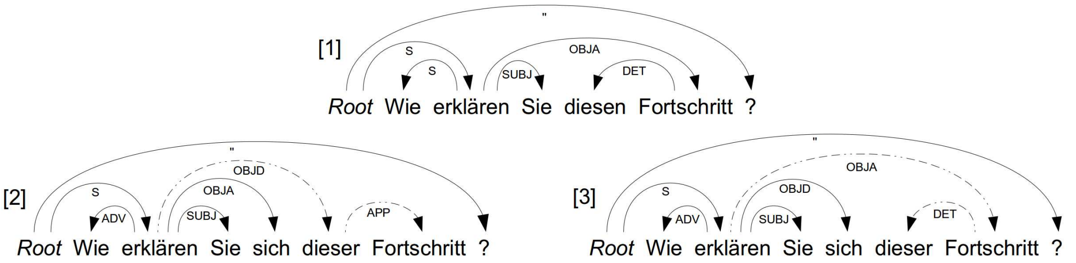

[1]
"
S
S
SUBJ
OBJA
DET
Root Wie erklären Sie diesen Fortschritt ?
[2]
"
OBJD
S
OBJA
ADV
SUBJ
APP
Root Wie erklären Sie sich dieser Fortschritt ?
[3]
"
OBJA
OBJD
S
ADV
SUBJ
DET
Root Wie erklären Sie sich dieser Fortschritt ?

Figure 3: Example of modified parse tree using WCDG$_{\text{predictor}}$, where [1] is the dependency structure of the reference translation, [2] is the dependency structure of the candidate translation obtained using WCDG and [3] is the dependency structure of the candidate translation obtained using WCDG$_{\text{predictor}}$. The differences between [2] and [3] are marked with dashed-dotted lines.

Metric	WCDG	WCDG$_{\text{predictor}}$
HWCM-1	0.124	0.124
HWCM-2	0.131	**0.135**
HWCM-3	**0.132**	0.129
HWCM-4	**0.133**	0.128
HWCM-5	**0.133**	0.129
Average ρ	**0.130**	0.129
BLEU	0.182	
METEOR	0.191	

Table 2: Kendall τ rank correlations obtained using human judgments.

The results of the segment level evaluation are outlined in Table 2 together with the correlation results for smoothed BLEU and METEOR. We can observe that for all the versions of the HWCM metric, corresponding to the different lengths of the headword chains, the results for the two parser versions are very similar, showing a slight increase or decrease of the correlation scores. The result of the difference between the averaged correlation for WCDG and WCDG$_{\text{predictor}}$ is 0.001. Even though the number of concordant and discordant pairs remains almost the same for both versions of the parser, the actual pairs that are present in the sets differ depending on the parser used. Therefore, we decided to also calculate the correlation between the HWCM scores and the BLEU and METEOR scores in order to evaluate the predictor component at segment level. Since BLEU and METEOR are high performance metrics that are part of the state of the art in MT evaluation, an adjustment of the correlation score can be perceived as a means of assessing the quality of the modifications introduced by WCDG$_{\text{predictor}}$. The results are presented in Table 3 and we can observe an increase of the correlations with BLEU and METEOR when using WCDG$_{\text{predictor}}$.

Metric	BLEU		METEOR	
	WCDG	WCDG$_{\text{pred}}$	WCDG	WCDG$_{\text{pred}}$
HWCM-1	0.573	0.573	0.571	0.571
HWCM-2	0.589	**0.597**	0.552	**0.564**
HWCM-3	0.576	**0.587**	0.528	**0.541**
HWCM-4	0.569	**0.578**	0.520	**0.531**
HWCM-5	0.568	**0.577**	0.519	**0.530**
Average τ	0.575	**0.582**	0.538	**0.547**

Table 3: Kendall τ rank correlations obtained using BLEU and METEOR scores.

5.5 Examples

In this subsection we present two examples for candidate translations, which have benefited from the addition of the predictor component. In the first example, depicted in Figure 5, the reference translation consists of the interrogative sentence *"Wie erklären Sie* ***diesen*** *Fortschritt?"* (engl. *"How do you explain this progress?"*), while the candidate translation is *"Wie erklären Sie sich* ***dieser*** *Fortschritt?"* (engl. *"How do you explain yourself this progress?"*). In the case of the candidate translation, due to the incorrect nominative case of the demonstrative pronoun *dieser* (engl. *this*), WCDG fails to identify its desired attachment, and it also fails to do so for the noun *Fortschritt* (engl. *progress*). Based on the predictions made, WCDG$_{\text{predictor}}$ manages to correct the attachments, so that *dieser* is attached as a determiner to *Fortschritt* and *Fortschritt* becomes the accusative object attached to the finite verb *erklären* (engl. *explain*). In turn, these corrections increase the average HWCM score of this sentence from 0.42 to 0.47. Furthermore, one

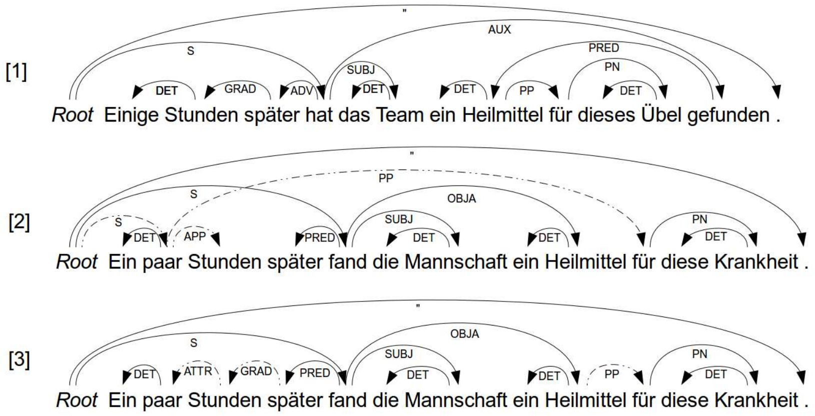

[1]
"
AUX
S
PRED
PN
SUBJ
DET
GRAD
ADV
DET
DET
PP
DET
Root Einige Stunden später hat das Team ein Heilmittel für dieses Übel gefunden .
[2]
"
PP
S
OBJA
S
SUBJ
PN
DET
APP
PRED
DET
DET
DET
Root Ein paar Stunden später fand die Mannschaft ein Heilmittel für diese Krankheit .
[3]
"
OBJA
S
SUBJ
PN
DET
ATTR
GRAD
PRED
DET
DET
PP
DET
Root Ein paar Stunden später fand die Mannschaft ein Heilmittel für diese Krankheit .

Figure 4: Example of modified parse tree using WCDG$_{\text{predictor}}$, where [1] is the dependency structure of the reference translation, [2] is the dependency structure of the candidate translation obtained using WCDG and [3] is the dependency structure of the candidate translation obtained using WCDG$_{\text{predictor}}$. The differences between [2] and [3] are marked with dashed-dotted lines.

of the differences between the candidate and the reference translations is the presence of a superfluous token, the reflexive pronoun *sich* (engl. *yourself*), which can be perceived as a translation error. Except for the change of the edge label WCDG$_{\text{predictor}}$ leaves the attachment unaltered.

In the second example presented in Figure 6, the reference consists of the independent clause *"Einige Stunden später hat das Team ein Heilmittel für dieses Übel gefunden."* (engl. *"A few hours later, the team found a remedy for this evil."*), while the candidate translation is *"Ein paar Stunden später fand die Mannschaft ein Heilmittel für diese Krankheit."* (engl. *"A couple of hours later the team found a cure for this disease"*). In spite of the similarity between the candidate and the reference translations, WCDG was not able to correctly identify all the dependencies in the candidate translation. As a result, *paar* (engl. *couple*) was attached to the *Root*, while the noun *Stunden* (engl. *hours*) was identified to be an apposition to *paar*. Moreover, the preposition *für* (engl. *for*) was attached to *paar* as a prepositional phrase. In contrast, WCDG$_{\text{predictor}}$ correctly identifies the dependencies guided by predictions. This indicates that biased parsing is helpful even in the case of modeling errors. Therefore, *paar* is attached to *Stunden* as an attribute, *später* (engl. *later*) becomes the head of *Stunden* and the preposition *für* is correctly attached to the noun *Heilmittel* (engl. *remedy*). Like in the previous example, the corrected dependencies influence the average HWCM score, which increased from 0.23 to 0.27.

6 Conclusions and future work

The accuracy of external processing tools is essential for the syntactic evaluation of MT output. In this paper, we introduced a new method to bias the parsing of candidate translations in order to improve the quality of syntactic metrics. The method was tested on the English-German translation pair using HWCM (Liu and Gildea, 2005). Even though the results showed only slight improvements of the correlation scores, it is important to note that the method is independent of the syntactic metric used. The method is also language independent, as long as a constraint based dependency parser for the target language is available. In future work we plan to improve the performance of the predictor component by filtering the predictions, so that only the high quality ones are preserved. Furthermore, we aim to investigate if a more complex alignment tool improves our method. In order to achieve this, a rule based preprocessing of the candidate and reference translations could be used. As the quality of the predictions is influenced by the quality of the reference translation parse, we will examine how parsing errors influence the correlation scores.

Acknowledgments

This work was funded by a University of Hamburg Doctoral Fellowship.

References

Satanjeev Banerjee and Alon Lavie. 2005. METEOR: An Automatic Metric for MT Evaluation with improved correlation with human judgments. *Proceedings of the Workshop on Intrinsic and Extrinsic Evaluation Measures for MT and/or Summarization.*

Ondřej Bojar, Christian Buck, Chris Callison-Burch, Christian Federmann, Barry Haddow, Philipp Koehn, Christof Monz, Matt Post, Radu Soricut, and Lucia Specia. 2013. Findings of the 2013 Workshop on Statistical Machine Translation. *Proceedings of the Eighth Workshop on Statistical Machine Translation.*

Michael Denkowski and Alon Lavie. 2010. Choosing the Right Evaluation for Machine Translation: an Examination of Annotator and Automatic Metric performance on Human Judgment Tasks. *Proceedings of the Ninth Conference of the Association for Machine Translation in the Americas.*

Michael Denkowski and Alon Lavie. 2014. Meteor Universal: Language Specific Translation Evaluation for Any Target Language. *Proceedings of the EACL 2014 Workshop on Statistical Machine Translation.*

Kilian Foth. 2004. Writing weighted constraints for large dependency grammars. *Workshop Recent Advances in Dependency Grammar, COLING 2004.*

Kilian Foth, Michael Daum, and Wolfgang Menzel. 2004a. A broad-coverage parser for German based on defeasible constraints. *KONVENS 2004, Beiträge zur 7 Konferenz zur Verarbeitung natürlicher Sprache.*

Kilian Foth, Michael Daum, and Wolfgang Menzel. 2004b. Interactive grammar development with WCDG. *Proceedings of the 42nd Annual Meeting of the Association for Computational Linguistics.*

Kilian Foth, Tomas By, and Wolfgang Menzel. 2006. Guiding a constraint dependency parser with supertags. *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics.*

Kilian Foth and Wolfgang Menzel. 2006. Hybrid Parsing: Using Probabilistic Models as Predictors for a Symbolic Parser. *Proceedings of the 44th Annual Meeting of the Association for Computational Linguistics.*

Kuzman Ganchev, Jennifer Gillenwater, and Ben Taskar. 2009. Dependency Grammar Induction via Bitext Projection Constraints. *Proceedings of the 47th Annual Meeting of the Association for Computational Linguistics.*

Spence Green, Daniel Cer, and Christopher Manning. 2014. Phrasal: A Toolkit for New Directions in Statistical Machine Translation. *Proceedings of the Ninth Workshop on Statistical Machine Translation.*

Birgit Hamp and Helmut Feldweg. 1997. GermaNet - a Lexical-Semantic Net for German. *Workshop Automatic Information Extraction and Building of Lexical Semantic Resources for NLP Applications.*

Liang Huang, Wenbin Jiang, and Qun Liu. 2009. Bilingually-Constrained (Monolingual) Shift-Reduce Parsing. *Conference on Empirical Methods in Natural Language Processing EMNLP 2009.*

Wenbin Jiang and Qun Liu. 2010. Dependency Parsing and Projection Based on Word-Pair Classification. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics.*

Lidia Khmylko, Kilian A. Foth and Wolfgang Menzel. 2009. Co-Parsing with Competitive Models. *Proceedings of the 11th International Conference on Parsing Technologies.*

Ding Liu and Daniel Gildea. 2005. Syntactic Features for Evaluation of Machine Translation. *Workshop on Intrinsic and Extrinsic Evaluation Measures for Machine Translation and/or Summarization.*

Chin-Yew Lin and Franz J Och. 2004. ORANGE: A Method for Evaluating Automatic Evaluation Metrics for Machine Translation. *Proceedings of the 20th International Conference on Computational Linguistics COLING 2004.*

Matouš Macháček and Ondřej Bojar. 2013. Results of the WMT13 Metrics Shared Task. *Proceedings of the Eighth Workshop on Statistical Machine Translation.*

Patrick McCrae. 2007. Integrating cross-modal context for PP attachment disambiguation. *Proceedings of the 3rd International Conference on Natural Computation ICNC 2007.*

Ryan McDonald, Kevin Lerman, and Fernando Pereira. 2006. Multilingual Dependency Parsing with a Two-Stage Discriminative Parser. *Tenth Conference on Computational Natural Language Learning.*

Wolfgang Menzel and Ingo Schröder. 1998. Decision Procedures for Dependency Parsing Using Graded Constraints. *Workshop On Processing Of Dependency-Based Grammars.*

Karolina Owczarzak, Joseph van Genabith, and Andy Way. 2007. Labeled dependencies in Machine Translation evaluation. *Proceedings of the Second Workshop on Statistical Machine Translation.*

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics.*

Maja Popović. 2011. Hjerson: An Open Source Tool for Automatic Error Classification of Machine Translation Output. *Prague Bull. Math. Linguistics, 96, 59-68.*

Rudolf Rosa, Ondřej Dušek, David Mareček, and Martin Popel. 2012. Using Parallel Features in Parsing of Machine-Translated Sentences for Correction of Grammatical Errors. *Proceedings of SSST-6, Sixth Workshop on Syntax, Semantics and Structure in Statistical Translation.*

Ingo Schröder, Wolfgang Menzel, Kilian Foth, and Michael Schulz. 2000. Modeling dependency grammar with restricted constraints. *Traitement automatique des langues (T. A. L.).*

Ingo Schröder, Horia Pop, Wolfgang Menzel, and Kilian Foth. 2001. Learning grammar weights using genetic algorithms. *Proceedings of the Euroconference Recent Advances in Natural Language Processing.*

Ingo Schröder. 2002. Natural Language Parsing with Graded Constraints. *Ph.D. thesis, Dept. of Computer Science, University of Hamburg.*

Antonio Toral, Sudip Kumar Naskar, Federico Gaspari, and Declan Groves. 2012. DELiC4MT: A Tool for Diagnostic MT Evaluation over User-defined Linguistic Phenomena. *Prague Bull. Math. Linguistics, Vol. 98, 121-132.*

Daniel Zeman, Mark Fishel, Jan Berka, and Ondřej Bojar. 2011. Addicter: What Is Wrong with My Translations?. *Prague Bull. Math. Linguistics, Vol. 96, 79-88.*

Ming Zhou, Bo Wang, Shujie Liu, Mu Li, Dongdong Zhang, and Tiejun Zhao. 2008. Diagnostic evaluation of machine translation systems using automatically constructed linguistic checkpoints. *Proceedings of the 22nd International Conference on Computational Linguistics COLING 2008.*

A Language Model Sensitive to Discourse Context*

Tae-Gil Noh
Institute für Computerlinguistik
Heidelberg University
69120 Heidelberg, Germany
noh@cl.uni-heidelberg.de

Sebastian Padó
Institute für Maschinelle Sprachverarbeitung
University of Stuttgart
70569 Stuttgart, Germany
pado@ims.uni-stuttgart.de

Abstract

The paper proposes a meta language model that can dynamically incorporate the influence of wider discourse context. The model provides a conditional probability in forms of $P(\text{text}|\text{context})$, where the context can be arbitrary length of text, and is used to influence the probability distribution over documents. A preliminary evaluation using a 3-gram model as the base language model shows significant reductions in perplexity by incorporating discourse context.

1 Introduction

Language models (LMs) are designed to distinguish likely from unlikely texts, for example judging that P("I broke my leg") is more likely than P("I ate my leg"). This type of prediction helps various tasks like Speech Recognition and Machine Translation (Pieraccini, 2012; Koehn, 2010).

The most prominent family of LMs in widespread use today is the family of *n-gram models* (Manning and Schütze, 1999; Zweig and Burges, 2012) which model the probability of a word as its conditional probability given the n-1 preceding words, $P(x_n|x_1, \ldots, x_{n-1})$. This assumption makes estimation of the model parameters easy, but the resulting models cannot take into account broader discourse context. For example, consider the two sentences "I broke my hand." and "I broke my promise." According to a standard LM (Brants and Franz, 2006), both are about equally likely to appear in written text. However, if the previous sentence was "I fell from a ladder." a human reader can easily predict that "I broke my hand" is much more likely to follow than "I broke my promise". This cannnot be accounted for straightforwardly within n-gram language models since it would involve raising n to high values.

The method in this paper dynamically incorporates the influence of wider discourse context into a LM which we call the *Conditioned Language Model (CLM)*. It models the influence of context by defining a conditional probability distribution in the form of $P(text|context)$, where both texts and context can be word sequences of arbitrary length. The model builds on the observation that not all documents in a large corpus are equally relevant for a given $text$. Inspired by the use of LMs in Information Retrieval (Manning et al., 2008), we assign weights to corpus documents based on the $context$, in effect giving documents which make the $context$ more likely a higher weight in the scoring of the $text$. For example, using the $context$ "fell from a ladder" would assign higher weight to documents about household accidents and lead to higher probabilities for $text$s like "broke my hand".

The CLM is not a standalone language model, but a meta-model similar to smoothing or domain adaptation methods. It can be applied to any base language model appropriate for LM-based IR. We present an efficient implementation of the CLM and a pilot evaluation on a news corpus with an underlying trigram LM. We find that the CLM can use discourse context to improve predictions for sentences in unseen documents, significantly reducing per-word perplexity compared to the base LM, with the highest reductions for small (i.e., specific) contexts.

2 The Model

2.1 The Query Likelihood Model

Our Conditioned Language Model builds on *document-based language models* as commonly used in Information Retrieval, such as the *query likelihood model* (Ponte and Croft, 1998; Miller et al., 1999; Manning et al., 2008). The query likelihood model constructs a LM M_{d_i} from each document d_i in a corpus. The model scores each document d_i relative to a query q formulated as a set of terms $\{t_1 t_2 \ldots t_k\}$ by the conditional probability $P(d_i|q)$ which can be written as

$$P(d_i|q) = P(q|d_i)P(d_i)/P(q) \qquad (1)$$

Since $P(q)$ is fixed for a given query and $P(d)$ is often set to the uniform distribution, it is sufficient to optimize $P(q|d_i)$, the probability that a query q would be drawn by random sampling from the document d_i. It is generally computed by assuming that the query decomposes into a sequence of smaller units (terms or n-grams u_k) which can be assumed to be conditionally independent of one another given the document:

$$P(q|d_i) = P(u_1 \ldots u_k|d_i) = \prod_k P(u_k|d_i) \qquad (2)$$

Finally, the probability of each unit given a document is generally defined as an interpolation of the collection LM and the document LM:

$$P(u_k|d_i) = \lambda P_{M_{d_i}}(u_k) + (1-\lambda)P_{M_C}(u_k) \qquad (3)$$

where M_C is a LM trained on the whole collection, while M_{d_i} is a LM just for d_i. This interpolation counteracts sparsity, ensuring that all $P(u_k|d_i)$ are defined over the same events and assign some probability to units even if they do not appear in d_i.

2.2 The Conditioned Language Model

Our Conditioned Language Model (CLM) extends the Query Likelihood Model in a manner that is fairly straightforward when the models are visualized as generative processes, as shown in Figure 1. In the query likelihood model (shown on the left), the document generates the query; in the conditioned language model (on the right-hand side), the document generates both the text and its context.

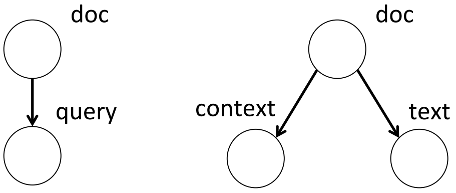


Figure 1: The query likelihood model (left) and the conditioned language model (right)

We assume that context and text are generated using the same process, defined in Eq. (2). The important extension of the CLM is that it allows us to define a conditional probability for a text t given some context c, $P(t|c)$. By marginalizing over documents, it can be defined as:

$$P(t|c) = \frac{P(c,t)}{P(c)} = \frac{\sum_i P(c,t,d_i)}{\sum_i P(d_i,c)} \qquad (4)$$

$$= \frac{\sum_i P(d_i)P(c|d_i)P(t|d_i)}{\sum_i P(d_i)P(c|d_i)} \qquad (5)$$

Assuming a uniform prior over documents yields:

$$P(t|c) = \sum_i \left(\frac{P(c|d_i)}{\sum_j P(c|d_j)} P(t|d_i) \right) \qquad (6)$$

$$= \sum_i P(d_i|c)P(t|d_i) \qquad (7)$$

The step from Eq. (6) to Eq. (7) involves an application of Bayes' rule as well as the assumption of a uniform prior over the documents.

