Published in: Calzolari, Nicoletta/Choukri, Khalid/Cieri, Christopher/Declerck, Thierry/Goggi, Sara/Hasida, Koiti/Isahara, Hitoshi/ Maegaard, Bente/Mariani, Joseph/Mazo, Hélène/Moreno, Asuncion/ Odijk, Jan/Piperidis, Stelios/Tokunaga, Takenobu (eds.): Proceedings of the eleventh international conference on language resources and evaluation (LREC 2018), 7-12 May 2018, Miyazaki, Japan. - Paris, France: European language resources association (ELRA), 2018. Pp. 2169-2174

GeCoTagger: Annotation of German Verb Complements with Conditional Random Fields

Monica Fürbacher, Roman Schneider
Institute for the German Language (IDS)
R5 6-16, 68161 Mannheim, Germany
{fuerbacher, schneider}@ids-mannheim.de

Abstract

Complement phrases are essential for constructing well-formed sentences in German. Identifying verb complements and categorizing complement classes is challenging even for linguists who are specialized in the field of verb valency. Against this background, we introduce an ML-based algorithm which is able to identify and classify complement phrases of any German verb in any written sentence context. We use a large training set consisting of example sentences from a valency dictionary, enriched with POS tagging, and the ML-based technique of Conditional Random Fields (CRF) to generate the classification models.

Keywords: Grammar and Syntax, Verb Valency, Machine Learning Methods

1. Introduction

Verb complements are indispensable for constructing a correct grammatical sentence in German. The appropriate usage of complements is one fundamental skill for language learners, so the concept of verb complements - or verb valency - is not only an established field of linguistic research, but also often used for didactical purposes. The popular valency dictionary VALBU (=*Valenzwörterbuch deutscher Verben*) (Schumacher et al., 2004) and its expanded online counterpart E-VALBU (=*Elektronisches Valenzwörterbuch deutscher Verben*) (Kubczak, 2009) support both linguists and language learners by providing detailed descriptions of nearly 700 German verbs with more than 3,000 reading variants. Besides other linguistically motivated information, the dictionaries contain authentical example sentences, extracted from DeReKo ((Kupietz et al., 2010)), with a manually added fine-grained markup of verb complement classes.

Unfortunately, compiled dictionaries are naturally limited and cannot cover all possible sentences of a living language and even not the range of all existing verbs. Filling this gap manually seems to be an unpromising task, because it consumes much time and is error-prone. An automatic classification of complements for each verb in any sentence would solve this sophisticated problem. Though we see a remarkable increase of machine learning (ML) tools for natural language processing, we do not know of any empirical approach for the automatic classification of verb complements.

For the development of our ML-based classification algorithm, we compile a corpus of 28,649 example sentences provided by the XML representation of E-VALBU (IDS-Mannheim, 2010) (Müller-Spitzer and Schneider, 2009). The corpus will then be POS-tagged and lemmatized. With this data set, we will train ML-models with different parameters, based on Conditional Random Fields (CRF). The result will not only indicate whether complements can be identified at all by our algorithm, but also prove whether the complements will be correctly classified.

2. Complement classification

The following section briefly addresses theoretical knowledge for the task of classifying verb complements. It covers some linguistic basics regarding complements and gives necessary background information for developing a classification algorithm based on machine learning.

2.1. Linguistic background

A grammatically correct German sentence consists of three main components: the verbal complex, at least one verb complement and the facultative supplements. Those are also called primary components (Zifonun et al., 1997). The verbal complex can consist of a main verb, an auxiliary verb or a modal verb. Each verb determines the number of obligatory complements which are also referred to as actants (Tesnière and Engel, 1980). This characteristic of the verb is called verb valency (Bussmann, 2008) (Schumacher, 1996); for a comparison of English and German see (Fischer, 1997).

Depending on the verb, none or up to four obligatory complements can be required. Thus, there are five different verb valency classes. Moreover, verbs require specific kinds of complements. For German, there exist eight complement classes: the subject, genitive, dative, accusative, adverbial, prepositional, predicative and verbal complement class. However, a sentence does not only contain obligatory complement phrases, but also facultative ones. The latter are often called supplements. While the obligatory complements are dependent on the verb, the supplement's occurrence is usually independent of the verb (Engel and Schumacher, 1978).

This possibility of a complement being facultative leads to some problems. For example, there are bivalent verbs which can occur in a monovalent way like the German verb *essen* (*to eat*) which is declared as bivalent, even if it can occur with only one actant, as demonstrated in the following example (1).

(1a) $[\text{Ich}]_{\text{Ksub}}$ esse.
I eat.

(1b) [Ich]$_{Ksub}$ esse [einen Apfel]$_{Kakk}$.
I eat an apple.

2.2. Machine learning

For solving the task of complement classification, the ML-based technique of Conditional Random Fields (CRF) is used. A CRF is an undirected graphical model which was introduced by Lafferty et al. (2001). It is defined as a linear-chain CRF with the random input variable *x* over observation sequences and the random output variable *y* over label sequences. Based on the fundamental theorem of the random fields, the applied formula for this joint distribution is given in (2).

In this, f_k and g_k are the binary feature functions while θ contains two parameters which are estimated from the training set and by using the improved iterative scaling algorithm. Those feature functions are defined by the means of the transitions between the observation sequences and the states or label sequences.

Furthermore, a graph *G* of the label sequence *Y* is defined as a linear chain whose cliques consist of edges *E=(i, i+1)* and nodes or vertices *V=(1,2, ... m)*. Whereas the edges focus on the transition of the observation sequence and the previous and current labels, the vertices creates the features for current label and the corresponding observation sequence.

(2) $$p_\theta(x|y) \\ exp(\sum_{e \in E,k} \lambda_k f_k(e, y|_{e'} x) \\ + \sum_{v \in V,k} \mu_k g_k(v, y|_{v'} x))$$

CRF is proven for many applications like POS-tagging (Lafferty et al., 2001) (Patel and Gali, 2008), named-entity-recognition (McCallum and Li, 2003) (Watrin et al., 2014), shallow parsing (Sha and Pereira, 2003) or sentence boundary detection (Liu et al., 2005), to name just a few.

3. Algorithm for complement classification

3.1. Data Set

For the dataset construction, we use example sentences from the E-VALBU corpus. E-VALBU is a freely available electronic valency dictionary with valency information for 677 German verbs. The verb selection is based on the vocabulary used for the certification