Eq. (7) can be used to illustrate the relationship between the traditional LM and the CLM. In a traditional n-gram based LM, the only straightforward ways to incorporate information akin to our context would be to concatenate text and context into a combined query $c + t$. Due to the independence assumptions of the LM , $P(c + t) = P(c)P(t)$.[1] Thus, text and context are independent of each other.

This is fundamentally different in the CLM where text and context are generally *not* independent. If occurrences of t and c are associated (i.e. there are many documents that can generate both c and t), then $P(c,t) > P(c)P(t)$. Conversely, if they are unlikely to be observed together, $P(c,t) < P(c)P(t)$. The reason for this behavior is that even though the model assumes that context

[1] This is true provided that there are no units of the type $\ldots c_m$ `</s> <s>` $t_1 \ldots$ "cutting across" the boundary between c and t. We believe that this is a reasonable assumption.

and text are generated independently given the document (as shown in Figure 1), knowing the context can be understood to update the document distribution so that it is non-uniform and conditioned on the context ($P(d_i|c)$). In this way, the CLM assigns to every text t (given c) the probability that the text is generated from a document, where the contribution of the document is weighted by its probability given the context c.

This results in a dynamic LM which can be interesting for a range of applications, by encoding previous knowledge into the context variable. Examples include lexical substitution tasks (McCarthy and Navigli, 2009) or sentence completion tasks that have been specifically articulated as challenges for LM (Zweig and Burges, 2012). The example that we used in the introduction can also be phrased as such a problem: *I fell from a ladder and broke my **hand / promise / heart**.*

3 Efficiency Considerations

Querying the CLM for a text t involves calling every document LM to compute $P(t|d_i)$, which is potentially expensive. To improve efficiency, we can take advantage of the fact that $P(t|d_i)$ is typically very non-uniformly distributed: only a very small number of documents are highly relevant for a given text. To assess this effect, we have experimented with retrieving just the top N documents. We index all documents with the Apache Solr[2] search engine and retrieve the first N documents returned by a Boolean search for the query t.[3] We set the document-based probability term $P_{M_{d_i}}$ from $P(t|d_i)$ from Eq. (3) to 0 for all documents that are not returned. This cuts off the “long tail” of the document-based distribution part of $P(t|d_i)$. We find that setting N to 10,000 typically captures 99% of the total probability mass of $P(t|d_i)$ and yields quasi-optimal performance.

Calculating $P(t|d_i)$ for large Ns of documents (e.g., 10,000) seems like a serious time complexity overhead. However, it is not necessary to actually call the document LMs N times. In fact, the Query Likelihood model is generally used to produce a document ranking, for which task it also needs to compute $P(query|d_i)$ for all d_i. Implementations solve this task efficiently by keeping inverted indices that not only record document IDs, but also the probability of n-grams for each document model. Such index structures can be very large, but provide near real-time calculations of $P(query|d_i)$ on large document sets. The same strategies can be used to compute the two terms comprising the CLM (cf. Eq. (7)), with just a constant overhead (computing two terms instead of one for each document plus a weighted sum).

4 Experimental Setup

We presents a pilot evaluation using per-word perplexity, a standard task-independent proxy for improvements in language modeling. Perplexity be understood as the amount of information necessary to encode the text, with lower numbers indicating better models.

Language Model. We construct our base LM from the 1.6M AFP news articles (700M words) from English Gigaword corpus (Parker et al., 2011) using SRILM (Stolcke, 2002). The collection model M_C is trained one the complete corpus. Document n-gram LMs M_{d_i} are generated from each document. All models are trained using a standard setup: trigrams with Katz back-off (Katz, 1987) and Good-Turing smoothing (Gale and Sampson, 1995). The CLM is implemented as described in Eq. (6) and Section 3.

Baselines. We consider two baselines. The first one is the collection model P_{M_C} which does not use any document models. $P_{\text{CLM}}(s_n|\emptyset)$ is the CLM without context. This model corresponds to the Query Likelihood Model (Eq. (3)), assuming a uniform distribution over documents.

Test and validation data. We use a set of 50 news articles from APW February 2010 for the optimization of the interpolation parameter λ. The final test evaluation takes place on the unseen first 500 news articles of Gigaword APW January 2010 subcorpus (11K sentences, 220K words).

5 Experimental Results

5.1 Parameter Optimization

The CLM has one free parameter, namey λ (Eq. (3)), the interpolation ratio between the document models and the collection model. Before

[2] `http://lucene.apache.org/solr/`

[3] The documents are ranked by the number of matching terms. Ties are broken randomly.

λ	0.1	0.2	0.3	0.4	0.5
$P_{\text{CLM}(s_n\|\emptyset)}$	149.6	**147.9**	150.2	155.9	165.0

Table 1: Parameter optimization: Per-word perplexity on the validation set for various values of λ.

	Model (with $\lambda = 0.2$)	Perplexity (% gain over $P_{\text{CLM}}(s_n\|\emptyset)$)
BLs	P_{M_C} (collection model)	154.429
	$P_{\text{CLM}}(s_n\|\emptyset)$	135.453
Experiment	$P_{\text{CLM}}(s_n\|s_{n-1})$	125.330 (7.47%)
	$P_{\text{CLM}}(s_n\|s_{n-2}s_{n-1})$	125.214 (7.55%)
	$P_{\text{CLM}}(s_n\|s_{n-3}\dots s_{n-1})$	124.098 (8.38%)
	$P_{\text{CLM}}(s_n\|s_{n-4}\dots s_{n-1})$	126.750 (6.42%)
	$P_{\text{CLM}}(s_n\|s_1\dots s_{n-1})$	130.426 (3.71%)
	$P_{\text{CLM}}(s_n\|s_{\text{title}})$	130.734 (3.48%)
UBs	$P_{\text{CLM}}(s_n\|s_{n-1}s_n)$	93.496 (30.97%)
	$P_{\text{CLM}}(s_n\|s_{n-2}\dots s_n)$	100.722 (25.64%)
	$P_{\text{CLM}}(s_n\|s_{n-3}\dots s_n)$	106.559 (21.33%)

Table 2: Per-word perplexity (sentences as targets): baselines (BLs), experiment, upper bounds (UBs)

proceeding to the final evaluation, we optimize λ on our validation set. Since we assume that the document models are fairly sparse and high values of λ correspond to document model dominance, we only consider λ values between 0.1 and 0.5.

Table 1 shows the perplexities of the baseline CLM model without context ($P_{\text{CLM}}(s_n|\emptyset)$) for various values of λ. The selection of λ heavily affects the model, with generally better perplexity for lower values of λ. This matches our intuition: we need to strongly smooth the document models with the collection model. However, the document models are informative after all. We achieve the highest reduction in perplexity for $\lambda = 0.2$. We use this value for the remainder of the experiments.

5.2 Main Evaluation

The results of our main experiment are shown in Table 2, which consists of three parts. The top part of Table 2 shows the two baselines. Note that the CLM without context ($P_{\text{CLM}}(s_n|\emptyset)$) already performs substantially better than the collection model P_{M_C}. $P_{\text{CLM}}(s_n|\emptyset)$ is essentially the average probability of generating s_n in the query likelihood model. It already takes benefit of document-level statistics in addition to collection-level statistics, which results in better estimation. Correspondingly, we adopt $P_{\text{CLM}}(s_n|\emptyset)$ as point of reference for all comparisons concerning the effect of context. All gains reported in the table are relative to this model.

The middle part of Table 2 shows the results for various settings of the Conditioned Language Model. We estimate the probability of individual target sentences, comparing various definitions of context as conditioning events as to their effectiveness in predicting the target. More specifically, we consider sentence windows of one to four previous sentences before the target text as well as longer discourse context, such as all preceding sentences in the document or the document tiele. For example, for each sentence s_n, the model $P(s_n|s_{n-1}s_{n-2})$ uses the two previous sentences, s_{n-1} and s_{n-2}, together as context.

All CLM models with context improve over the base model $P_{\text{CLM}}(s_n|\emptyset)$. Significance testing with bootstrap resampling (Efron and Tibshirani, 1993) showed that all performance gains are significant (all models: p<0.001). The best context among the evaluated models is a three-sentence window before the target sentence, which reduces the per-word perplexity by 8.38% compared to the null context CLM, a reduction of 19.64% compared to the collection model P_{M_C}. Both two-sentence and four-sentence models do clearly worse. It appears that the three-sentence window strikes the best balance between providing a rich context and diluting the local information too much. In comparison, wider discourse context performs much worse: the two CLM versions that take the complete prior context or the document title into account only obtain complexity reductions of between 3% and 4%. Our interpretation is that the CLM is able to pick up a modest amount of discourse coherence in terms of lexical distributions that slowly changes over the course of a document.

The bottom part of Table 2 aims at establishing an upper bound for the perplexity improvements that can be expected from the CLM by including the target sentence into the context. For example, the model $P_{\text{CLM}}(s_n|s_{n-1}s_n)$ uses the target sentence itself and its previous sentence as the context. Our rationale comes from the application of the CLM to tasks like sentence completion (Section 2.2). This involves a research question in its own right, namely defining which part of the problems should serve as the context and which as the text. While the split can simply be made along phrase boundaries ($P($broke my hand | I fell from a ladder and$)$), we believe that better results can be obtained if some parts of the problem are included in both t and c. For example, $P($broke my hand | I fell from a ladder and broke$)$ asks the model simultaneously to focus on documents that talk both about ladders and about breaking. In general, it seems a good idea to make as rich as possible both the context (for good document selection) and the text (for good plausibility estimation). Our "upper bound" models show the limit of this approach when the text is a proper subset of the context.

The results show that in this setup, sentence-window CLMs reduce perplexity greatly. The best model does so by 30.97%. It is the one-sentence window CLM, which is expected since larger contexts dilute the target sentence information.

6 Related Work

In n-gram LMs, more context can be integrated by simply increasing n. While the resulting complexity and efficiency issues can be addressed (Talbot and Osborne, 2007; Wood et al., 2009), it is difficult to go beyond n=5 even with trillions of words (Brants and Franz, 2006).

The CLM can be regarded as a type of adaptive LM. Adaptive LMs generally construct full-fledged models from specific datasets such as domains (Rosenfeld, 1996; Lin et al., 2011; Shi et al., 2012), LDA-style topics (Hsu and Glass, 2006; Trnka, 2008), or occurrences of individual words (Sicilia-Garcia et al., 2000). Once generated, the models are combined based on their match with the test topic or domain. Among such domain adaptation approaches, *training data selection* (Moore and Lewis, 2010; Axelrod et al., 2011) is most related to our work. It focuses on a small part of the training corpus particularly similar to the test domain. This is mirrored in the CLM's use of $P(d_i|c)$ to weigh documents based on context.

The two main differences are: (1), the selection is not made on the basis of a corpus, but of a relatively small context; (2), our CLM is more dynamic: the weighting is not given at training time, but by specifying a context at test time.

Other previous studies have explicitly introduced novel modeling strategies to incorporate long distance dependencies such as caching (Iyer and Ostendorf, 1999), triggering events (Rosenfeld, 1996), or neural networks (Schwenk, 2007; Mikolov and Zweig, 2012). Compared to these approaches, the CLM has two advantages: (a) it can be seen as a wrapper around standard LMs and can thus take advantage of all previous research; (b) it supports a wide range of context definitions, while previous work hard-coded context types.

7 Conclusion

This paper presents the Conditioned Language Model, a meta language model which can incorporate discourse context or previous knowledge. It models $P(\text{text}|\text{context})$, where both text and context can be arbitrary word sequences. We have described an approximation to make computation feasible for large document collections, and our preliminary evaluation shows that a small window context helps predicting target sentences, reducing per-word perplexity by 8.4% compared to the model without context. We interpret this as encouragement that the CLM can providing judgments about the likelihood of texts that incorporate discourse information in a natural and general manner, going beyond the capabilities of traditional n-gram LMs.

Our next steps will address more thorough evaluation of the model. It can replace LMs used in applications like MT or ASR. However, what we feel to be more promising is the use of CLM's conditional probabilities for "semantic" NLP tasks such as lexical substitution or cloze completion (cf. Section 2.2). Much work on such tasks is based on lexical association measures at the word level such as pointwise mutual information. The CLM can be understood as a natural generalization, namely an association measure at the sentence level, based on document distributions.

Acknowledgments.

We gratefully acknowledge partial funding by the European Commission (project EXCITEMENT (FP7 ICT-287923)).

References

Amittai Axelrod, Xiaodong He, and Jianfeng Gao. 2011. Domain adaptation via pseudo in-domain data selection. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 355–362, Edinburgh, United Kingdom.

Thorsten Brants and Alex Franz. 2006. Web 1t 5-gram version 1. Technical report, Linguistic Data Consortium, Philadelphia.

Bradley Efron and Robert J. Tibshirani. 1993. *An Introduction to the Bootstrap*. Chapman and Hall, New York.

William A Gale and Geoffrey Sampson. 1995. Good-turing frequency estimation without tears. *Journal of Quantitative Linguistics*, 2(3):217–237.

Bo-June (Paul) Hsu and James Glass. 2006. Style & topic language model adaptation using hmm-lda. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 373–381, Sydney, Australia.

Rukmini M. Iyer and Mari Ostendorf. 1999. Modeling long distance dependence in language: Topic mixtures versus dynamic cache models. *IEEE Transactions on Speech and Audio Processing*, 7(1):30–39.

Slava Katz. 1987. Estimation of probabilities from sparse data for the language model component of a speech recognizer. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 35(3):400–401.

Philipp Koehn. 2010. *Statistical Machine Translation*. Cambridge University Press.

Jimmy Lin, Rion Snow, and William Morgan. 2011. Smoothing techniques for adaptive online language models: Topic tracking in tweet streams. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 422–429, San Diego, CA.

Christopher D Manning and Hinrich Schütze. 1999. *Foundations of Statistical Natural Language Processing*. The MIT Press.

Christopher D. Manning, Prabhakar Raghavan, and Hinrich Schütze. 2008. *Introduction to information retrieval*. Cambridge University Press.

Diana McCarthy and Roberto Navigli. 2009. The english lexical substitution task. *Language Resources and Evaluation*, 43(2):139–159.

Tomas Mikolov and Geoffrey Zweig. 2012. Context dependent recurrent neural network language model. In *Proceedings of the IEEE Spoken Language Technology Workshop*, pages 234–239, Miami, FL, USA.

David R. H. Miller, Tim Leek, and Richard M. Schwartz. 1999. A hidden markov model information retrieval system. In *Proceedings of the 22Nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 214–221, Berkeley, CA.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224, Uppsala, Sweden.

Robert Parker, David Graff, Junbo Kong, Ke Chen, and Kazuaki Maeda. 2011. English gigaword fifth edition. Technical report, Linguistic Data Consortium, Philadelphia.

Roberto Pieraccini. 2012. *The Voice in the Machine: Building Computers That Understand Speech*. The MIT Press.

Jay M. Ponte and W. Bruce Croft. 1998. A language modeling approach to information retrieval. In *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, pages 275–281.

Ronald Rosenfeld. 1996. A maximum entropy approach to adaptive statistical language modelling. *Computer Speech & Language*, 10(3):187–228.

Holger Schwenk. 2007. Continuous space language models. *Computer Speech and Language*, 21(3):492–518.

Yangyang Shi, Pascal Wiggers, and Catholijn M Jonker. 2012. Adaptive language modeling with a set of domain dependent models. In *Proceedings of Text, Speech and Dialogue*, pages 472–479, Brno, Czech Republic.

E. I. Sicilia-Garcia, Ji Ming, and F. J. Smith. 2000. A dynamic language model based on individual word domains. In *Proceedings of the 18th Conference on Computational Linguistics*, pages 789–794, Saarbrücken, Germany.

Andreas Stolcke. 2002. Srilm-an extensible language modeling toolkit. In *Proceedings of 7th International Conference on Spoken Language Processing*, pages 1045–1048, Denver, CO.

David Talbot and Miles Osborne. 2007. Smoothed bloom filter language models: Tera-scale LMs on the cheap. In *Proceedings of EMNLP*, pages 468–476, Prague, Czech Republic.

Keith Trnka. 2008. Adaptive language modeling for word prediction. In *Proceedings of the 46th Annual Meeting of the Association for Computational Linguistics on Human Language Technologies: Student Research Workshop*, pages 61–66, Columbus, Ohio.

Frank Wood, Cédric Archambeau, Jan Gasthaus, Lancelot James, and Yee Whye Teh. 2009. A stochastic memoizer for sequence data. In *Proceedings of the 26th Annual International Conference on Machine Learning*, pages 1129–1136, Montreal, Canada.

Geoffrey Zweig and Chris J. C. Burges. 2012. A challenge set for advancing language modeling. In *Proceedings of the NAACL-HLT 2012 Workshop on the Future of Language Modeling for HLT*, pages 29–36, Montreal, Canada.

German Perception Verbs: Automatic Classification of Prototypical and Multiple Non-literal Meanings

Benjamin David, Sylvia Springorum, Sabine Schulte im Walde
Institut für Maschinelle Sprachverarbeitung
Universität Stuttgart
{benjamin.david,sylvia.springorum,schulte}@ims.uni-stuttgart.de

Abstract

This paper presents a token-based automatic classification of German perception verbs into literal vs. multiple non-literal senses. Based on a corpus-based dataset of German perception verbs and their systematic meaning shifts, we identify one verb of each of the four perception classes *optical, acoustic, olfactory, haptic*, and use Decision Trees relying on syntactic and semantic corpus-based features to classify the verb uses into 3-4 senses each. Our classifier reaches accuracies between 45.5% and 69.4%, in comparison to baselines between 27.5% and 39.0%. In three out of four cases analyzed our classifier's accuracy is significantly higher than the according baseline.

1 Introduction

In contrast to Word Sense Disambiguation in general (cf. Agirre and Edmonds (2006); Navigli (2009)), most *computational approaches to modelling literal vs. non-literal meaning* are still restricted to a binary distinction between two sense categories (literal vs. non-literal), rather than between multiple literal and non-literal senses. For example,[1] Bannard (2007), and Fazly et al. (2009) identified light verb constructions as non-literal verb uses; Birke and Sarkar (2006), Birke and Sarkar (2007), Sporleder and Li (2009), and Li and Sporleder (2009) distinguished literal vs. idiomatic meaning. Concerning metonymic language, most approaches address various senses, which are however very restricted to two domains, locations and organizations (Markert and Nissim, 2002; Nastase and Strube, 2009; Nastase et al., 2012). One of the few studies going beyond a binary classification is represented by Shutova et al. (2013) who classified literal vs. metaphorical verb senses on a large scale and for multiple non-literal meanings. Cook and Stevenson (2006) also took multiple sense distinctions into account, focusing on English 'up' particle verbs.

In this paper,[2] we address the automatic classification of German perception verbs into literal vs. non-literal meanings. Our research goes beyond a binary classification and distinguishes between multiple non-literal senses. Taking the PhD thesis by Ibarretxe-Antunano (1999) as a starting point, a preparatory step places German perception verbs into four classes: optical, acoustic, olfactory and haptic. In the main part, we then choose one perception verb from each class ('betrachten', 'hören', 'wittern', 'spüren')[3] which each have multiple literal/non-literal senses, and rely on syntactic and semantic corpus-based features and a Decision Tree classifier to perform a token-based assignment to senses. We address both a binary (literal vs. non-literal) and a multiple sense discrimination.

The paper describes related work in Section 2, specifies the perception verbs and their features in Sections 3 and 4, and performs automatic token-based word sense classification in Section 5.

[1] See Section 2 for details on related work.

[2]

[3] Since the verbs have multiple meanings, we do not translate them here but in Section 3.

2 Related Work

Computational work on non-literal meaning comprises research from various sub-fields. Approaches to *light verb constructions* (Bannard, 2007; Fazly et al., 2007; Fazly et al., 2009) relied on measures of syntactic variation of phrases, in combination with standard association measures, to perform a type-based classification. Approaches to *literal vs. non-literal/figurative/idiomatic meaning* performed binary classifications (Birke and Sarkar, 2006; Birke and Sarkar, 2007; Sporleder and Li, 2009; Li and Sporleder, 2009), relying on various contextual indicators: Birke and Sarkar exploited seed sets of literal vs. non-literal sentences, and used distributional similarity to classify English verbs. Li and Sporleder defined two models of text cohesion to classify V+NP and V+PP combinations. All four approaches were token-based.

Approaches to metaphoric language predominantly focus on binary classification. The most prominent research has been carried out by Shutova, best summarized in Shutova et al. (2013). Shutova performed both metaphor identification and interpretation, focusing on English verbs. She relied on a seed set of annotated metaphors and standard verb and noun clustering, to classify literal vs. metaphorical verb senses. Gedigian et al. (2006) also predicted metaphorical meanings of English verb tokens, relying on manual rather than unsupervised data, and a maximum entropy classifier. Turney et al. (2011) assume that metaphorical word usage is correlated with the abstractness of a word's context, and classified word senses in a given context as either literal or metaphorical. Their targets were adjective-noun combinations and verbs.

Approaches to metonymic language represent a considerable development regarding features and classification approaches since 2002: Markert and Hahn (2002) proposed a rule-based ranking system exploring the contribution of selectional preferences vs. discourse and anaphoric information; Markert and Nissim (2002) presented the first supervised classifier for location names and compared window co-occurrences, collocations and grammatical features; Nissim and Markert (2005) extended the framework towards organization names and focused on grammatical features; Nastase and Strube (2009) enriched the feature set from Markert and Nastase by lexical features from the Sketch Engine (Kilgarriff et al., 2004) fed into WordNet supersenses, and by encyclopedic knowledge from Wikipedia relations; Nastase et al. (2012) used an unsupervised classifier and global context relying on the Wikipedia concept network.

3 Dataset of German Perception Verbs

In this section, we describe the creation of our dataset of German perception verbs in three steps: (i) sampling the perception verbs (Section 3.1), (ii) identification of literal and non-prototypical meanings (Section 3.2), and (iii) corpus annotation with perception senses (Section 3.3).

3.1 Sampling of Perception Verbs

As there is no available resource providing a complete list of German perception verbs, we combined the information of several dictionaries and thesauri to create such a dataset. As a starting point, we defined a base verb for each type of perception: *sehen* 'see' for optical verbs, *hören* 'hear' for acoustic verbs, *riechen* 'smell' for olfactory verbs, *tasten* 'touch' for haptic verbs and *schmecken* 'taste' for gustatory verbs. Using these verbs as starting points, all their synonyms or closely related words were determined, relying on Ballmer and Brennenstuhl (1986) and Schumacher (1986). Using the enlarged set of verbs, we again added all their synonyms and closely related words. We repeated this cycle and at the same time made sure that each additional verb belongs exclusively to the desired perception class, until no further changes occurred. The sampling process determined 54 optical, 15 acoustic, 9 olfactory, 12 haptic and one gustatory verbs.

For the classification experiments, we selected one verb from each perception class, disregarding the sole gustatory verb. The selected olfactory and haptic verbs only undergo passive perception meanings, the optical verb only undergoes active perception meanings, and the acoustic verb holds both active and passive perception meanings.[4]

[4] Active perception is controlled perception (as in 'listens to the music'); passive perception is non-controlled perception (as in 'hears faint barking').

3.2 Non-Prototypical Meanings

Analyzing the senses of the perception verbs in our dataset was carried out in accordance with *Polysemy and Metaphor in Perception Verbs: A Cross-Linguistic Study* (Ibarretxe-Antunano, 1999), which systematically determined non-prototypical meanings of perception verbs cross-linguistically for English, Spanish and Basque. For example, Ibarretxe-Antunano (1999) identified three major groups of shifted meanings for vision verbs, (i) the *Intellection group* including *to understand, to foresee, to visualize, to consider, to revise*; (ii) the *Social group* including *to meet, to visit, to receive, to go out with, to get on badly*; (iii) the *Assurance group* including *to ascertain, to make sure, to take care*. We applied her cross-lingual meaning shifts to all German perception verbs in our dataset, if possible, to identify the meanings of the perception verbs. As in Ibarretxe-Antunano (1999), the applicability was determined by corpus evidence (see below).

The following lists specify the main senses of the perception verbs that were selected for the classification experiments, with the first category in each list referring to the literal meaning.

Optical: *'betrachten'*

- to look at (lit.)
- to define/name/interpret
- to analyze objectively
- to analyze subjectively

Acoustic: *'hören'*

- to hear (lit.)
- to (dis-)like/ignore
- to obey
- to be informed

Olfactory: *'wittern'*

- to sense (by smell, lit.)
- to advance towards a goal/event
- to predict

Haptic: *'spüren'*

- to feel (lit.)
- to realize
- to feel (emotions)
- to suspect

Taking the acoustic verb 'hören' as an example, we illustrate the corpus uses of the verb by one sentence for each sense.

- to hear (lit.):
  'Er *hörte* die Wölfe heulen.'
  He heard (lit.) the wolves howl.
- to (dis-)like/ignore:
  'Sie können es nicht mehr *hören*.'
  They don't want to hear about it anymore.
- to obey:
  'Wenn er nicht *hört*, gibt's kein Futter.'
  If he doesn't obey/listen, he doesn't get food.
- to be informed:
  'Davon habe ich noch nie *gehört*.'
  I never heard/read/etc. about that.

3.3 Annotation of Verb Senses

Based on the sense definitions, we performed a manual annotation to create a gold standard for our classification experiments: A random selection of 200 sentences for each of the four selected perception verbs was carried out, gathering 50 sentences for each meaning. As an exception, 'wittern' (olfactory) only has three prominent meanings, resulting in 150 annotated sentences. The random selection was based on a subcategorization database (Scheible et al., 2013) extracted from a parsed version (Bohnet, 2010) of the *SdeWaC* corpus (Faaß and Eckart, 2013), a web corpus containing 880 million words.

These randomly selected sentences were annotated by two native speakers of German with a linguistic background (doctoral candidates in computational linguistics). The annotators were asked to label each sentence with one of the specified meanings of the respective verb. In cases where the annotators disagreed, the first author of this paper took the final decision. Agreement and majority class baselines are shown in Table 1.

Verb	Perception	Baseline	Agreement
betrachten	optical	33.5%	63.0%
hören	acoustic	35.5%	64.5%
spüren	haptic	27.5%	75.0%
wittern	olfactory	39.0%	69.4%

Table 1: Baseline and inter-annotator agreement.

4 Syntax-Semantic Verb Features

The feature vector used to classify verb instances is split into three subsets of features: syntactic, verb-modifying and semantic features. The subsets are described in the following subsections.

4.1 Syntactic and Verb-Modifying Features

The syntactic and the verb-modifying features rely on the subcategorization database by Scheible et al. (2013). This resource is a compact but linguistically detailed database for German verb subcategorization, containing verbs extracted from the SdeWaC along with the following information:

(1) *verb information*: dependency relation of the target verb according to the TIGER annotation scheme (Brants et al., 2004; Seeker and Kuhn, 2012); verb position in the sentence; part-of-speech tag and lemma of the verb;

(2) *subcategorization information*: list of all verb complements;

(3) *applied linguistic rule* that was used to extract the verb and subcategorization information from the dependency parses;

(4) *whole sentence*.

Based on the database information, we defined the following features:

Syntactic features:

- ***Sentence Rule***: Rule to extract the verb and subcategorization information; relies on the verb form and the dependency constellation of the verb.
- ***Sentence Form***: Dependency relations of the verb complex according to TIGER.
- ***Adjective***: Presence of an adjective represented by a Boolean value.
- ***Accusative Object***: Presence of an accusative object represented by a Boolean value.
- ***Subjunction***: Either ”none” or the lemma of the subjunction if available.
- ***Modal Verb***: Either ”none” or the lemma of the modal verb if available.
- ***Negation***: Presence of a negation represented by a Boolean value.

Verb-modifying features:

- ***Verb Form***: Part-of-speech tag.
- ***Adverb***: Presence of an adverb represented by a Boolean value.
- ***Adverbial or Prepositional Object***: A Boolean value for each preposition introducing a prepositional object.

4.2 Semantic Features

The semantic features rely on two different resources, GermaNet and German Polarity Clues.
(1) Information on hypernymy is extracted from ***GermaNet***, which has been modelled along the lines of the Princeton WordNet for English (Miller et al., 1990; Fellbaum, 1998) and shares its general design principles (Hamp and Feldweg, 1997; Kunze and Wagner, 1999). Lexical units denoting the same concept are grouped into synonym sets (‘synsets’), which are interlinked via conceptual-semantic relations (such as hypernymy) and lexical relations (such as antonymy).

GermaNet provides up to 20 hypernym levels. We used the most common concepts from the 3rd level (counted down from the unique top level):

- Texture
- Situation
- Quality
- Cognitive Object
- Common Object
- Pronouns (added to the original net)
- None available (added to the original net)

(2) Information on adverb and adjective sentiment is extracted from the ***German Polarity Clues*** (Waltinger, 2010), which labels adjectives and adverbs as ”positive”, ”negative” or ”neutral”.

We extracted the following semantic features:

Semantic features:

- ***Subject Hypernym***: Hypernym of the subject.
- ***Object Hypernym***: Hypernym of the direct accusative object.
- ***Adverb/Adjective Sentiment***: Either ”none” if no adverbs or adjectives are available; or the adverb/adjective sentiment label.

5 Classification

Our classification experiments were performed with WEKA. The classifier algorithm used is *J48*, a Java reimplementation of the *C4.5* algorithm (Quinlan, 1993). For training and testing, ten-fold cross-validation was applied.

The classification experiments were done separately for each perception type, i.e. for each verb. Table 2 lists the classification results for the verb *hören*,[5] distinguishing between the subsets of syntactic, verb-modifying and semantic features as well as the results for the combined vector. *Instances* refers to the number of sentences for the respective meaning. *Fraction* shows the proportion of instances of one meaning in relation to all classified instances for the respective verb. *Classifier accuracy* shows the proportion of instances which have been correctly classified by our classifier; significance according to chi-square is marked, if applicable. *Annotator agreement* is the proportion of instances in which the two annotators chose the same meaning.

6 Discussion

In the following, we provide qualitative analyses and discussions regarding our classifications.

6.1 Features

For the optical perception verb *sehen* and the acoustic perception verb *hören*, the verb-modifying and the semantic subset of features, as well as the combined set of all features, significantly beat the baselines (33.5% and 35.5%, respectively). The two subsets are equally successful at classifying optical and acoustic verb instances, reaching between 52.5% and 55.5%.

For the haptic perception verb *spüren*, each of the subset vectors and the overall feature vector provide results significantly better than the baseline (27.5%). The best subset vector for this verb is the syntactic one with an accuracy of 43.0%.

The olfactory perception verb *wittern* is not classified significantly better than the baseline (39.0%) by any subset or the combined set. The best subset vector for classification is the syntactic one with 43.9% accuracy.

The semantic subset vector turns out to be the overall best with an average of 47.2%. For all but the olfactory verb classification any subset of features returns higher accuracy than the baseline.

6.2 Ambiguity

The classification results and confusion matrices (see an example in Table 3) show that ambiguity is the biggest source of misclassifications. In the confusion matrices one can observe that often meaning ”A” is misinterpreted as meaning ”B”, which is in turn often misinterpreted as meaning ”A”. Interestingly, meanings confused by the classification algorithm are very similar to those confused by human annotators.

	0	1	2
0: Prototypical	28	0	26
1: Adv. towards Goal	15	1	30
2: Predict	21	0	43

Table 3: Confusion matrix for olfactory/syntactic.

6.3 Lack of Detailed Semantic Data

The hypernym data covers a very high level of abstraction. This data distinguishes between, for example, *texture* and *objects*, but it does not distinguish between, for example, *animals* and *plants*, which might have been more desirable. High levels of abstraction had to be chosen for this research project as the lower levels of abstraction would have resulted in several hundred feature values and thus most probably have run into severe sparse data problems. Future work will nevertheless address an improved identification of semantic levels of abstraction.

6.4 Literal Meaning as Residual Class

The varying results by feature subsets for a verb's prototypical instances suggest to have a closer look at their classification. The correctly classified instances increase and decrease in proportion to the correctly classified instances of all other meanings. Looking into the decision trees which result in classification as ”prototypical” instances, it turns out that the prototypical meaning shows residual class characteristics, cf. Table 4: It

[5] The results for the three other verbs are omitted for space reasons. We nevertheless include them into our discussion below. See David (2013) for further results.

	Instances	Fraction	Accuracy	Agreement
syntactic	200	100.0%	46.0%	68.5%
prototypical	71	35.5%	46.5%	59.2%
to (dis)like	11	5.5%	36.4%	81.8%
to obey	47	23.6%	70.2%	95.7%
to be informed	71	35.5%	31.0%	57.7%
verb-modifying	200	100.0%	***__53.0%__	68.5%
prototypical	71	35.5%	50.7%	59.2%
to (dis)like	11	5.5%	0.0%	81.8%
to obey	47	23.6%	51.1%	95.7%
to be informed	71	35.5%	64.8%	57.7%
semantic	200	100.0%	***__55.5%__	68.5%
prototypical	71	35.5%	40.8%	59.2%
to (dis)like	11	5.5%	0.0%	81.8%
to obey	47	23.6%	70.2%	95.7%
to be informed	71	35.5%	70.4%	57.7%
overall	200	100.0%	***__57.0%__	68.5%
prototypical	71	35.5%	39.4%	59.2%
to (dis)like	11	5.5%	18.2%	81.8%
to obey	47	23.6%	72.3%	95.7%
to be informed	71	35.5%	70.4%	57.7%

Table 2: Classification results for acoustic perception verb *hören* (baseline: 35.5%).

	optic	acoust	olfac	haptic	*avg*
baseline	19.0	35.5	32.9	25.0	28.1
annotation	84.2	59.2	92.4	92.0	82.0
syntactic	5.3	46.5	51.8	54.0	50.8
verb-mod	2.6	50.7	37.0	56.0	47.9
semantic	2.6	40.8	72.2	20.0	44.3
overall	42.1	39.4	44.4	46.0	43.0

Table 4: Prototypical meaning by subsets.

seems that only the inability to determine a non-prototypical meaning through the use of distinct features results in a classification as prototypical.

6.5 Choice of Non-literal Meanings

The classification results also depend on whether fine-grained or coarse-grained senses are used. A fine-grained sense definition would lead to less variation within a sense class but to a higher number of meanings. This in turn would require more manually annotated data to cover all meanings with enough corpus examples, therefore we decided to only use the reduced and coarse-grained sense selection. However, it is not clear where to draw the line, as there are cases where a verb can have two meanings at once in one context.

7 Conclusion

This paper presented a token-based automatic classification of German perception verbs into literal vs. multiple non-literal senses. Based on a corpus-based dataset of German perception verbs and their systematic meaning shifts, following Ibarretxe-Antunano (1999), we identified one verb of each of the four perception classes optical, acoustic, olfactory and haptic, and used Decision Trees relying on syntactic and semantic corpus-based features to classify the verb uses into 3 to 4 senses each. Our classifier reached accuracies between 45.5% and 69.4%, in comparison to baselines between 27.5% and 39.0%. In three out of four cases analysed our classifier's accuracy was significantly higher than the according baseline.

References

Eneko Agirre and Philip Edmonds, editors. 2006. *Word Sense Disambiguation: Algorithms and Applications*. Springer-Verlag. URL: http://www.wsdbook.org/.

Thomas T. Ballmer and Waltraud Brennenstuhl. 1986. *Deutsche Verben*. Gunter Narr Verlag, Tübingen.

Colin Bannard. 2007. A Measure of Syntactic Flexibility for Automatically Identifying Multiword Expressions in Corpora. In *Proceedings of the ACL*

Workshop on A Broader Perspective on Multiword Expressions, pages 1–8, Prague, Czech Republic.

Julia Birke and Anoop Sarkar. 2006. A Clustering Approach for the Nearly Unsupervised Recognition of Nonliteral Language. In *Proceedings of the 11th Conference of the European Chapter of the ACL*, pages 329–336, Trento, Italy.

Julia Birke and Anoop Sarkar. 2007. Active Learning for the Identification of Nonliteral Language. In *Proceedings of the Workshop on Computational Approaches to Figurative Language*, pages 21–28, Rochester, NY.

Bernd Bohnet. 2010. Top Accuracy and Fast Dependency Parsing is not a Contradiction. In *Proceedings of the 23rd International Conference on Computational Linguistics*, pages 89–97, Beijing, China.

Sabine Brants, Stefanie Dipper, Peter Eisenberg, Silvia Hansen-Schirra, Esther König, Wolfgang Lezius, Christian Rohrer, George Smith, and Hans Uszkoreit. 2004. TIGER: Linguistic Interpretation of a German Corpus. *Research on Language and Computation*, 2(4):597–620.

Paul Cook and Suzanne Stevenson. 2006. Classifying Particle Semantics in English Verb-Particle Constructions. In *Proceedings of the ACL/COLING Workshop on Multiword Expressions: Identifying and Exploiting Underlying Properties*, pages 45–53, Sydney, Australia.

Benjamin David. 2013. Deutsche Wahrnehmungsverben: Bedeutungsverschiebungen und deren manuelle und automatische Klassifikation. Studienarbeit. Institut für Maschinelle Sprachverarbeitung, Universität Stuttgart.

Gertrud Faaß and Kerstin Eckart. 2013. SdeWaC – a Corpus of Parsable Sentences from the Web. In *Proceedings of the International Conference of the German Society for Computational Linguistics and Language Technology*, pages 61–68, Darmstadt, Germany.

Afsaneh Fazly, Suzanne Stevenson, and Ryan North. 2007. Automatically Learning Semantic Knowledge about Multiword Predicates. *Language Resources and Evaluation*, 41:61–89.

Afsaneh Fazly, Paul Cook, and Suzanne Stevenson. 2009. Unsupervised Type and Token Identification of Idiomatic Expressions. *Computational Linguistics*, 35(1):61–103.

Christiane Fellbaum, editor. 1998. *WordNet – An Electronic Lexical Database*. Language, Speech, and Communication. MIT Press, Cambridge, MA.

Matt Gedigian, John Bryant, Srini Narayanan, and Branimir Ciric. 2006. Catching Metaphors. In *Proceedings of the 3rd Workshop on Scalable Natural Language Understanding*, pages 41–48, New York City, NY.

Birgit Hamp and Helmut Feldweg. 1997. GermaNet – a Lexical-Semantic Net for German. In *Proceedings of the ACL Workshop on Automatic Information Extraction and Building Lexical Semantic Resources for NLP Applications*, pages 9–15, Madrid, Spain.

B. Iraide Ibarretxe-Antunano. 1999. *Polysemy and Metaphor in Perception Verbs*. Ph.D. thesis, University of Edinburgh.

Adam Kilgarriff, Pavel Rychlý, Pavel Smrz, and David Tugwell. 2004. The Sketch Engine. In *Proceedings of the 11th EURALEX International Congress*, pages 105–111, Lorient, France.

Claudia Kunze and Andreas Wagner. 1999. Integrating GermaNet into EuroWordNet, a Multilingual Lexical-Semantic Database. *Sprache und Datenverarbeitung*, 23(2):5–19.

Linlin Li and Caroline Sporleder. 2009. Classifier Combination for Contextual Idiom Detection Without Labelled Data. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 315–323, Singapore.

Katja Markert and Udo Hahn. 2002. Understanding Metonymies in Discourse. *Artificial Intelligence*, 136:145–198.

Katja Markert and Malvina Nissim. 2002. Metonymy Resolution as a Classification Task. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 204–213.

George A. Miller, Richard Beckwith, Christiane Fellbaum, Derek Gross, and Katherine J. Miller. 1990. Introduction to Wordnet: An On-line Lexical Database. *International Journal of Lexicography*, 3(4):235–244.

Vivi Nastase and Michael Strube. 2009. Combining Collocations, Lexical and Encyclopedic Knowledge for Metonymy Resolution. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 910–918, Singapore.

Vivi Nastase, Alex Judea, Katja Markert, and Michael Strube. 2012. Local and Global Context for Supervised and Unsupervised Metonymy Resolution. In *Proceedings of the Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 183–193, Jeju Island, Korea.

Roberto Navigli. 2009. Word Sense Disambiguation: A Survey. *ACM Computing Surveys*, 41(2).

Malvina Nissim and Katja Markert. 2005. Learning to buy a Renault and talk to BMW: A Supervised Approach to Conventional Metonymy. In *Proceedings of the 6th International Workshop on Computational Semantics*, pages 225–235, Tilburg, The Netherlands.

Ross Quinlan. 1993. *C4.5: Programs for Machine Learning*. Morgan Kaufmann.

Silke Scheible, Sabine Schulte im Walde, Marion Weller, and Max Kisselew. 2013. A Compact but Linguistically Detailed Database for German Verb Subcategorisation relying on Dependency Parses from a Web Corpus: Tool, Guidelines and Resource. In *Proceedings of the 8th Web as Corpus Workshop*, pages 63–72, Lancaster, UK.

Helmut Schumacher. 1986. *Verben in Feldern*. de Gruyter, Berlin.

Wolfgang Seeker and Jonas Kuhn. 2012. Making Ellipses Explicit in Dependency Conversion for a German Treebank. In *Proceedings of the 8th International Conference on Language Resources and Evaluation*, pages 3132–3139, Istanbul, Turkey.

Ekaterina Shutova, Simone Teufel, and Anna Korhonen. 2013. Statistical Metaphor Processing. *Computational Linguistics*, 39(2):301–353.

Caroline Sporleder and Linlin Li. 2009. Unsupervised Recognition of Literal and Non-Literal Use of Idiomatic Expressions. In *Proceedings of the 12th Conference of the European Chapter of the ACL*, pages 754–762, Athens, Greece.

Peter Turney, Yair Neuman, Dan Assaf, and Yohai Cohen. 2011. Literal and Metaphorical Sense Identification through Concrete and Abstract Context. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing*, pages 680–690, Edinburgh, UK.

Ulli Waltinger. 2010. German Polarity Clues: A Lexical Resource for German Sentiment Analysis. In *Proceedings of the 7th International Conference on Language Resources and Evaluation*, Valletta, Malta.

Corpus-Based Linguistic Typology: A Comprehensive Approach

Dirk Goldhahn **Uwe Quasthoff** **Gerhard Heyer**
Natural Language Processing Group,
University of Leipzig,
Germany
dgoldhahn,quasthoff,heyer
@informatik.uni-leipzig.de

Abstract

This paper will have a holistic view at the field of corpus-based linguistic typology and present an overview of current advances at Leipzig University. Our goal is to use automatically created text data for a large variety of languages for quantitative typological investigations. In our approaches we utilize text corpora created for several hundred languages for cross-language quantitative studies using mathematically well-founded methods (Cysouw, 2005). These analyses include the measurement of textual characteristics. Basic requirements for the use of these parameters are also discussed. The measured values are then utilized for typological studies. Using quantitative methods, correlations of measured properties of corpora among themselves or with classical typological parameters are detected. Our work can be considered as an automatic and language-independent process chain, thus allowing extensive investigations of the various languages of the world.

1 Introduction

Text corpora are a versatile basis for linguistic analyses. They allow for investigations of different aspects of languages, among others grammatical levels like morphology or syntax. The World Wide Web is a comprehensive source used for the creation of corpora. One advantage of using Web text is the availability of data for a large variety of languages. Since linguistic typology is concerned with cross-linguistic universals of language, Web corpora are an attractive source for investigations in this field. But as of today, few typological studies make use of automatically created text corpora or even Web based corpora.

Our goal is to use the textual data of the Leipzig Corpora Collection (LCC) (Quasthoff, 1998; Quasthoff, 2006; Biemann, 2007) in typological studies. Hence, we created an automatic and language independent process chain, which includes all steps necessary for this intention. As a first step this involves the acquisition and creation of the text corpora of LCC for several hundred languages. Following this, various measurements of different complexity such as average word length or average number of syllables per word are taken on these corpora. In addition constraints for their application for typological analyses are determined. Finally these measurements are utilized for corpus-based typological analyses applying quantitative methods. We determine correlations between measurements and dependencies between measurements and typological parameters like morphological type or position of case marking. This also allows to predict such classical typological features such as word order. Focus of the paper will be on general methodologies. Simple examples will be presented to illustrate the possibilities of the approaches. They will elucidate that even when using a fully automatic analysis, which in many cases will be very superficial, general characteristics of languages can still be captured. By allowing for broad analyses using large datasets this approach can complement existing typological methodologies and form a basis for further manual inspection and interpretation.

2 Acquisition and creation of Web corpora

Web corpora of LCC form the basis of the following typological investigations. LCC offers access to corpus-based monolingual full form dictionaries via a Web interface, Web services and allows for the download of text data. Corpora for more than 200 languages are available online[1]. The dictionaries contain statistical information for each word of the corpus.

All in all, corpora for more than 1,000 languages have been created which will be used for the following analyses. Corpus sizes vary from several hundred million to about 8000 sentences (languages where only the New Testament exists). Because of copyright issues many of them cannot be made available online. This holds, among others, for Bible texts. All corpora and their corresponding statistics are created from web pages. Thus, a process chain for the automatic acquisition, creation and statistical evaluation of corpora from web sources has been implemented which is presented in Goldhahn (2012).

3 Corpus-based statistics

3.1 Measurements

This paper aims at using simple corpus-based statistics for typological studies. Thus, measurements on the corpora have to be taken. Therefore, all in all more than 100 features are measured for each corpus using an automated framework. The measurements are conducted on different levels of language like sentences, words or letters. These levels are easy to identify for nearly all languages. Thus they are ideal for comparable studies. Other measurements are concerned with entities, which are more difficult to determine, such as syllables.

Among the measurements are features such as:

- Average word length in characters
- Average sentences length in words or characters
- Text coverage of the most frequent words
- Different measurements of vocabulary richness
- Entropy on word and character level
- Average number of syllables per word or sentence
- Average syllable length
- Ratio of prefixes and suffixes

For a more complete list of possible features see Goldhahn (2013).

Some values can be determined quite easily like the average sentence length in words, as long as word and sentence boundaries are identified correctly. Most languages use white space to separate words which allows for easy segmentation. Few languages are lacking clear word boundaries in their written form (e.g. Chinese and Japanese). In such cases specific tools such as Stanford Word Segmenter[2] are used.

For few measurements, only a rough approximation is possible. Examples are features concerned with prefixes and suffixes. Without analyzing the morphological processes of a language in detail, assertions about affixes are difficult. Therefore we chose to consider typical word beginnings and endings. Among them we identified those which discriminate many otherwise identical word pairs. This appeared to be a good approximation for affixation in many languages. Syllables also turned out to be difficult entities to measure, mainly because of the varying use of certain letters as vowels or consonants as well as the use of diphthongs depending on language. Therefore we used the language independent algorithm of Sukhotin (1988) to determine vowels and consonants of each language using text samples. The number of syllables of a word is equal to the number of syllable peaks. For most languages the number of peaks is close to the number of vowels in a word, since diphthongs are normally rarely encountered. On this basis statistics concerned with syllables can be computed und used for further analysis.

[1] http://corpora.uni-leipzig.de

[2] http://nlp.stanford.edu/software/segmenter.shtml

3.2 Constraints on measurements

Measurements on text corpora depend on different factors such as:

- Language
- Preprocessing
- Measurement process
- General characteristics of the texts like genre, text type or text size

In this paper we are mainly interested in changes of certain measured properties dependent on language. The preprocessing and the process of measurement, which can also have an influence on the resulting values (Eckart, 2012), are kept identical at all time. But other general characteristics of the corpora differ greatly between the texts in question. Therefore not every measured value is useful for typological analyses. Certain constraints have to be met or considered to enable proper insights from typological experiments.

First of all we examined the relative standard deviation (SD) of the measurements between languages. Especially in case of roughly approximated measurements a high cross-language variance can improve results of statistical tests used in the following typological analyses. Table 1 depicts relative SD for different measurements.

In addition we inspected the influence of textual characteristics such as text type, subject area or corpus size. This can lead to limitations concerning the usability of different corpora for certain measured parameters. Hence, it is desirable to have a higher cross-language SD in comparison to these other textual properties. Examples for these comparisons can be found in Table 2.

Furthermore, classical typological parameters normally vary less within groups of genealogically related languages. Since we aim at relating our measurements to typological features, the same is expected for our measurements. Table 3 shows, that this is generally the case. Exceptions in certain language families - just as word length in this case - can be subject for further investigations.

Negative results in these analyses do not necessarily exclude certain statistics from further investigations. But one has to assume that they will have impact on the results of typological studies conducted. As an example, the well-known Type-Token-Ratio (TTR), which measures the ratio of the number of different words forms to the number of total words in a text, is examined. It is well known that TTR is susceptible to changes of text size. When conducting a study with corpora of varying size, this will probably reduce the statistical significance of the results or even produce invalid results. One solution is to unify the amounts of text of the different corpora used. Since this results in throwing away valuable data, it is often an alternative to modify the statistic in question. Type-Token-Ratio is a measure of vocabulary richness. However, other measures of this property such as Turing's Repeat Rate, which measures the average number of words until a random word in the text occurs again, are hardly influenced by corpus size.

4 Typological analyses

4.1 Linguistic typology

Linguistic typology is concerned with the classification of languages according to their structural properties. This allows for the identification of possible and preferred structures of language. On the one hand typology determines typological parameters used for language classification. On the other hand it examines regularities or universals, which these parameters follow. Among them are relationships between different typological features (Greenberg, 1963).

4.2 Methods

In this section we relate simple features of text corpora, which can be determined using automatic means, with classical typological parameters of language, which describe different levels of language such as morphology or syntax. Furthermore we try to relate different measured features. We use quantitative methods like correlation analysis (Pearson product-moment correlation coefficient) and tests of significance (Wilcoxon, 1945; Mann, 1947) to analyze and confirm such relationships (Cysouw, 2005). In addition we predict typological parameters using methods of supervised machine learning or Bootstrapping approaches. In comparison to other works in this field (Fenk-Oczlon, 1999)

Measurement	Average	Relative SD
Average word length (Types)	9.11	0.37
Average word length (Tokens)	5.55	0.46
Average sentence length in words	26.87	0.27
Average sentence length in char.	161.98	0.23
Ratio of suffixes and prefixes	4.10	1.96
Text coverage of top 100 words	56.82	0.21

Table 1: Average values and relative standard deviation for corpus-based measurements.

Measurement	$\frac{\text{SD(Language)}}{\text{SD(Corpus size)}}$	$\frac{\text{SD(Language)}}{\text{SD(Text type)}}$	$\frac{\text{SD(Language)}}{\text{SD(Subject area)}}$
Average sentence length in words	107.41	8.65	13.20
Average sentence length in characters	77.032	6.23	7.67
Ratio of suffixes and prefixes	18.78	17.69	25.84
Syllables per sentence	30.25	8.22	7.33
Type-Token-Ratio	1.16	8.21	6.13
Turing's Repeat Rate	238.95	6.37	8.69
Text coverage of the top 100 words	530.85	7.93	8.75

Table 2: Comparison of standard deviations of corpus-based measurements. Values larger than 1 imply a higher cross-language standard deviation compared to the standard deviation when varying other features such as corpus size. Values much larger than 1 are desirable.

Measurement	$\frac{\text{SD(Random)}}{\text{SD(Germanic)}}$	$\frac{\text{SD(Random)}}{\text{SD(Indo-European)}}$	$\frac{\text{SD(Indo-European)}}{\text{SD(Germanic)}}$
Average word length (types)	5.22	0.71	7.38
Average word length (tokens)	4.51	0.72	6.26
Average sentence length in words	5.16	2.30	2.24
Average sentence length in characters	3.61	2.37	1.52
Type-Token-Ratio	2.54	1.80	1.41
Turing's Repeat Rate	6.46	2.09	3.08
Ratio of suffixes and prefixes	4.70	2.71	1.73
Text coverage of the top 100 words	4.18	1.33	3.14

Table 3: Comparison of cross-language standard deviations between language groups of different coherence. A random sample of languages is compared to a sample of Indo-European or Germanic languages. In general values larger than 1 are expected and imply a higher standard deviation in the less coherent language group.

the process does not contain any manual steps. We use automatically generated text resources combined with an automatic measurement process. Together they allow for the analysis of big textual data in several hundred languages while considering a high number of features and possible relations.

4.3 Results

4.3.1 Correlations between measurements

We were able to detect several correlations between measured parameters of corpora. By applying correlation analysis to comparable corpora in 730 languages we achieved results of high statistical significance and found interesting correlations or confirmed known ones. Some of them will be presented in this section. Since the focus of this paper is on methodology, only very brief interpretations of results will be offered. Such results can be a starting point for typologists that need to analyze each language in detail in order to accomplish a full interpretation.

We found:

- A negative correlation between average length of words and average length of sentences (in words): $Kor_e = -0.55, p < 0,001\%$, sample size of 730. The longer the average word of a language is, the fewer words are usually utilized (or needed) to express a sentence.

- A negative correlation between average number of syllables per word and average number of words per sentence: $Kor_e = -0,49, p < 0,001\%$, sample size of 730. The more syllables the average word of a language has, the fewer words are typically used to express a sentence.

4.3.2 Relationships between measurements and typological parameters

We also determined various relations between measured parameters and classical typological parameters using tests of significance. Typological information was taken from the World Atlas of Language Structures (WALS[3]) (Cysouw, 2007b). Since typological data is sparse, sample sizes are usually smaller than 730 languages.

Only a small sample of all results which were achieved is presented in this paper. For a full overview see (Goldhahn, 2013).

We found a significant relation between ratio of suffixes and prefixes and position of case marking (end of word vs. beginning of word):

- $p < 0.001\%$, mean values of 10.48 and 0.7 and sample sizes of 57 and 11.

Our simple automated measurements regarding affixes are sufficient to capture a relation to actual processes of affixation in languages. Although we obviously measure more than just case marking a significant relation can still be established. It seems that case marking has a big influence on our measurement.

Morphological type
We found:

- A significant relation between average length of words of a language and its morphological type (concatenative vs. isolating): $p < 1\%$, mean values of 8.43 and 6.95 and sample sizes of 68 and 8.

- A significant relation between measured amount of affixation of a language and its morphological type (concatenative vs. isolating): $p < 0.5\%$, mean values of 21.20 and 10.06 and sample sizes of 68 and 8.

- A found a significant relation between entropy on word level of a language and its morphological type (concatenative vs. isolating): $p < 0.05\%$, mean values of 9.95 and 8.64 and sample sizes of 68 and 8.

Several measurements we conducted, among them those concerning average word length or affixation, are related to morphological features of languages. Opposing features such as concatenative and isolating morphological type are presented as an obvious example.

Syllables
We confirmed results of Frank-Oczlon (1999) for a considerably larger sample of languages with higher significance. We also enriched these results with further findings. Among others we discovered significant relations between:

- Average number of syllables per sentence and word order (SOV vs. SVO): $p < 0.001\%$, mean values of 56.95 and 45.27 and sample sizes of 111 and 137.

- Average number of syllables per word and morphological type (concatenative vs. isolating): $p < 5\%$, mean values of 2.06 and 1.64 and sample sizes of 68 and 8.

4.3.3 Prediction of typological parameters

Typological parameters have been determined for many languages of the world

[3] http://wals.info/

and can be looked up in collections such as WALS. But for many parameters only partial knowledge is available. Hence, ways to predict typological features based on automatic measurements would be of great help.

Supervised machine learning
One way to predict typological parameters is the use of supervised machine learning. To illustrate the possibilities of this approach the example of morphological type of a language will be discussed. Once again only concatenative and isolating languages will be analyzed, which form two extremes concerning morphological properties. Table 4 shows the probabilities of correct classification of morphological type using 76 languages which WALS assigned to one of these two classes. As input different measured features or combinations of them were utilized. Especially when using a mix of several features for prediction, high accuracies of over 90% were achieved.

Applying this method the usability of a high number of corpus features for predicting different typological parameters can be investigated.

Features used	**Correctly classified**
Words per sentence	74.40%
Number of word forms	87.76%
Words per sentence, number of word forms, syllables per word	91.84%

Table 4: Probability of correct prediction of morphological type (concatenative vs. isolating) using a Support Vector Machine based on different feature sets.

Bootstrapping
Furthermore we utilized automatic Bootstrapping approaches to determine typological parameters. Some information such as part of speech of words (POS), which is necessary to predict certain typological parameters, can only be assigned automatically for few well-resourced languages. Using parallel text such as Bibles it is possible to align corresponding words across languages (Melamed, 1996; Biemann, 2005; Cysouw, 2007a). This knowledge can then be used to spread information about features like POS to further languages. By applying graph partitioning algorithms such as Chinese Whispers (Biemann, 2006) we were able to successfully transfer this information about POS to languages without known POS-Tagger. This knowledge together with information about translational equivalents was then used to predict the typological parameter of word order. Therefore word order information of a source language (German) was transferred to the target languages using sample sentences (see Figure 1). This way we were able to successfully determine the correct word order for sample languages. See Goldhahn (2013) for details about the methodologies of this example, such as the use of a simplified Tagset.

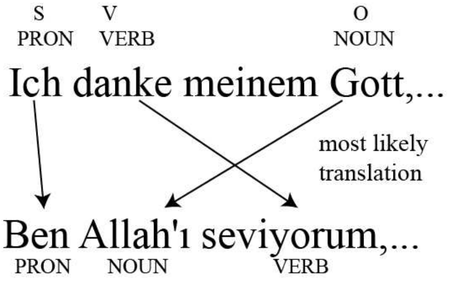


Figure 1: Depiction of the information used to predict word order in a target language.

5 Conclusion

In this paper, we presented a novel approach to corpus-based linguistic typology allowing for a new kind of typological analyses. Using an automatic process chain we were able to measure statistical features of corpora of web text for several hundred languages. These properties were applied in quantitative typological analyses to detect correlations with classical typological parameters or to predict such parameters. Several simple results were presented. They give insight into the possibilities of the methodologies described in this paper and show that despite using a superficial automatic approach general characteristics of languages can still be captured. By adding further features that can be measured automatically or by analyzing relationships to additional typological parameters a wide area of typological issues can be investigated. Since this approach facilitates broad analyses of very large datasets it can complement existing typological work and form a basis for further manual inspection and interpretation.

References

Baroni, M.; Bernardini, S. 2004. BootCaT: Bootstrapping corpora and terms from the web. *Proceedings of LREC 2004.*

Biemann, C.; Quasthoff, U. 2005. Dictionary acquisition using parallel text and cooccurrence statistics. *Proceedings of NODALIDA 2005*, Joensuu, Finland.

Biemann, C. 2006. Chinese whispers: an efficient graph clustering algorithm and its application to natural language processing problems. *Proceedings of the First Workshop on Graph Based Methods for Natural Language Processing* (pp. 73-80). Association for Computational Linguistics.

Biemann, C.; Heyer, G.; Quasthoff, U.; Richter, M. 2007. The Leipzig Corpora Collection - Monolingual corpora of standard size. *Proceedings of Corpus Linguistic 2007*, Birmingham, UK.

Cysouw, M. 2005. Quantitative methods in typology. In Altmann, G.; Khler, R.; Piotrowski, R. (eds.). *Quantitative linguistics: an international handbook*, 554 - 578. Berlin: Mouton de Gruyter.

Cysouw, M.; Wlchli, B. 2007a. Parallel texts: using translational equivalents in linguistic typology. *Sprachtypologie und Universalienforschung*, 60(2), 95-99.

Cysouw, M. 2007b. Special issue on analyzing the World Atlas of Language Structures. *Sprachtypologie und Universalienforschung*.

Eckart, T.; Quasthoff, U.; Goldhahn, D. 2012. The Influence of Corpus Quality on Statistical Measurements on Language Resources. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012).*

Fenk-Oczlon, G.; Fenk, A. 1999. Cognition, Quantitative Linguistics, and Systemic Typology. *Linguistic Typology*, 3: 151 - 177.

Goldhahn, D.; Eckart, T.; Quasthoff, U. 2012. Building Large Monolingual Dictionaries at the Leipzig Corpora Collection: From 100 to 200 Languages. *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC 2012).*

Goldhahn, D. 2013. Quantitative Methoden in der Sprachtypologie: Nutzung korpusbasierter Statistiken. Dissertation, University of Leipzig, Leipzig, Germany.

Greenberg, J. H. 1963. Some universals of grammar with particular reference to the order of meaningful elements. *Universals of language*, 2, 73-113.

Mann, H. B.; Whitney, D. R. 1947. On a test of whether one of two random variables is stochastically larger than the other. *The annals of mathematical statistics*, 18(1).

Melamed, I. D. 1996. Automatic construction of clean broad-coverage translation lexicons. *Proceedings of the 2nd Conference of the Association for Machine Translation in the Americas*, Montreal, Canada.

Quasthoff, U. 1998. Projekt der deutsche Wortschatz. Heyer, G.; Wolff, Ch. (eds.), *Linguistik und neue Medien*, Wiesbaden, pp. 93-99.

Quasthoff, U.; Richter, M.; Biemann, C. 2006. Corpus Portal for Search in Monolingual Corpora. *Proceedings of LREC 2006.*

Sukhotin, B. V. 1988. Optimization algorithms of deciphering as the elements of a linguistic theory. *Proceedings of the 12th conference on Computational linguistics-Volume 2* (pp. 645-648). Association for Computational Linguistics.

Wilcoxon, F. 1945. Individual comparisons by ranking methods. *Biometrics bulletin*, 1(6), 80-83.

Cleaning the Europarl Corpus for Linguistic Applications

Johannes Graën **Dolores Batinic** **Martin Volk**
Institute of Computational Linguistics
University of Zurich, Zurich, Switzerland
{graen|batinic|volk}@cl.uzh.ch

Abstract

We discovered several recurring errors in the current version of the Europarl Corpus originating both from the web site of the European Parliament and the corpus compilation based thereon. The most frequent error was incompletely extracted metadata leaving non-textual fragments within the textual parts of the corpus files. This is, on average, the case for every second speaker change.

We not only cleaned the Europarl Corpus by correcting several kinds of errors, but also aligned the speakers' contributions of all available languages and compiled everything into a new XML-structured corpus. This facilitates a more sophisticated selection of data, e.g. querying the corpus for speeches by speakers of a particular political group or in particular language combinations.

1 Introduction

Koehn (2005) first presented his compilation of the Europarl Corpus comprising the European Parliament's debates in 2001 and has continued updating it with the latest available data since then. For compiling the transcriptions of the debates into a corpus, he downloaded the particular web pages for any available language[1] from the Parliament's web site[2]. He then parsed the HTML source code in order to separate markup, meta-information, like the structure of the debates (chapters and turns), speaker information and so forth, as well as actual textual data (i.e. topics, comments and the respective speakers' speeches) and transferred the latter two into plain text files. As of today, the corpus has grown to 968 plenary sessions from 1996 to 2011 in up to 21 languages in parallel[3].

The plenary sessions consist of a list of agenda items (called 'chapters') themselves consisting of speech contributions of one or more speakers in combination with descriptive comments by the transcribers (hereinafter called 'turns'). Usually, the first and last turn within a chapter lies with the president of the European Parliament.

The Europarl Corpus is widely used for many diverse language technology applications such as "word sense disambiguation, anaphora resolution, information extraction" (ibid.), statistical machine translation (Cohn and Lapata 2007), grammar projection (Bouma et al. 2008) "unsupervised part-of-speech tagging" (Das and Petrov 2011) or "learning multilingual semantic representations" (Hermann and Blunsom 2014).

When we evaluated the appropriateness of the Europarl Corpus for our intended applications (namely part-of-speech tagging, chunking and parsing as well as word, chunk and tree alignments over parallel texts), we encountered sev-

[1] Before 2004, the European Union had 11 official languages which are part of the first Europarl Corpus version.

[2] Available at http://www.europarl.europa.eu/.

[3] Although Maltese and Irish have become official languages as of 2004 and 2007 respectively, no transcripts of the debates are available in these languages.

eral problems which might be negligible when the data is analyzed statistically, but impair the results with regard to individual examples. We found that shallow cleaning already led to a considerable improvement on further processing steps like part-of-speech tagging or turn alignment. The latter is necessary before sentence alignment due to partial absence of translations that would evoke wrong sentence alignments in most cases. That is why we decided to clean Koehn’s corpus for our own purposes and with the objective of making the result publicly available again.

Crawling the web pages of the European Parliament’s debates and extracting the aforesaid information by ourselves could have been an alternative way to obtain clean data. As some of the errors originate from the European Parliament’s website and thus need to corrected anyway and the debates before the parliamentary term of 1999–2004 are no longer available online, we opted for cleaning the existing corpus data.

For publishing our corpus compilation, we decided to store the cleaned version of the Europarl Corpus in XML format, which will enable a more fine-grained selection of data than the original plain text files. By means of XPath (Clark, DeRose, et al. 1999), one can query the corpus for particular speakers, dates or political groups. Building a sub-corpus of transcribed speeches in one language with translations to a second one and comparing it to another one where transcriptions and translations are arranged the other way round is just as simple as using XPath expressions to match the language code of a speaker’s contribution and the language label of the text. Additionally, it is possible to address the structure of the respective discourse (from sequential comments of each group’s position to dialogue alike controversies).

2 Error classification

Koehn refers to the extraction of textual data from the website of the European Parliament as a “cumbersome enterprise”. Hence, several formatting problems such as the diversified use of encoding alternatives for certain characters, HTML entities or wrong quotation marks (see Ex. 1) made it into his published corpus.

```
(1)  1. Non sono contrario a Eurojust, ma non deve
        trasformarsi in una "super-istituzione”.
     2. Je l’appellerais un vote d’»avis conforme
        élargi», qui n’est pas …
     3. … und zwar wegen der Existenz so genannter
        ,,Energieinseln" wie dem Ostseeraum, …
```

In addition to that, numerous errors were introduced by Koehn’s corpus compilation. On the one hand, speaker information was often incompletely extracted from the web pages, i.e. meta-information such as the language used by the speaker is classified as part of the speakers’ utterance (see Ex. 2), or comprises textual information, mostly comments structuring the transcriptions (see Ex. 3).

```
(2)  1. (RO) Бих искала да поздравя г-н Stolojan за …
     2. , Kommissionen. (EN) Når det gælder …
     3. Miller (PSE). (EN) Herr Präsident, ich …
     4. ). (IT) Κυρία Πρόεδρε, …
     5. (RO) Бих искала да поздравя г-н Stolojan за …
```

```
(3)  1. <SPEAKER ID="66" LANGUAGE="PL"
        NAME="Protasiewicz (PPE-DE )."
        AFFILIATION="(Applaus)">
     2. <SPEAKER ID="11" LANGUAGE="" NAME="Tannock
        (PPE-DE )." AFFILIATION="(">
     3. <SPEAKER ID="115" LANGUAGE="" NAME=""
        AFFILIATION="The Minutes of the previous
        sitting were approved.)"/>
```

On the other hand, Koehn applied shallow tokenization rules, which in some languages resulted in partly tokenized texts. When apostrophized prepositional articles in Italian and French are separated from the following word with a white space (“all’ uomo” instead of “all’uomo”) and the text is fed to a tokenizer afterwards, for instance, the already tokenized parts will be handled again, thus leading to potential erroneous output as shown in Figure 1.

A more severe problem is that text in HTML tags within the textual parts of the pages is omitted (Example 4 shows a sample sentence from the website of the European Parliament (1) and its counterpart in Koehn’s corpus compilation (2)) and the original text is thus unrecoverable.

```
(4)  1. Il caso Terni è, per molti versi, la punta di
        un <span class="italic">iceberg</span>.
     2. Il caso Terni è, per molti versi, la punta di
        un .
```

Apart from the errors introduced by Koehn, we identified several problems originating from the text of the original web pages. Besides certain

```
1. L' ordine del giorno reca la fissazione dell' ordine dei lavori.
   /NUM '/PON ordine/NOM del/PRE:det giorno/NOM recere/VER:cpre il/DET:def fissazione/NOM <unknown>/NOM '/PON
   ordine/NOM del/PRE:det lavorio|lavoro/NOM ./SENT
2. L'ordine del giorno reca la fissazione dell'ordine dei lavori.
   il/DET:def ordine/NOM del/PRE:det giorno/NOM recere/VER:cpre il/DET:def fissazione/NOM del/PRE:det ordine/NOM
   del/PRE:det lavorio|lavoro/NOM ./SENT
```

Figure 1: Example sentence with corresponding output of the TreeTagger (cf. Schmid 1994) which performs tokenization before tagging. (1) shows the the sentence as it appears in Koehn's corpus, in (2) the partial tokenization is undone. Relevant corresponding parts are highlighted, the correct tagging is underlined.

parts being absent in otherwise completely translated documents, comments were often not translated (see Ex. 5). While this kind of missing data cannot be adjusted at all, correction candidates include, for example, wrong punctuation (especially quotation marks), the common misspelling of `è` in Italian as `e'` and `perchè` instead of `perché` (The same applies to `ché` and `affinché`), missing space characters in front of French punctuation signs and wrong number formats.

(5)
```
1. Schluss der Sitzung
   <P>
   (The sitting closed at 22.25)
```

We classified the errors and problem as shown in Figure 2 according to their source, impact as well as frequency and grouped them into categories which best describe their nature. The impact of a particular type of error is classified as “low”, “medium” or “high”, depending on how further processing tools are affected by the particular error type, i.e. whether they skip or autocorrect it or produce wrong output or analysis. We evaluated the output of tools for common processes such as tokenization, part-of-speech tagging or parsing in order to decide what impact a particular error type might have. However, it will vary for different kind of applications.

3 Correcting errors and enriching the corpus

In our cleaning of the corpus, we traversed all of Koehn's corpus files for each plenary session and extracted structural elements, meta-information, comments and either original (transcribed) or translated speeches. According to the type of data, we used different cleaning rules.

Language specifications not belonging to the official or semi-official languages of the European Union, for example, are in most cases obvious mistakes (e.g. using `uk` (Ukrainian) or `gb` (not assigned) as language code instead of `en` (English) for a British speaker) and not taken over to the cleaned corpus so that the respective turns lack the attribute for the original language of the utterance. In order to enable the speaker turn alignment, we also identified the utterances of the chairmen of the parliament (see Ex. 6 for examples) for each language by comparing their names to the respective language's term variants for the president (in German, for instance: “Der Präsident”, “Die Präsidentin”, “Präsident” and “Präsidentin”).

(6)
```
1. <SPEAKER ID="213" NAME="Le Président">
2. <SPEAKER ID="213" NAME="Πρόεδρος">
3. <SPEAKER ID="213" NAME="elnök">
4. <SPEAKER ID="213" NAME="De Voorzitter">
5. <SPEAKER ID="213" NAME="Talmannen">
```

In addition, we split multiple speaker names like “Bonde, Lis Jensen □i Sandbæk” (taken from the Romanian text) into the particular names (“Bonde”, “Lis Jensen” and “Sandbæk”) and marked the turn as having multiple authors. Multiple authors are only possible in written parts added to the transcripts, usually being the “explanations of vote”, in order to facilitate the interlingual alignment of turns and the access to the author information in the corpus.

Having removed the meta-information, we applied a set of further cleaning rules to the actual textual information. This set comprises corrections for wrong characters and punctuation, marks URLs, parliamentary reports as well as legislative procedures and undoes the partly performed tokenization on a per language level. We also applied a multitude of rules to identify all kinds of correct and wrong quotation marks and unified them.[4] Additional language-specific rules help us to meet orthography requirements.

[4] We provide the correct language-specific use of quotation marks in the respective language as additional informa-

category	error/problem	source[5]	impact	frequency[6]
coding	invalid UTF-8 encoding	K	low	$< 10^{-5}$/files
	undecoded HTML entities	EP	medium	$< 10^{-2}$/lines
	code variants[7]	EP	medium	$> 6\,\%$/lines
orthography	consistently misspelled words	EP	medium	$< 2\,\%$/lines[8]
	wrong/incoherent quotation marks (see Ex. 1)	both	low	$< 1\,\%$/lines
missing data	words omitted (see Ex. 4)	K	high	$> 10^{-3}$/lines
	comments untranslated (see Ex. 5)	EP	low	$> 10^{-5}$/lines
	non-matching turns[9]	both	high	$> 10^{-3}$/turns
processing	text partly tokenized	K	medium	$> 10^{-3}$/tokens
	text marked as meta-information (see Ex. 3)	K	low	$> 1\,\%$/lines
	meta-information marked as text (see Ex. 2)	K	high	$> 6\,\%$/lines

Figure 2: Error classification scheme.

After the corresponding documents for a particular plenary session in any available language have been mapped to internal representations, we aligned the corresponding speakers' turns in all languages. In the majority of cases (58%), the respective documents have exactly the same structure so that the alignment process is straightforward. For all other cases, we searched for possible alignments with respect to the given order of turns and calculated a score based on the Levenshtein distance between two speaker names, the property of a speaker being president of the European Parliament or not, the chapter a turn is listed in and the count of textual parts within that turn. We then computed a list of alignments minimizing that measure. In this vein, we are able to correct wrongly aligned turns, i.e. for instance those that were only based on the id attribute given by Koehn (2005).

4 Evaluation

We calculated that at least 6% of all lines from Koehn's corpus erroneously contain meta-information, which we were able to correct. Sole language information was the most frequent case (cf. the last example of Fig. 2). This quantity corresponds to 50% of the meta-data definitions in the corpus, thus implying that in half of the cases Koehn's meta-data extraction rules failed.

About 1% of the text lines in Koehn's Europarl contain comments introduced by the transcriber of the sessions. We marked all these comments, even some that did not possess any specific formatting, by comparing lines with a handcrafted list. In about 2% of the lines we were able to detect and eliminate non textual fragments originating from the European Parliament's web pages.

We found quotes in 3% of the lines which we marked as such by applying rules that were made to recognize even the wrong quotation marks (cf. Fig. 1).

Since we created restrictive rules to only correct these systematic errors that we identified, none of them remain uncorrected and a quantitative evaluation of corrected errors would be futile. Nonetheless, 12% of all apostrophized articles, prepositions and prepositional articles, for instance, features a following white space (probably due to Koehn's shallow tokenization rules) which is prone to cause problems as depicted in Figure 1.

Some errors known to us, including, for instance, Catalan text within the Spanish parts or

tion so that the text with markup can instantaneously be converted to its correct plain form.

[5] Either the Website of the European Parliament (EP) or Koehn's compilation (K) or both.

[6] The frequencies are calculated or estimated based on the source corpus files, their lines (text or meta-information) or tokens.

[7] Various hyphens and dashes as well as homoglyphs.

[8] Calculated for misspelled è in Italian. As this is the most frequent case of misspelling and we found that approximately 2 % of the lines of the Italian texts, the overall frequency needs to be lower.

[9] The number of turns of the respective languages in a particular chapter or session don't match. This can be due to a wrong classification of text as meta-information, meta-information as text or the absence of one or more turns.

turns in one language located in a different chapter than the corresponding parts of the other languages and hence not being aligned, occur only infrequently. Thus, we decided to leave them alone.

5 Conclusions

We improved the quality of the Europarl Corpus described by Koehn (ibid.) and recompiled in 2012 by

- correcting the classification of meta-data versus textual data,
- unifying punctuation marks and other kinds of character classes,
- marking identifiers of political groups in the European Parliament and
- removing fragments of characters which are legacy of the original source but do not belong to the textual data.

Furthermore, we enriched the corpus by

- marking agenda items, comments and speech parts where distinguishable,
- marking quotes in a way that they can be converted to each language's preferences, and
- aligning speakers' contributions (turns) in all available languages.

We considered adding the respective speakers' mother tongue as supplementary information in order to pave the way for an even deeper linguistic investigation. Unfortunately, we did not find any data that could have enabled us to do so with a reasonable effort and accuracy.

The resulting edited and recompiled corpus serves (like Koehn's original one) as a rich source for any kind of linguistic application, but in addition to that provides easier access to cleaned text and meta-data both being arranged in an XML structure (see Listing 1). Corresponding speaker turns (one being the speech's transcription and the others translation of it) are aligned by sharing the same turn node.

The corrected and structured Europarl Corpus as well as some technical documentation can be obtained from `http://pub.cl.uzh.ch/purl/costep`.

6 Future work

The debates of the European Parliament keep being an important resource of parallel texts in many languages for a multitude of language technology applications. New web pages are added for every completed plenary sessions and made available in an increasing number of languages.

We place great importance on the availability of up-to-date, turn-aligned parallel texts from the European Parliament's debates and suggest to integrate the tasks of crawling the web pages, cleaning the textual data and aligning the respective speaker's turns. We believe that in this vein certain types of errors can be corrected with less effort while others can be entirely avoided.

Acknowledgment

This research was supported by the Swiss National Science Foundation under grant 105215_146781/1 through the project "SPARCLING – Large-scale Annotation and Alignment of Parallel Corpora for the Investigation of Linguistic Variation".

References

Bouma, Gerlof et al. (2008). "Parallel LFG Grammars on Parallel Corpora: A base for practical triangulation". In: *Proceedings of the Lexical Functional Grammar (LFG) Conference*. (Sydney). International Lexical Functional Grammar Association (ILFGA), pp. 169–189.

Clark, James, Steve DeRose, et al. (1999). *XML path language (XPath) version 1.0*.

Cohn, Trevor and Mirella Lapata (2007). "Machine translation by triangulation: Making effective use of multi-parallel corpora". In: *Annual Meeting of the Association for Computational Linguistics (ACL)*. (Prague). Vol. 45. 1. Association for Computational Linguistics (ACL), pp. 728–735.

Das, Dipanjan and Slav Petrov (2011). "Unsupervised Part-of-speech Tagging with Bilingual Graph-based Projections". In: *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL): Human Language Technologies - Volume 1*. (Portland), pp. 600–609.

```xml
<?xml version="1.0" encoding="UTF-8"?>
<session date="2010-05-05">
  <chapter id="1">
    <headline language="bg">Възобновяване на сесията</headline>
    <headline language="cs">Pokračování zasedání</headline>
    <headline language="da">Genoptagelse af sessionen</headline>
    <headline language="de">Wiederaufnahme der Sitzungsperiode</headline>
    <headline language="el">Επανάληψη της συνόδου</headline>
    <headline language="en">Resumption of the session</headline>
    <headline language="es">Reanudación del período de sesiones</headline>
    <headline language="et">Istungjärgu jätkamine</headline>
    <headline language="fi">Istuntokauden uudelleen avaaminen</headline>
    <headline language="fr">Reprise de la session</headline>
    <headline language="hu">Az ülésszak folytatása</headline>
    <headline language="it">Ripresa della sessione</headline>
    <headline language="lt">Sesijos atnaujinimas</headline>
    <headline language="lv">Sesijas atsākšana</headline>
    <headline language="nl">Hervatting van de zitting</headline>
    <headline language="pl">Wznowienie sesji</headline>
    <headline language="pt">Reinício da sessão</headline>
    <headline language="ro">Reluarea sesiunii</headline>
    <headline language="sk">Pokračovanie prerušeného zasadania</headline>
    <headline language="sl">Nadaljevanje zasedanja</headline>
    <headline language="sv">Återupptagande av sessionen</headline>
    <turn id="1">
      <speaker president="yes" language="el">
        <text language="bg">
          <p type="speech">Възобновявам сесията на Европейския парламент, прекъсната на 22 април 2010 г.</p>
          <p type="speech">Протоколът от 22 април 2010 г. беше раздаден.</p>
          <p type="speech">Има ли някакви коментари?</p>
          <p type="comment">Протоколът от предишното заседание е одобрен</p>
        </text>
        <text language="cs">
          <p type="speech">Prohlašuji přerušené zasedání Evropského parlamentu ze dne 22. dubna 2010 za obnovené.</p>
          <p type="speech">Zápis z jednání ze dne 22. dubna 2010 byl rozdán.</p>
          <p type="speech">Má někdo připomínky?</p>
          <p type="comment">Zápis z předchozího zasedání byl schválen</p>
        </text>
        <text language="da">
          <p type="speech">Jeg erklærer Europa-Parlamentets session, der blev afbrudt torsdag den 22. april 2010, for genoptaget.</p>
          <p type="speech">Protokollen fra mødet den 22. april 2010 er omdelt.</p>
          <p type="speech">Hvis ingen gør indsigelse, betragter jeg den som godkendt.</p>
          <p type="comment">Protokollen fra foregående møde godkendtes</p>
        </text>
        <text language="de">
          <p type="speech">Ich erkläre die am 22. April 2010 unterbrochene Sitzung des Europäischen Parlaments für wieder aufgenommen.</p>
          <p type="speech">Das Protokoll vom 22. April 2010 wurde ausgeteilt.</p>
          <p type="speech">Gibt es dazu Anmerkungen?</p>
          <p type="comment">Das Protokoll der vorherigen Sitzung wird angenommen</p>
        </text>
        <text language="el">
          <p type="speech">Κηρύσσω την επανάληψη της συνόδου του Ευρωπαϊκού Κοινοβουλίου η οποία είχε διακοπεί στις 22 Απριλίου 2010.</p>
          <p type="speech">Τα Συνοπτικά Πρακτικά της συνεδρίασης της 22ας Απριλίου 2010 έχουν διανεμηθεί.</p>
          <p type="speech">Υπάρχουν παρατηρήσεις επ' αυτών;</p>
          <p type="comment">Εγκρίνονται τα Συνοπτικά Πρακτικά της προηγούμενης συνεδρίασης</p>
        </text>
        <text language="en">
          <p type="speech">I declare resumed the session of the European Parliament adjourned on 22 April 2010.</p>
          <p type="speech">The Minutes of 22 April 2010 have been distributed.</p>
          <p type="speech">Are there any comments?</p>
          <p type="comment">The Minutes of the previous sitting were approved</p>
        </text>
```

Listing 1: Excerpt from a turn-aligned XML corpus file for a particular plenary session.

Hermann, Karl Moritz and Phil Blunsom (2014). “Multilingual Models for Compositional Distributed Semantics”. In: *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (ACL)*. (Baltimore). Association for Computational Linguistics (ACL).

Koehn, Philipp (2005). “Europarl: A parallel corpus for statistical machine translation”. In: *Machine Translation Summit*. (Phuket). Vol. 5. Asia-Pacific Association for Machine Translation (AAMT), pp. 79–86.

Schmid, Helmut (1994). “Probabilistic part-of-speech tagging using decision trees”. In: *Proceedings of International Conference on New Methods in Natural Language Processing (NeMLaP)*. (Manchester). Vol. 12, pp. 44–49.

Atomic: an open-source software platform for multi-level corpus annotation

Stephan Druskat[*†], **Lennart Bierkandt**[*%], **Volker Gast**[*†], **Christoph Rzymski**[*†], **Florian Zipser**[**‡]
[*]Friedrich Schiller University Jena, Dept. of English and American Studies, Jena/Germany
[**]Humboldt University of Berlin, Dept. of German Studies and Linguistics, Berlin/Germany
[†]{firstname.lastname}@uni-jena.de
[%]post@lennartbierkandt.de
[‡]zipseflo@hu-berlin.de

Abstract

This paper[1] presents Atomic, an open-source[2] platform-independent desktop application for multi-level corpus annotation. Atomic aims at providing the linguistic community with a user-friendly annotation tool and sustainable platform through its focus on extensibility, a generic data model, and compatibility with existing linguistic formats. It is implemented on top of the Eclipse Rich Client Platform, a pluggable Java-based framework for creating client applications. Atomic - as a set of plug-ins for this framework - integrates with the platform and allows other researchers to develop and integrate further extensions to the software as needed. The generic graph-based meta model Salt serves as Atomic's domain model and allows for unlimited annotation levels and types. Salt is also used as an intermediate model in the Pepper framework for conversion of linguistic data, which is fully integrated into Atomic, making the latter compatible with a wide range of linguistic formats. Atomic provides tools for both less experienced and expert annotators: graphical, mouse-driven editors and a command-line data manipulation language for rapid annotation.

1 Introduction

Over the last years, a number of tools for the annotation and analysis of corpora have been developed in linguistics. Many of these tools have been created in the context of research projects and designed according to the specific requirements of their research questions. Some tools have not been further developed after the end of the project, which often precludes their installation and use today.

Some of the available annotation tools, such as MMAX2 (Müller and Strube, 2006), @nnotate (Plaehn, 1998) and EXMARaLDA (Schmidt, 2004), have been designed to work on specific annotation types and/or levels (e.g., token-based or span-based; coreference annotations, constituent structures, grid annotations). However, the analysis of some linguistic phenomena greatly profits from having access to data with annotations on more than one or a restricted set of levels. For instance, an empirical, corpus-based analysis of scope relations and polarity sensitivity cannot be conducted without richly annotated corpora, with token-based annotation levels such as part-of-speech and lemma, and additionally syntactic and sentence-semantic annotations. And analyses of learner corpora, for example, benefit from an annotation level with target hypotheses, and a level that records discrepancies between target hypothesis and learner production.[3] Similarly, in order to analyze the information structure of utterances, annotations on the levels of syntax, phonology and morphology are essential, as

[1] This work is licensed under a Creative Commons Attribution 4.0 International License (CC BY 4.0). Page numbers and proceedings footer are added by the organizers. License details: http://creativecommons.org/licenses/by/4.0/.

[2] Atomic is open source under the Apache License 2.0.

[3] For an overview see Lüdeling and Hirschmann (forthcoming).

information structure is "interweaved with various [...] linguistic levels" (Dipper et al., 2007).[4] And finally, typological questions such as those that have been the topic of the research project "Towards a corpus-based typology of clause linkage"[5] – in the context of which Atomic has been developed – require corpora that have fine-grained annotations, at various levels.[6] Consequentially, any software designed for unrestricted multi-level corpus annotation should meet the following requirements.

(1) The data model has to be generic and able to accommodate various types of annotation based on the requirements of different annotation schemes. (2) The architecture needs to exhibit a high level of compatibility, as the corpora available for a specific research question may come in different formats. (3) The software must be extensible, since new types of annotation will have to be accommodated, and some may require new functionalities. (4) As the software should be usable by a wide variety of annotators – e.g., experienced researchers as well as students and specialists of various languages including field workers – it needs to provide support and accessibility for users with different levels of experience in corpus annotation.

There has been a trend in corpus linguistics towards the creation of multi-level corpora for a number of years now, which has of course had an effect on tool development as well. Therefore, there already exist some annotation tools which can handle annotations on more than one level, e.g., TrED (Pajas and Štěpánek, 2004) and MATE (Dybkjær et al., 1999). TrED has been developed mainly for the annotation of tree-like structures and therefore handles only limited types of annotations. These do not fully cover some of the required levels for, e.g., typological research questions. MATE, a web-based tools platform, is specifically aimed at multi-level annotation of spoken dialogue corpora in XML formats. Atomic's architecture avoids these limitations by using a generic graph-based model capable of handling potentially unlimited annotation types in corpora of spoken and written texts (cf. 2.2). WebAnno (Yimam et al., 2013) is another multi-level annotation tool currently under development, designed with a focus on distributed annotations. It has a web-based architecture partly based on brat.[7] In contrast to this, Atomic is built as a rich client for the desktop based on the Eclipse Rich Client Platform (RCP, McAffer et al. (2010)), which offers a number of advantages over the implementation as a web application, most importantly ease of extensibility, platform-independence, data security and stability, and network-independence.

The Eclipse RCP comes with a mature, standardised plug-in framework,[8] something which very few web frameworks are able to provide. And while it is a non-trivial task in itself to develop a truly browser-independent web application, the Eclipse RCP provides platform-independence out-of-the-box, as software developed against it run on the Java Virtual Machine.[9] Desktop applications also inherently offer a higher grade of data security than web-based tools, as sensitive data – such as personal data present in the corpus or its metadata – does not have to leave the user's computer. Additionally, access to the corpus data itself will be more stable when it is stored locally rather than remotely, as desktop applications are immune to server failures and the unavailability of server administrators. And finally, a desktop tool such as Atomic, which is self-contained in terms of business logic and data sources, can be used without an internet connection, which is important not only to field workers but also to anyone who wants to work in a place with low or no connectivity, e.g. on public transport.[10]

In terms of interoperability and sustainability, Atomic has been specifically designed to complement the existing interoperable software ecosystem of ANNIS (Zeldes et al., 2009), the search and visualisation tool for multilayer

[4] See also Lüdeling et al. (forthcoming).

[5] Cf. http://linktype.iaa.uni-jena.de.

[6] Annotation levels should minimally include morphology, syntax and information structure, ideally also semantics (sentence semantics and reference) and phonology (including prosody).

[7] Cf. http://brat.nlplab.org/.

[8] Eclipse Equinox, cf. 2.1.

[9] Ibid.

[10] Nevertheless it is of course possible to extend Atomic to use remote logic and/or data sources, such as databases.

corpora, Pepper (Zipser et al., 2011), the converter framework for corpus formats, and LAUDATIO (Krause et al., 2014), the long-term archive for linguistic data. Thus, it seamlessly integrates the compilation and annotation of resources with their analysis and long-term accessibility.

2 Architecture

In order to fulfill the above-mentioned requirements, Atomic's architecture is particularly concerned with extensibility, a generic data model, and a feature set which focuses on accessibility.

2.1 Extensibility

It should be possible for other research groups to build extensions for Atomic for their specific needs (e.g., new editors, access to remote data sources) and integrate them into the platform. This would increase the sustainability of both the platform and the extensions.

We have therefore decided to develop Atomic on top of the Eclipse RCP, an open-source Java application platform which operates on sets of plugins. The RCP itself is also a set of plugins running on a module runtime for the widely distributed Java Virtual Machine (JVM).[11] Hence applications developed on top of it run on any system for which a JVM is available.[12] It enables the implementation of Atomic as a set of plugins and their integration into the application platform, which in turn makes it possible for Atomic to interact with, and benefit from, the vast number of plugins available in the Eclipse ecosystem. These include version control system interfaces (for, e.g., git and Subversion), plugins for distributed real-time collaboration, an R development environment, a TEX editor, and many more. By adding one of the version control system interfaces to Atomic, for example, the corpus data itself, the annotations and all metadata can be versioned in atomic detail, which can be utilised for collaborative corpus annotation.[13]

Eclipse is tried and tested technology which has originally been developed by IBM in 2001 and is now under the aegis of the Eclipse Foundation. Due to its long existence and high impact it is supported by a very large community.[14] Eclipse is used by a wide spectrum of disciplines, mostly from IT and the sciences.[15] These parameters make Eclipse a highly sustainable technology, more so than any single research project can hope to achieve.

Atomic's extensibility is further enhanced through its use of the Salt data model, whose feature set (cf. 2.2) in combination with the capabilities of the Eclipse RCP allow for the creation of very diverse extensions for Atomic, such as for the annotation of historic text, or of speech data. Salt has successfully been used, for example, as an intermediate model for the data used in the RIDGES[16] project, which was possible because the model allows for different text segmentations over the same tokens in the context of the representation of different transcription and annotation levels. And as Salt supports audio and video data sources in addition to textual sources, it has also been used as an intermediate model for the dialogue data of the BeMaTaC[17] project.

2.2 Data model

Atomic's domain model is Salt (Zipser and Romary, 2010), a graph-based metamodel for linguistic data. Salt's generic nature and general lack of semantics makes it independent of specific linguistic analyses, tagsets, annotation schemes and theories, and its graph-based nature allows for the modeling of nearly all conceivable kinds of linguistic structures as nodes and edges. Tokens, spans, hierarchies, and primary texts are all represented as nodes. There can be an unlimited number of edges between nodes, which permits the creation of very diverse types of structures, such as co-reference chains, dependencies, constituent trees, and simpler morphological token annota-

[11] Eclipse Equinox is an implementation of the OSGi specification for a dynamic component model for the JVM.

[12] This includes all major operating systems including Windows, Mac OS X, and Linux.

[13] In Atomic's current iteration, this would be achieved by versioning a corpus document's SaltXML files, cf. 2.2.

[14] Eclipse has over 200 open-source projects and "millions of users" under its umbrella (Eclipse Foundation, 2011).

[15] The Eclipse science community is organized in the Science Working Group at the Eclipse Foundation, cf. http://science.eclipse.org.

[16] Register in Diachronic German Science, cf. http://hdl.handle.net/11022/0000-0000-24EC-E.

[17] Berlin Map Task Corpus, http://u.hu-berlin.de/bematac.

tions. An annotation in Salt is represented as an attribute-value pair with an additional optional namespace tag, and is therefore not restricted to specific tagsets. Salt has originally been designed as a main memory model, but could also be mapped to graph-based persistence technologies such as Neo4j.[18] Salt provides a Java API which is open source under the Apache License 2.0. It was designed using the Eclipse Modeling Framework (EMF, Steinberg et al. (2009)). Models can be persisted via an XMI serialisation of the EMF model as SaltXML, a stand-off format which bundles annotations in one file per document. SaltXML is used as Atomic's default persistence format.

A graph-based domain model such as Salt can be mapped to a graphical representation model of annotation graphs for Atomic relatively easily, as this is supported by the underlying technology: Solutions for the creation of editors and visualizations of EMF-based domain models are available in the Eclipse ecosystem. Projects like the Eclipse Graphical Modeling Framework (GMF) and the Eclipse Graphical Editing Framework (GEF (Rubel et al., 2011), used for Atomic's annotation graph editor, cf. 2.3) have been specifically designed for this purpose.

2.3 Usability and features

User-friendliness starts at the acquisition and installation of software: Atomic is provided as a single zip archive file on the Atomic website,[19] and is available for Linux, Mac OS X, and Windows. No installation as such is necessary, simply extracting the archive to a directory of choice suffices. Unlike other tools – including locally installed web applications –, Atomic is self-contained inasmuch as no further dependencies such as databases, server backends, etc. have to be installed, and the Eclipse RCP plugins are included in the distributed zip file.

At its heart, Atomic consists of a workspace-driven navigator; a document editor for overview, basic annotation and segmentation of corpus documents; a graphical editor for mouse-and-keyboard-based annotation; a command-line shell for rapid annotation with the native annotation language AtomicAL (Figure 1). Additionally, the current version of Atomic includes a dedicated editor for co-reference annotations.

The navigation view provides an interface to the user's workspaces as well as the usual project management features.

The document editor is a simple, text-based overview of a corpus document's primary text. It can be used for token-based annotation, segmentation of a corpus document into processable units, and navigation of a document. The editor is the initial entry point for annotation work. Subsequently, the user is forwarded to the annotation graph editor for further, more granular annotation of the selected corpus segment, span, or sentence.

The annotation graph editor – which offers the user editing facilities based on a graphical representation of the complete, if relatively abstract, annotation graph – is implemented on top of the Eclipse Graphical Editing Framework, and provides intuitive, mouse-based annotation with support for a set of hotkeys for advanced users. The use of well-established graphical user interface metaphors in the editor, such as the tools palette, make it easy for less experienced users to build sophisticated annotation graphs quickly. More experienced annotators can resort to a command-line shell driven by the Atomic Annotation Language (AtomicAL), a data manipulation language for annotation graphs originally developed for use in the GraphAnno annotation tool.[20] AtomicAL enables rapid annotation, as it works with one-char commands (e.g., *a* for "**a**nnotate this element", or *p* for "group these elements under a new **p**arent"), followed by optional flags (e.g., for changing the annotation level), a list of target elements, and a list of annotations.[21] Additionally, annotation options can be restricted by annotating against freely configurable tagsets, defined by the user via project-specific preferences.

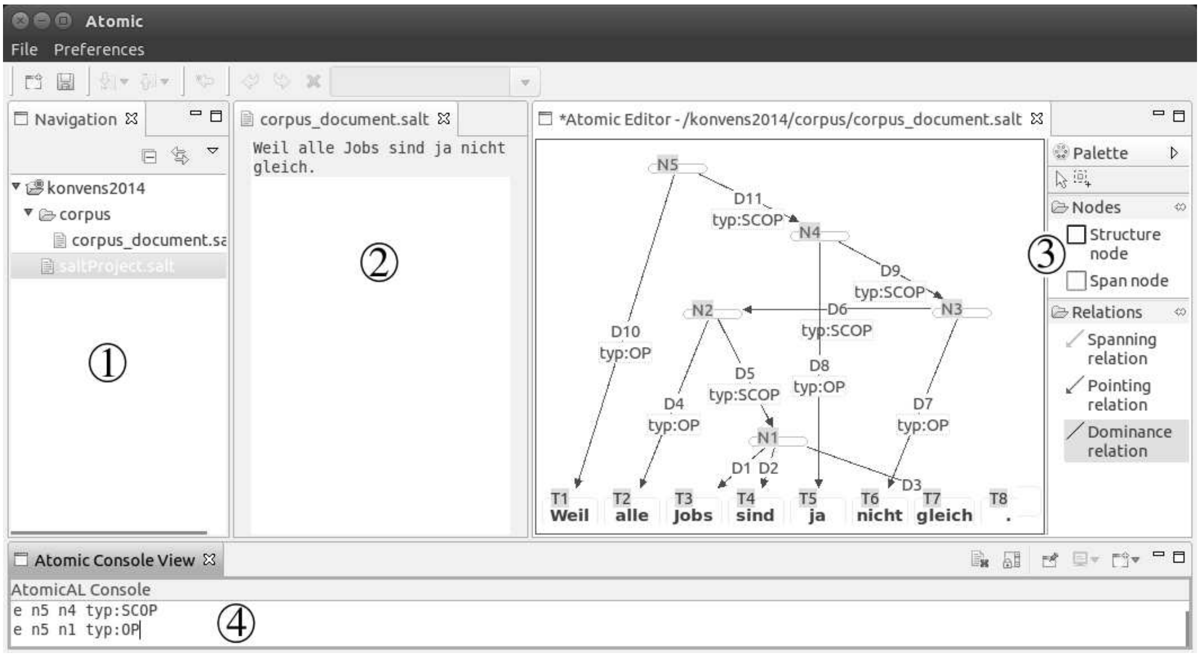
While the annotation graph editor is an all-purpose editor which allows for annotation on arbitrary levels, it may be imperfectly suited for specific annotation tasks. It is therefore desir-

[18] Cf. http://www.neo4j.org/.

[19] http://linktype.iaa.uni-jena.de/atomic.

[20] URL: http://linktype.iaa.uni-jena.de/?nav=graph-anno.

[21] E.g., commands like "On the syntax level, create a new syntactical structure, assign it the category *VP*, and create dominance relations from it to tokens T1, T2, T3, T4, and T5" can be expressed as `p -ls t1..t5 cat:VP`.



```
Weil alle Jobs sind ja nicht
gleich.
```

```
e n5 n4 typ:SCOP
e n5 n1 typ:OP
```

Figure 1: Atomic's application window in a typical configuration: ① Navigation view, ② document editor, ③ graphical annotation editor, ④ AtomicAL command-line shell.

able to include further editors in Atomic which specialise in such tasks, e.g., annotating on only one specific annotation layer or a set of layers. Some tools for example, which are not actively developed anymore, have fulfilled these tasks to the user's satisfaction, who in turn will be well-accustomed to them.[22] It therefore suggests itself to re-create the functionality of these tools in an extension to Atomic, and the above-mentioned co-reference editor is such an attempt.

2.4 Compatibility

For a newly developed multi-level annotation tool, compatibility with existing tools will have a major impact on its half-life and sustainability, and is therefore an indispensible requirement, as corpora should be easily transferrable between the existing tools and the new one. A corpus originally created with, e.g., EXMARaLDA should be importable into Atomic for further annotation, as should be a corpus which has been pre-annotated in an NLP pipeline such as WebLicht (Hinrichs et al., 2010), in order to correct or enhance the annotations. Furthermore, if the annotators are comfortable with using a certain tool for certain annotations, they should still be able to use Atomic for those annotation levels that their favourite tool does not support. As mentioned above, Atomic includes the Pepper framework to tackle this problem area. It is a universal format conversion framework which follows an intermediate model approach, with the Salt meta-model as the intermediate model. To convert data from format X to format Y, X is mapped to Salt and subsequently mapped to Y. The detour via Salt reduces the number of mappings that need to be implemented from $n^2 - n$ mappings (for direct mapping) to $2n$.

Pepper, like Atomic, is plugin-based and comes with a lot of modules realizing such mappings, for instance for EXMARaLDA, TigerXML, tiger2, PAULA, MMAX, Penn Treebank, TreeTagger's XML format, CoNLL, RST, the ANNIS format, and many more.[23] Since Pepper and Atomic share the same data model – Salt –, it has been easy to

[22] Examples include the MMAX2 (Müller and Strube, 2006) co-reference editor, and @nnotate (Plaehn, 1998) for syntax annotations.

[23] This also includes modules for processing TCF (cf. http://korpling.github.io/pepperModules-TCFModules/), an XML format developed within the WebLicht architecture and used by WebAnno as interchange format. The Pepper TCF modules, therefore, provide Atomic with compatibility to WebAnno, as data processed in the latter can be imported into Atomic.

integrate it into Atomic, which in turn provides import and export wizards for all the existing formats. Thus, Atomic is equipped with support for all of the formats for which a mapping exists in Pepper, making it compatible with a wide variety of existing linguistic annotation and search tools. Support for further data formats can be achieved by developing Pepper import and/or export modules for the desired format.

3 Outlook

Following the initial release of Atomic, and some feedback and optimization iterations, we plan to enhance Atomic with additional editors for specific annotation types. Additionally, Atomic should integrate NLP pipelines like UIMA (Ferrucci and Lally, 2004), WebLicht, etc., to provide semi-automatic workflows from within the tool. We also plan to embed support for distributed collaboration via one of the above-mentioned existing plugins.

4 Conclusion

Software for multi-level corpus annotation is subject to a number of requirements. It should operate on a generic data model to allow for potentially unlimited types of annotations, be easily extensible so that new types of annotations and the tooling required for them can be added to it by third parties, and be compatible to other software and data formats in order to make it usable for enriching pre-annotated corpora with additional annotation levels. Additionally, it should be accessible to users with different levels of experience in corpus annotation, as some annotations may be provided not only by corpus linguists, but also by less experienced annotators.

In this paper we have introduced Atomic, an open-source desktop application based on the Eclipse Rich Client Platform which aims at fulfilling the above-mentioned requirements. It does so by operating on the generic graph-based data model Salt, whose lack of semantics allows for potentially unlimited types of annotations, which in Salt are modeled as nodes and edges. Atomic's architecture allows for ease of extensibility through its use of the Eclipse RCP plugin framework, and by incorporating the converter framework Pepper provides compatibility with a wide range of corpus and annotation formats. The tooling making up the core of Atomic is capable of accommodating a user base with potentially diverse levels of experience in corpus annotation: Easily accessible tools such as the annotation graph editor with its tools palette and point-click-and-type workflow are made available for less experienced annotators, while expert annotators can resort to a command-line interface powered by the native data manipulation language AtomicAL.

Atomic complements an ecosystem of software for corpus linguistics, affiliated through a shared data model and a conversion framework based on it respectively. This, together with its high degree of extensibility, makes Atomic a potentially very sustainable tool.

Despite its ready-for-use set of features for multi-level corpus annotation, Atomic is not complete, finalised software. It rather intends to be a platform for corpus annotation tooling upon which the community can build customized solutions for specific research questions as well as feature-complete tools for more general annotation tasks.

References

Dipper, Stefanie, Götze, Michael, and Skopeteas, Stavros (eds.). 2007. *Information Structure in Cross-Linguistic Corpora: Annotation Guidelines for Phonology, Morphology, Syntax, Semantics, and Information Structure*. Interdisciplinary Studies on Information Structure 7, special issue.

Dybkjær, Laila; Møller, Morten Baun; Bernsen, Niels Ole; Carletta, Jean; Isard, Amy; Klein, Marion; McKelvie, David and Mengel, Andreas. 1999. *The MATE Workbench*. In: Proceedings of ACL-1999, demo session, University of Maryland, June 1999, 12-13.

Eclipse Foundation. 2011. *The open-source Developer Report. 2011 Eclipse Community Survey*. URL: http://www.eclipse.org/org/community_survey/Eclipse_Survey_2011_Report.pdf

Ferrucci, David and Lally, Adam. 2004. *UIMA: An Architectural Approach to Unstructured Information Processing in the Corporate Research Environment*. Cambridge University Press, Cambridge.

Hinrichs, Marie; Zastrow, Thomas and Hinrichs, Erhard W. 2010. *WebLicht: Web-based LRT Services in a Distributed eScience Infrastructure*. LREC, European Language Resources Association.

Krause, Thomas; Lüdeling, Anke; Odebrecht, Carolin; Romary, Laurent; Schirmbacher, Peter; Zielke, Dennis. 2014. *LAUDATIO-Repository: Accessing a heterogeneous field of linguistic corpora with the help of an open access repository*. Digital Humanities 2014 Conference. Poster Session. July 2014, Lausanne.

Lüdeling, Anke and Hirschmann, Hagen. forthcoming. *Error Annotation*. In: Granger, Sylviane; Gilquin, Gaetanelle and Meunier, Fanny (eds.). The Cambridge Handbook of Learner Corpus Research. Cambridge University Press, Cambridge.

Lüdeling, Anke; Ritz, Julia; Stede, Manfred and Zeldes, Amir. forthcoming. *Corpus Linguistics*. In: Fery, Caroline and Ishihara, Shinishiro (eds.), OUP Handbook of Information Structure, Oxford University Press, Oxford.

McAffer, Jeff; Lemieux, Jean-Michel and Aniszczyk, Chris. 2010. *Eclipse Rich Client Platform*. 2nd ed. Addison-Wesley, Boston.

Müller, Christoph and Strube, Michael. 2006. *Multilevel annotation of linguistic data with MMAX2*. In: Braun, Sabine; Kohn, Kurt and Mukherjee, Joybrato (eds.). Corpus Technology and Language Pedagogy: New Resources, New Tools, New Methods. Peter Lang, Frankfurt/Main, Germany.

Pajas, Petr and Štěpánek, Jan. 2004. *Recent Advances in a Feature-Rich Framework for Treebank Annotation*. In: Proceedings of the 22nd International Conference on Computational Linguistics. Manchester, 673-680.

Plaehn, Oliver. 1998. *Annotate Programm-Dokumentation (NEGRA Project Report)*. Universität des Saarlandes, Saarbrücken, Germany.

Rubel, Dan; Wren, Jaime and Clayberg, Eric. 2011. *The Eclipse Graphical Editing Framework (GEF)*. Addison-Wesley, Boston.

Schmidt, Thomas. 2004. *Transcribing and annotating spoken language with EXMARaLDA*. In: Proceedings of the LREC-Workshop on XML-based richly annotated corpora, Lisbon 2004, ELRA, Paris.

Steinberg, David; Budinsky, Frank; Paternostro, Marcelo and Merks, Ed. 2009. *EMF: Eclipse Modeling Framework 2.0*. Addison-Wesley, Boston.

Yimam, Seid Muhie; Gurevych, Iryna; Eckart de Castilho, Richard; and Biemann Chris. 2013. *WebAnno: A Flexible, Web-based and Visually Supported System for Distributed Annotations*. In: Proceedings of ACL-2013, demo session, Sofia, Bulgaria.

Zeldes, Amir; Ritz, Julia; Lüdeling, Anke and Chiarcos, Christian. 2009. *ANNIS: A Search Tool for Multi-Layer Annotated Corpora*. In: Proceedings of Corpus Linguistics 2009, July 20-23, Liverpool, UK.

Zipser, Florian and Romary, Laurent. 2010. *A model oriented approach to the mapping of annotation formats using standards*. In: Proceedings of the Workshop on Language Resource and Language Technology Standards, LREC 2010, Malta.

Zipser, Florian; Zeldes, Amir; Ritz, Julia; Romary, Laurent; Leser, Ulf. 2011. *Pepper: Handling a multiverse of formats*. In: 33. Jahrestagung der Deutschen Gesellschaft für Sprachwissenschaft, Göttingen, February 2011.

WebNLP – An Integrated Web-Interface for Python NLTK and Voyant

Manuel Burghardt, Julian Pörsch, Bianca Tirlea, & Christian Wolff
Media Informatics Group, University of Regensburg
{manuel.burghardt,christian.wolff}@ur.de
{julian.poersch,bianca.tirlea}@stud.uni-regensburg.de

Abstract

We present *WebNLP*, a web-based tool that combines natural language processing (NLP) functionality from *Python NLTK* and text visualizations from *Voyant* in an integrated interface. Language data can be uploaded via the website. The results of the processed data are displayed as plain text, XML markup, or Voyant visualizations in the same website. WebNLP aims at facilitating the usage of NLP tools for users without technical skills and experience with command line interfaces. It also makes up for the shortcomings of the popular text analysis tool Voyant, which, up to this point, is lacking basic NLP features such as lemmatization or POS tagging.

1 Introduction

Modern corpus linguistics has been on the rise since the late 1980s (Hardie, 2012), largely because of the availability of vast amounts of digital texts and computer tools for processing this kind of data. Since then, corpus linguistics has produced a number of important subfields, such as *web as a corpus* (cf. Kilgarriff and Grefenstette, 2003; Baroni et al., 2009), *language in the social media* (cf. Beißwenger and Storrer, 2009) or using language data for *sentiment and opinion mining* (cf. Pak and Paroubek, 2010). More recently it has been claimed that the mass of digital text available for automatic analysis constitutes a new research paradigm called *culturomics* (Michel et al., 2010) and that the recent arrival of the *digital humanities* opens up additional fields of application for corpus linguistics and text mining. Taking the increased amount of digital text data which is readily available into consideration, Gregory Crane has asked the well justified question “what to do with a million books” (Crane, 2006). The question is partially answered by Moretti (2013), who introduces the idea of *distant reading* of texts, as opposed to the more traditional, hermeneutic *close reading*, which is particularly popular in the field of literary studies. The idea of *distant reading* suggests to interpret literary texts on a more generic level by aggregating and analyzing vast amounts of literary data.

All these novel types of applications require basic NLP analysis such as tokenization, lemmatization, POS tagging, etc. Currently, there is no lack of adequate tools than can be used to process large amounts of text in different languages. Prominent examples are GATE (*General Architecture for Text Engineering*)[1] or the UIMA framework (*Unstructured Information Management Infrastructure*)[2]. However, most of these tools can be characterized as having a fairly high entry barrier[3], confronting non-linguists or non-computer scientists with a steep learning curve, due to the

[1] Available at https://gate.ac.uk; all web resources described in this article were last accessed on May 4, 2014.

[2] Available at http://uima.apache.org

[3] Hardie (2012) gives a short overview of the development of corpus analysis tools while at the same time discussing their usability requirements.

fact that available tools are far from offering a smooth *user experience* (UX). This may possibly be caused by complex interaction styles typically encountered in command line interfaces, by suboptimal interface design for *graphical user interfaces* (GUIs) or by the necessity of bringing together disparate tools for a specific task.

Nowadays, a decent UX is a basic requirement for the approval of any application such as office tools or smartphone apps (Nielsen and Budiu, 2013). At the same time, a large and well accepted body of knowledge on *usability* and *user centered design* (cf. Shneiderman, 2014) is at our disposal. However, tools developed for scientific purposes like corpus linguistics or text mining do not seem to take advantage of these knowledge sets: It appears that many tools are designed by scientists who may have acquired the necessary programming and software engineering skills, but who are lacking experience and training in user interface design and usability engineering. As a result, many tools are functionally perfect, but an obvious mess as far as usability aspects are concerned.

In the following, we will not introduce yet another tool, but we rather try to provide an integrated, easy-to-use interface to existing NLP and text analysis tools.

2 Tools for NLP and text analysis

There are a number of available tools that can be used for NLP tasks and quantitative text analysis (cf. the notion of *distant reading*). This section introduces some of the most prominent tools, and also makes the case for the newly created *WebNLP* prototype.

2.1 Python NLTK

Python NLTK[4] (Bird, 2006) is a widely used toolkit that allows the user to perform sophisticated NLP tasks on textual data and to visualize the results. One drawback of NLTK, however, is its command line interface. Also, a basic understanding of the programming language *Python* is necessary for using it. Depending on the target platform, setting up the NLTK environment can be rather cumbersome. For these reasons, many humanities scholars who are lacking technical skills in Python and command line interfaces may refrain from using NLTK as a means for NLP.

2.2 TreeTagger

TreeTagger[5] (Schmid, 1994), another widely used NLP tool, tries to address this issue by providing a GUI (only available for Microsoft Windows)[6]. The output of the tool can however not be visualized in the same GUI.

2.3 Voyant Tools

Voyant[7] (cf. Ruecker et al., 2011) is a web-based tool that is very popular in the digital humanities community. It allows the user to import text documents and performs basic quantitative analysis of the data (word count, term frequency, concordances, etc.). The results of this analysis are visualized in the browser, e.g. as KWIC lists, word clouds or collocation graphs. While the tool is easy to use via a modern web browser, Voyant is lacking a feature to perform basic NLP operations (e.g. lemmatization) on the data before it is analyzed.

2.4 The case for WebNLP

It shows that many of the existing tools are either not accessible to non-technical users due to their technical complexity, or that they are lacking important functionality. The goal of this work is to provide an easy-to-use interface for the import and processing of natural language data that, at the same time, allows the user to visualize the results in different ways. We suggest that NLP and data analysis should be combined in a single interface, as this enables the user to experiment with different NLP parameters while being able to preview the outcome directly in the visualization component of the tool. We believe that the immediate visualization of the results of NLP operations makes the procedure more transparent for non-technical users, and will encourage them to utilize NLP methods for their research.

[4] Available at `http://www.nltk.org/`

[5] Available at `http://www.cis.uni-muenchen.de/~schmid/tools/TreeTagger/`

[6] Available at `http://www.smo.uhi.ac.uk/~oduibhin/oideasra/interfaces/winttinterface.htm`

[7] Available at `http://voyant-tools.org/`

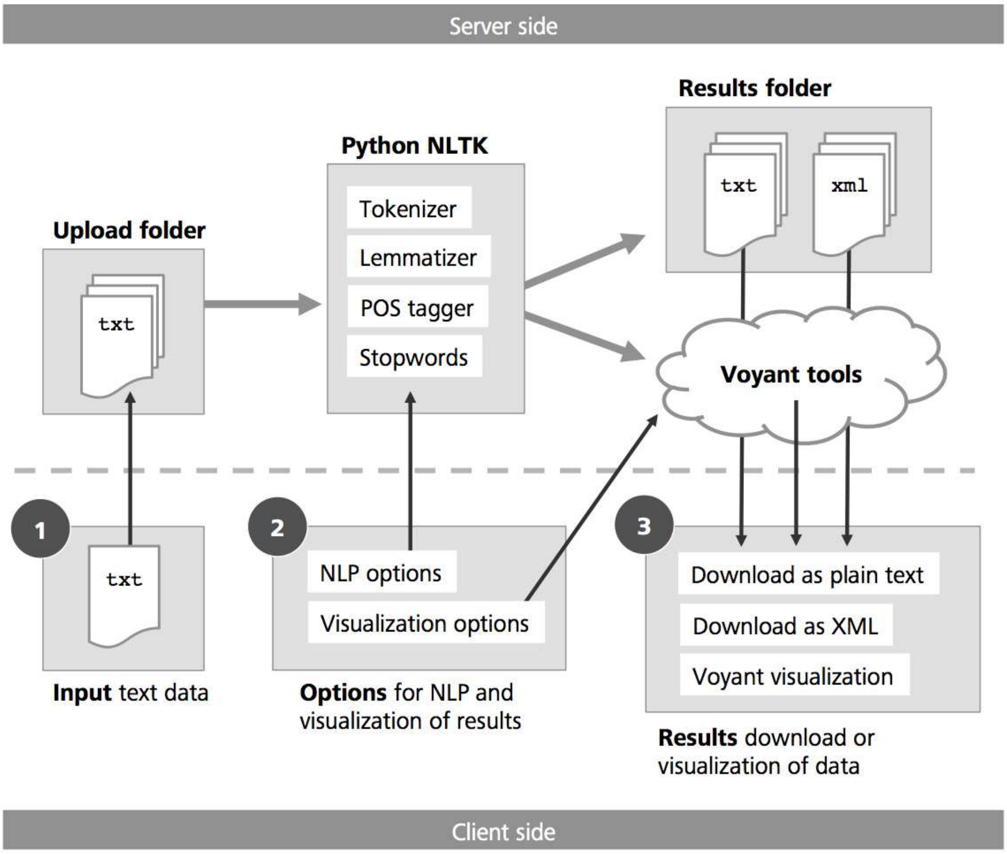


Figure 1: WebNLP architecture and main components.

In order to achieve this goal, we integrate two existing tools (Python NLTK and Voyant) in a combined user interface named *WebNLP*[8].

3 WebNLP

In this section we describe the basic architecture of WebNLP and explain the main functions and interface components of the tool.

3.1 Tool architecture

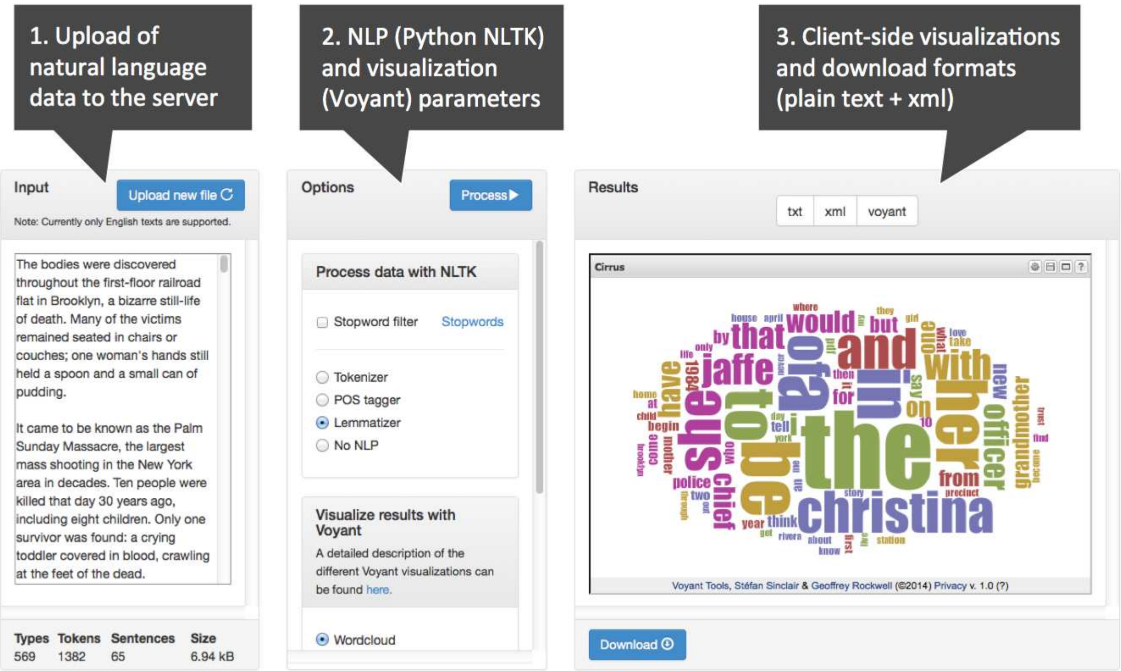
We decided to implement the interface as a web service for several reasons:

- No installation or setup of Python NLTK and related Python modules by the user is required.
- Previous experience and familiarity of non-technical users with web services and interactive elements such as *form fields*, *radio buttons*, etc.
- Seamless integration of the existing web tool Voyant, which allows the user to quickly analyze and visualize language data in the browser.
- Opportunities for future enhancements of the tool, e.g. collaboration with other users, sharing of data and results, etc.

WebNLP uses a client-server architecture to provide an easy-to-use interface via modern web browsers, while the NLP functions are executed on our server (cf. Figure 1). The interface on the client side is structured in three main areas (cf. Figure 2) which will be explained in more detail in the next section. All interface logic is implemented by means of *JavaScript*, the page layout utilizes a template from the popular front-end framework *Bootstrap*[9]. The communication between client and server is realized by means of *PHP* and *AJAX*.

[8] WebNLP is currently available as a prototype at `http://dh.mi.ur.de/`

[9] Bootstrap is available at `http://getbootstrap.com/`.



1. Upload of natural language data to the server

2. NLP (Python NLTK) and visualization (Voyant) parameters

3. Client-side visualizations and download formats (plain text + xml)

Note: Currently only English texts are supported.

The bodies were discovered throughout the first-floor railroad flat in Brooklyn, a bizarre still-life of death. Many of the victims remained seated in chairs or couches; one woman's hands still held a spoon and a small can of pudding.

It came to be known as the Palm Sunday Massacre, the largest mass shooting in the New York area in decades. Ten people were killed that day 30 years ago, including eight children. Only one survivor was found: a crying toddler covered in blood, crawling at the feet of the dead.

Types	Tokens	Sentences	Size
569	1382	65	6.94 kB

Process data with NLTK

□ Stopword filter   Stopwords

○ Tokenizer
○ POS tagger
● Lemmatizer
○ No NLP

Visualize results with Voyant

A detailed description of the different Voyant visualizations can be found here.

● Wordcloud

Figure 2: WebNLP interface with three main areas: input, options, results.

A number of Python NLTK scripts (e.g. for tokenization, lemmatization, etc.) can be called from the client interface and are then executed on the server. The results are displayed on the client side by calling different visualization forms of the web service Voyant, which is embedded in the WebNLP interface as an HTML *iframe*. At the same time, the NLTK processed data is stored on the server as plain text or as text with XML markup, which are both available for download on the client side.

3.2 Input: Upload of natural language data

The input field allows the user to upload text documents to the NLP application on the server. Data may either be entered directly in the text area form field, or by making use of the file upload dialog. Currently, only files in plain text format (.txt) can be processed by the NLTK tools on our server. Another restriction for the current implementation of the tool is concerned with the language of the text documents: At the moment, only NLTK scripts for processing English language data have been integrated into the tool. However, the system architecture is designed in a modular fashion that allows the administrators to add more NLTK scripts for other languages at a later point in time. Once the data has been uploaded to the server, a first NLTK pre-processing of the data is executed, analyzing the overall number of tokens, types and sentences in the file. This information is displayed at the bottom of the input area after the upload of the file has been completed.

3.3 Options: NLP and visualization parameters

The second area in the interface contains options for the NLP and visualization of the uploaded data. The first set of options selects Python NLTK scripts on the server, that are then executed on the data. In the current tool version, the following main functions are available:

- Stop word filter; can be combined with any other parameter (a list of all stop words may be looked up in the interface)
- Tokenizer (words and punctuation marks)
- Part of speech tagger (tokenization implied)
- Lemmatizer (tokenization implied)
- No NLP (used if no additional NLP processing is needed)

The second group of options allows the user to select a visualization style for the processed data from Voyant. The following visualization[10] options are available in the current WebNLP prototype:

- Wordcloud
- Bubblelines
- Type frequency list
- Collocation clusters
- Terms radio
- Scatter plot
- Type frequency chart
- Relationships
- No visualization

Due to the internal NLP workflow on the server, currently only one NLP and one visualization option can be selected at a time. We are planning to implement a more flexible solution in the next version of WebNLP.

A short evaluation with a sample of five text documents with different file sizes indicates an almost linear increase of processing time related to text size. The smallest of the test documents had a size of 50 kB (approx. 11.000 tokens), the largest document had a size of 4230 kB (approx. 920.000 tokens). POS tagging for the smallest document took 18 seconds, lemmatization took 20 seconds. For the largest document, POS tagging took approx. 24 minutes, lemmatization took approx. 25 minutes. These results indicate that WebNLP in its current implementation is well-suited for small to medium sized corpora, but may be too slow for larger text collections.

3.4 Results: Client-side visualizations and download formats

The third interface area displays the results of the chosen NLP options in the selected Voyant visualization (e.g. word cloud view). The user may also switch to plain text or XML markup view of the results (these formats are also available for download).

Plain text view (original NLTK output):

```
( VBN , come )
    ...
```

XML view (custom WebNLP format):

```
<root>
  <token>
    <pos>VBN</pos>
    <word>come</word>
  </token>
    ...
</root>
```

4 Conclusions

Our tool provides access to existing NLP and visualization tools via a combined interface, thus acting as a GUI wrapper for these applications. While a thorough usability evaluation is still missing, we are confident that NLP functionality from the Python NLTK becomes more accessible through WebNLP, and that the combination with visualizations from the Voyant set of tools will be attractive for many applications of text technology. In its current implementation, WebNLP should be treated as a prototype that illustrates how a web-based interface to basic NLP and text visualization functions can be realized by means of standard web technologies. We are, however, planning to implement more NLTK functions, and to improve the performance as well as the interface of the service in the future.

References

Marco Baroni, Silvia Bernardini, Adriano Ferraresi, and Eros Zanchetta. The wacky wide web: a collection of very large linguistically processed web-crawled corpora. *Language resources and evaluation*, 43(3):209–226, 2009.

Michael Beißwenger and Angelika Storrer. Corpora of Computer-Mediated Communication. In Anke Lüdeling and Kytö Merja, editors, *Corpus Linguistics. An International Handbook*, pages 292–308. Mouton de Gruyter, Berlin, New York, 2009.

[10] A detailed description of the different Voyant visualization types can be found at `http://hermeneuti.ca/voyeur/tools`.

Steven Bird. Nltk: the natural language toolkit. In *Proceedings of the COLING/ACL on Interactive presentation sessions*, pages 69–72. Association for Computational Linguistics, 2006.

Gregory Crane. What do you do with a million books? *D-Lib Magazine*, 12(3), 2006.

Andrew Hardie. Cqpweb – combining power, flexibility and usability in a corpus analysis tool. *International Journal of Corpus Linguistics*, 17(3):380–409, 2012.

Adam Kilgarriff and Gregory Grefenstette. Introduction to the special issue on the web as corpus. *Computational linguistics*, 29(3):333–347, 2003.

Jean-Baptiste Michel, Yuan Kui Shen, Aviva P. Aiden, Adrian Veres, Matthew K. Gray, The Google Books Team, Joseph P. Pickett, Dale Hoiberg, Dan Clancy, Peter Norvig, Jon Orwant, Steven Pinker, Martin A. Nowak, and Erez Lieberman Aiden. Quantitative analysis of culture using millions of digitized books. *Science*, 331(6014):176–182, 2010.

Franco Moretti. *Distant reading*. London: Verso, 2013.

Jakob Nielsen and Raluca Budiu. *Mobile usability*. New Riders, Berkeley, CA, 2013.

Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. In *Proceedings of the LREC*, pages 1320–1326, 2010.

Stan Ruecker, Milena Radzikowska, and Stéfan Sinclair. *Visual interface design for digital cultural heritage: A guide to rich-prospect browsing*. Ashgate Publishing, Ltd., 2011.

Helmut Schmid. Probabilistic part-of-speech tagging using decision trees. In *Proceedings of international conference on new methods in language processing*, pages 44–49. Manchester, UK, 1994.

Ben Shneiderman. *Designing the user interface: strategies for effective human-computer interaction*. Pearson, 5th edition, 2014.

Data Exploration of Sentence Structures and Embellishments in German texts: Comparing Children's Writing vs Literature

Rémi Lavalley
Cooperative State University
Karlsruhe
Germany
lavalley@dhbw-karlsruhe.de

Kay Berkling
Cooperative State University
Karlsruhe
Germany
berkling@dhbw-karlsruhe.de

Abstract

It is of interest to study sentence construction for children's writing in order to understand grammatical errors and their influence on didactic decisions. For this purpose, this paper analyses sentence structures for various age groups of children's writings in contrast to text taken from children's and youth literature. While valency differs little between text type and age group, sentence embellishments show some differences. Both use of adjectives and adverbs increase with age and book levels. Furthermore books show a larger use thereof. This work presents one of the steps in a larger ongoing effort to understand children's writing and reading competences at word and sentence level. The need to look at variable from non-variable features of sentence structures separately in order to find distinctive features has been an important finding.

1 Introduction

Reading and writing are core competencies for success in any society. In Germany, the *Program for International Student Assessment* (PISA) study and the *Progress in International Reading Literacy Study* (PIRLS) (Bos, 2004) have shown that around 25% of German school children do not reach the minimal competence level necessary to function effectively in society by the age of 15. While the average performance is on par with other OECD countries, Germany falls short on higher levels of achievement and demonstrates a growing heterogeneity between genders and social backgrounds (Prenzel et al., 2013). Analyzing the types of errors that children make in their texts (Berkling and Reichel, 2014) it has been found that in the upper grades many grammatical issues persist that may be an indicator for the problems that become apparent in the above studies. It is therefore important to understand progression in sentence difficulty and its impact on didactics. Looking at research on sentence difficulty and text leveling, extensive research has been published for the English language. There are a number of works on defining sentence complexity or readability (Glöckner et al., 2006), (DuBay, 2008), (Sitbon and Bellot, 2008), (Benjamin, 2012), (Nelson et al., 2012), (Vajjala and Meurers, 2014). Sentence length, adverbs, morphemes, lexical analysis are some of a large number features that are used. Very often these features however do not represent the order in which the words appear in the text. Only few authors look at sentence structure and parse tree architectures (Schwarm and Ostendorf, 2005). In contrast, for German very few studies on this subject can be cited (Bamberger and Vanecek, 1984), (Hancke et al., 2012). Classifiers use some of the same features that had been used for English to classify difficulty levels of texts into major categories (child vs. adult writing). However, at this time, an automated categorization of reading texts for German does not exist. While there are some rules on readability, these are not defined at fine grained levels with a clear progression and therefore also not automated or tested in a systematic manner on readers.

Given the existing body of knowledge, it became clear that some fundamental research is needed in looking at the sentence construction in data before moving on to a discussion about difficulty levels. This paper therefore presents a systematic approach to automatically analyse existing texts. The first goal is to gain a deeper understanding of German sentence structure and its occurrence patterns in different types of written texts, namely children's literature and children's writing for beginners, fourth graders and eight graders.

After an Introduction, Section 2 will review the structure of the German sentence. Section 3 will detail the data that was used for the exploration. Section 4 and 5 describe the automatic processing of the data. Section 6 will present results. Section 7 draws conclusions for future work.

2 Parsing German Sentence Structure

In order to understand how sentences will be analyzed, this section will review German sentence structures, verb valency and adjective and adverbial embellishments.

2.1 Features Description

The following list denotes the German standard sentence structures:

V2: This is the most common structure in German language. The verb is in second position. The subject can be either in first position (*Er* ***arbeitet*** *viel. He* ***works*** *a lot.*), or placed after the verb if the first position is used by something else, such as an adverb (*Jetzt* ***arbeitet*** *er. He* ***is working*** *now.*) or interrogative word (*Wo arbeitet er? Where does he work?*). Some words are not counted in order to determine the verb position. For instance, coordinate conjunctions (*Und er* ***arbeitet****. And he* ***works***). In this case, the verb is considered to be in second position.

V1: The verb is in first position. This structure is generally used for the imperative form (*Sei ruhig! Be quiet!*) or for the interrogative form without interrogative word (*Hast du Hunger? Are you hungry?*)

VE: The verb is in final position. Generally used in subordinate clauses (*Ich denke,* ***dass*** *er zu viel arbeitet. I think* ***that*** *he is working too much.*)

NV: nominal clauses.

2.2 Valency

Verb valency refers to the number of arguments required by a verbal predicate. It includes the subject as well as the objects of the verb. (Ágel and Fischer, 2010). For the purpose of this work, only the items that have actually been attached are considered. The following example demonstrates this: *Lola gave her book.*, vs. *Lola gave her book to Lio.*. While the maximum number of arguments (theoretical valency) for the verb *to give* is 3: Subject (Lola), direct object (her book), indirect object (Lio), in the first sentence the 'used valency' is 2.

2.3 Adverbs and Adjectives

Carefully used adjectives and adverbs can be an indicator of a writers' command of the language. For example, *The little dark blue Smurf with glasses is really embarrassed* can be considered more descriptive writing than simply *The Smurf is embarrassed.* The study of adverbials complements that of verb valency. Consider for instance the two following sentences: *Ich gehe in die Schule. - I go to the school.* vs. *Jetzt gehe ich in die Schule. - I go to the school now.*. In both cases Valency equals 2 (*Ich - I*, *in die Schule - to the school*), while the feature of counting adverbs provides additional information about the usage of a temporal adverbial as *embellishment* to the original sentence construction.

3 DATA

The data chosen for this study comes from the Karlsruhe Database of children's writing (Berkling et al., 2014) and a selection of children's books.

3.1 Texts for Children (Books)

The corpus of literature was obtained through a random selection of books that are commonly read (as defined by the local public library) by children at the selected age groups[1]. Only Ger-

[1] **Grades 1 and 2:** Ages tend to be between 6 and 8 (merged into Grade 2); **Grade 4** Ages tend to be between

Grade No.	**2**	**4**	**8**
# books	21	15	11
Sentence length	7.4	9.7	12.1
# sentences kept	935	869	797
# children texts	237	258	245
Sentence length	10.5	13.3	12.5
# sentences kept	869	2133	1698

Table 1: Number of texts, average sentence length and sentences kept per grade, for both corpora

man authors were selected to eliminate effects of translation on quality. From each book, sample pages were selected and digitized resulting in the copurs statistics given in Table 1.

3.2 Text by Children (Childrens' Writings)

The children's data was collected in 2011–2013 from elementary schools and two types of secondary schools, Realschule and Hauptschule. Students' text was elicited in order to obtain an extended amount of freely written texts. The collection includes 1,752 texts from 1,730 students from grade 1 through 8 and is described in detail in a corresponding publication (Berkling et al., 2014).

The data is transcribed both in its original form (with spelling errors) and in a corrected version called target. While the target sentence has correctly written words, the grammatical errors and erroneous sentence structures remain leading to a non-trivial task of sentence structure analysis. For this study a subset of 740 texts written by children from grades 1, 2, 4 and 8 have been considered. The general statistics are summarized in Table 1.

4 Data Preparation

4.1 The Parser

All sentences in the databases were automatically parsed using the Berkeley's parser (Petrov et al., 2006) for German, with *-tokenize* (to use the integrated tokenizer) and *-accurate* (favours accuracy over speed) options. An example of such a parsing looks as follows, for the sentence *Das gibt ein Durcheinander!* (*This is a mess!*):

Output: *((PSEUDO (S (PDS Das) (VVFIN gibt) (NP (ART ein) (NN Durcheinander))) ($. !)))*

4.2 Sentence Decomposition

Given the parser output, a tool was developed to automatically classify the structure of the sentences. While finding the different clauses is generally done by the parser, a few manual rules to overcome the parser errors were added. The tool isolates the different components of a clause (POSTAG word) and stores them in a table in order of occurrence. Some components are thus grouped with higher entity, while others are not: In the example given above: *gibt* is tagged independently (VVFIN gibt) and in (NP (ART ein) (NN Durcheinander)) the parser has recognized a noun-phrase *ein Durcheinander* (*a mess*) and provides information about the different words (article and noun). We considered the external component as an entry in our table. Except in case of Verb phrase (VP), where the tool doesn't consider VP as one component but uses the isolated words information, as for instance, with this sentence:
(PDS Das) (VAFIN hat) (VP (PPER Karolina) (PRF sich) (AVP (ADV schon) (ADV immer)) (VVPP gewünscht)) [...]
In this case it's interesting to have the components of the VP as different entries. To compute the Valency (see Section 2.2), we need to additionally extract *Karolina* (Subject of the verb).

4.3 Data Cleaning

Some sentences were removed from both of the corpora, if they were too short (less than three words) or too long (more than 50 words). These lengths generally resulted from errors in the previous steps, such as transcription or OCR errors (e.g., a missing dot that leads to very long sentences). Analysing the data in a first path resulted in a very large number of different combinations of sentence structures. Given that most types occurred only in few sentences, the analysis will concentrate only on the 22 different structures that occur at least ten times on both of the corpora. Table 1 shows the number of sentences kept for the rest of the work presented here.

9 and 11; **Grade 8 and 8+:** Ages in this grade vary around 14 (merged into Grade 8)

5 Sentence Analysis

5.1 Sentence Structure

A clause structure is determined by the position of the main verb (finite) in the clause, which is tagged as V*FIN (VVFIN, VAFIN for auxiliaries, VMFIN for modal verbs). The tool categorizes a clause according to the structures defined in Section 2.1 by looking for verbs in their position.

5.2 Complex Structure Recognition

Many sentences consist of several clauses. The representation of the entire sentence therefore consists of a combination of classified clauses. The tool thus tags the entire sentence with the following notation scheme for Coordinate Clauses **CC** and Subordinate Clauses **SC** as examplified below.

CC: V2-V2 Ich$_{pos=1}$ mag$_{pos=2}$ das$_{pos=3}$ nicht$_{pos=4}$, aber$_{pos=0}$ ich$_{pos=1}$ gehe$_{pos=2}$ mit$_{pos=3...}$ ihnen ins Kino. *I don't like this, but I go to the cinema with them.*

SC: V2[VE] The verb is in second position in the main clause and in the final position in the subordinate clause. *Ich denke, dass ich ins Kino gehen* werde. *I think that I* will *go to the cinema.* In this case, an auxiliary verb is used in the subordinate clause to build the future tense (werde/will), this one is the conjugated verb and stands at the end of the clause.

SC: V2[VE]# The sharp symbol (#) is used to denote the fact that the subordinate clause occurs before the main clause. *Wenn du mir ein Blatt Papier gibst,* schreibe *ich dir einen Brief. (If you give me a paper, I* write *you a letter.*)

SC: V2[V2]# In this structure there is a main clause and a subordinate one, the subordinate stands before the main clause and they both have verbs in second position. In our corpus, it's mainly related to dialogs (*Ich rufe dich an,* sagt *Lola. I call you,* says *Lola.*). In this example, the subordinate clause is *Ich rufe dich an* (it is what Lola says) and in the main clause, the verb is considered as in second position because the first position is occupied by the subordinate clause.

There can be more than two clauses, such as 3 coordinates, one main clause with two interlocked subordinates (V2[NV[VE]]), one main clause with a subordinate made of two coordinates clauses (V2[VE-VE]), to name a few. The tool can represent all of these combinations.

5.3 Evaluation of Structure Classification

The tool developed for sentence structure analysis has been evaluated on 400 sentences manually annotated: 200 sentences coming from books and 200 from children writings. We also annotated these sentences as correct or not. 20% of the sentences extracted from books contained errors introduced during digitization: Non-existing words, space missing or added, or punctuation marks missing, sentences erroneously merged into one, missing comas, making it difficult to determine clauses (in German, subclauses are separated by commas). 30% of the sentences in children's writing contained errors: Spelling errors not corrected by annotators, grammar errors, such as words in wrong position, usage of an incorrect word (not corrected by annotators), sentences intentionally concatenated into one by the writers (making them difficult to parse). The overall precision of the tool is the same for both corpora: 80% of the sentences correctly labeled. The system has wrongly labeled structures for 38 sentences of the Books corpus (16 of these sentences had at least 3 clauses) and 41 of the children corpus (27 had at least 3 clauses). More than 3 clauses are usually a sign of bad sentence construction and are therefore difficult to parse.

5.4 Valency

The tool computes the number of arguments by going through the table containing the constituents of the sentence. Constituents are counted towards the valency counter as long as they are not excluded given the rules below. These are intended to bypass parser mistakes while keeping it as exhaustive and accurate as possible.

Word: The list denotes a number of POS-tags that cannot be a subject or an object of a verb, such as articles, other verbs (infinite, participles), preposition, adjectives, adverbs, and particles (separable verbs). If a word is not labeled with one of these POS-tags (KOUS, PTK, ADV, KON, V* denoting different types of verbs...) then it is an argument of the verb.

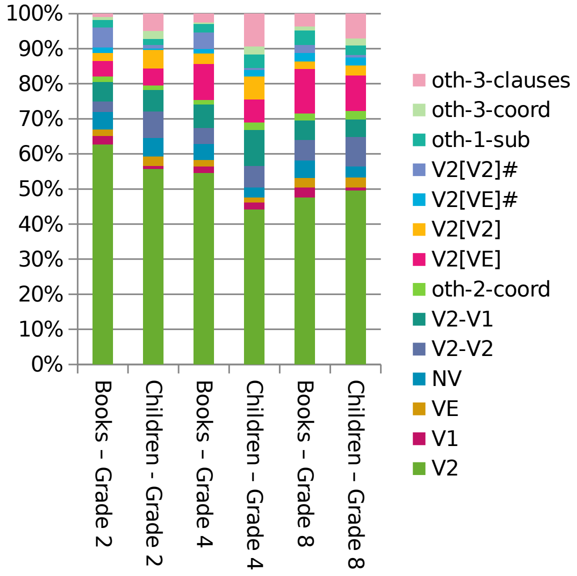


Figure 1: Partition of sentence structures, for both corpora at different grade levels.

Clause: Some clauses can be the object of a verb: *Ich denke, dass er zu viel arbeitet*. *Ich* (*I*) is the subject, while *dass er zu viel arbeitet* (*that he works too much*) is object of the verb *denke* (*think*). Other clauses cannot be objects. For example, *Er ist mehr intelligent, als ich. (He is more intelligent than me)*. These clauses are excluded by using a list of POS-tags (KOUS, KON, PROAV, KOKOM) combined with the list of words that introduce adverbial clauses (”bevor”, ”als”, ”wenn”, ”während”, ”indem”, ”solange”, ”bis”, ”weil”, ”da”, ”wie”, ”damit”, ”obwohl”, ”trotzdem”, ”obgleich”, ”denn”, ”seitdem”).

5.5 Adverbs and Adjectives per Sentence

Adjectives and adverbs are easily detected by the POS tags provided by the parser. The tool counts the number of words labeled as adverbs or adjectives according to the parser.

6 Results

6.1 Sentence Structures

In both corpora the average occurrence frequency of sentence structure per grade was computed as well as average use of adjectives/adverbs per sentence type for each sentence type and grade level. Figure 1 shows the partition of sentences structures by grade and corpora. Less frequent structures were merged into one of four super-categories: **”oth-2-coord”** contains all the sentences made of two coordinates clauses except V2-V1 and V2-V2, **”oth-1-sub”** contains sentences with a main clause and a subordinate clause other than the ones provided separately, **”oth-3-coord”** contains the sentences made of three coordinate clauses and **”oth-3-clauses”** the sentences made of 3 clauses including at least one subordinate.

We can observe the following: From Grade 2 (including Grade 1) to Grade 8, books use a decreasing number of V2 sentences (from 62% to 48%). Meanwhile children always have more or less 50% of their sentences of type V2. Children write a larger number of coordinate clauses (V2-V2, V2-V1, other 2 coordinates and other 3 coordinates) when compared to books. Inspecting the data, it can be seen that children create their own grammar rules and forget to split sentences. As children get older, they use less nominal sentences (NV). The proportion of subordinates clauses with verb ending (V2[VE]) increases with the grades in books (from 4 to 13%) - the same applies for children between 2nd and 8th grade. Children don’t really use inverted clauses (notation ending in #). In books these mainly occur with dialogues (*Ich arbeite nicht, sagt Lola - I don’t work, says Lola*), whereas the topics on which the children had to write didn’t especially involve dialogues even if some can be found in the texts. When children reach 8th grade, they tend to use the same structures that occur in books, i.e. the distribution of sentence types is roughly the same as that of 8th grade published literature.

6.2 Adverbs and Adjectives

Figure 2 depicts the mean number of adverbs used in sentences by books and children for the different structures. The last column is the mean number of adverbs on all the sentences, regardless of their structure. We can observe that children use almost as many adverbs as authors of books. The gap between Grade 2 and Grade 8 is more significant in books’ texts than in childrens’ writings. However, half of sentences have no adverbs at all (in Grade 2, it concerns 53% of children’ sentences and 54% of the books’ ones). This frac-

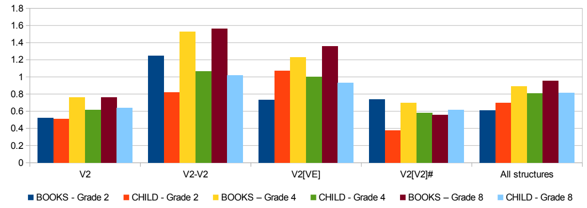


Figure 2: Mean number of adverbs per sentence for selected structures, for both of the corpora (Books or Children) at the different grades (2, 4, 8).

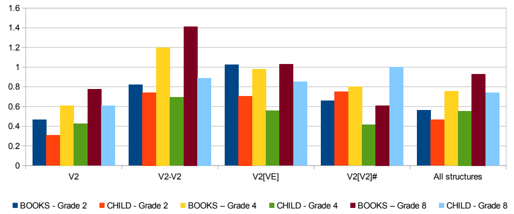


Figure 3: Mean number of adjectives per sentence for selected structures, for both of the corpora (Books or Children) at the different grades (2, 4, 8).

tion decreases with higher grades for both children (48% in 8th grade) and books (45%). Generally, the children have more sentences without adverbs than authors. The same observations can be made regarding the adjectives: in 2nd grade, 58% of books' sentences and 63% of children's ones don't have adjectives at all, whereas in 8th grade, these ratios are respectively 45% and 48%.

Figure 3 depicts the mean number of adjectives per sentence for different sentence structures. Such as for adverbs, children use them a little bit less than authors do, but more and more as they are getting older. The mean number of adjectives increased by 50% between Grade 2 and Grade 8.

6.3 Valency

According to our analysis, valency seems roughly invariant to age and the two text corpora used. The only kind of sentences on which significant differences have been observed is the V2 type. As shown in Table 2, children in Grade 2 have a different usage of objects compared to books: 44% of their verbs have only one complement (i.e., generally the subject), while this proportion is only 31% in books. Whereas this number slightly decreases in books to reach 22% in 8th grade, the children use really less constructions of this type compared to their early ages to reach 25% of their sentences, which is close to the proportion observed in books. Accordingly, the global repartition between the different valencies of verbs is the same for books and children's writings when they reach 8th grade.

	Books		Children	
	G 2	G 8	G 2	G 8
Val=1	31%	22%	44%	25%
Val=2	55%	58%	45%	50%
Val=3	12%	17%	9%	19%
Val=4	2%	4%	0%	4%

Table 2: Proportion of V2 sentences having Valency = 1 to 4, for both of the corpora, at grades 2 and 8.

7 Conclusion and Future work

The goal of this work is a systematic approach to automatically analyze large amounts of texts and their structures to gain a deeper understanding on tackling text difficulty. Rules to recognize typical German sentence structures were implemented based on the output of an open source POS-tagger. Looking at texts written by and for children, the sentences were analyzed based on the occurrence distribution of particular structures within the texts at different grade levels. In addition, embellishments clues (valency, adjectives and adverbs) were counted and compared in their mean occurrence within sentences. It was found that children in 2nd grade have a personal way of writing (e.g., structures used are different from those of authors), while in 8th grade they are to some extent getting closer to the level of writing of the books. Increasing use of adjectives and adverbs over the years approach the profiles found in literature. Future work includes looking at correlations of features and adding information about word usage, spelling errors and semantics. A significant gap between leisure reading and children's texts with respect to their textbooks is observable. Further study needs to quantify that and determine a reasonable progression for didactics to advance students' towards academic skills.

References

Vilmos Ágel and Klaus Fischer. 2010. Dependency Grammar and Valency Theory. *Bernd Heine & Heiko Narrog (Hgg.), The Oxford Handbook of Linguistic Analysis, Oxford*, pages 223–255.

Richard Bamberger and Erich Vanecek. 1984. Lesen-Verstehen-Lernen-Schreiben: Die Schwierigkeitstufen von Texten in deutscher Sprache. *Wien: Jugend und Volk*.

Rebekah George Benjamin. 2012. Reconstructing readability: Recent developments and recommendations in the analysis of text difficulty. *Educational Psychology Review*, 24(1):63–88.

Kay Berkling and Uwe Reichel. 2014. Der phonologische Zugang zur Schrift im Deutschen.

Kay Berkling, Johanna Fay, Masood Ghayoomi, Katrin Heinz, Rémi Lavalley, Ludwig Linhuber, and Sebastian Stücker. 2014. A Database of Freely Written Texts of German School Students for the Purpose of Automatic Spelling Error Classification. In *LREC Conference*.

Wilfried Bos. 2004. IGLU: Einige Länger der BRD im nationalen und internationalen Vergleich.

William H. DuBay. 2008. The principles of readability. 2004. *Costa Mesa: Impact Information*, 76.

Ingo Glöckner, Sven Hartrumpf, Hermann Helbig, Johannes Leveling, and Rainer Osswald. 2006. An architecture for rating and controlling text readability. *Proceedings of KONVENS 2006*, pages 32–35.

Julia Hancke, Sowmya Vajjala, and Detmar Meurers. 2012. Readability Classification for German using Lexical, Syntactic, and Morphological Features. In *COLING*, pages 1063–1080.

Jessica Nelson, Charles Perfetti, David Liben, and Meredith Liben. 2012. Measures of text difficulty: Testing their predictive value for grade levels and student performance. *Council of Chief State School Officers, Washington, DC*.

Slav Petrov, Leon Barrett, Romain Thibaux, and Dan Klein. 2006. Learning accurate, compact, and interpretable tree annotation. In *Proceedings of the 21st International Conference on Computational Linguistics and the 44th annual meeting of the Association for Computational Linguistics*, pages 433–440.

Manfred Prenzel, Christine Sälzer, Eckhard Klieme, and Olaf Köller. 2013. PISA 2012: Fortschritte und Herausforderungen in Deutschland. *Münster: Waxmann*.

Sarah E. Schwarm and Mari Ostendorf. 2005. Reading level assessment using support vector machines and statistical language models. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 523–530.

Laurianne Sitbon and Patrice Bellot. 2008. A readability measure for an information retrieval process adapted to dyslexics. In *Second international workshop on Adaptive Information Retrieval (AIR 2008 in conjunction with IIiX 2008)*, pages 52–57.

Sowmya Vajjala and Detmar Meurers. 2014. Exploring Measures of "Readability" for Spoken Language: Analyzing linguistic features of subtitles to identify age-specific TV programs. In *Proceedings of the 3rd Workshop on Predicting and Improving Text Readability for Target Reader Populations (PITR)@ EACL*, pages 21–29.

GSCL

ÖGAI

dgfs
Deutsche Gesellschaft
für Sprachwissenschaft

UBIQUITOUS
KNOWLEDGE
PROCESSING

CLARIN-D

University of Hildesheim

Institute of Information Science and Natural Language Processing
Marienburger Platz 22
D-31141 Hildesheim

https://www.uni-hildesheim.de/iwist/

https://www.uni-hildesheim.de/konvens2014/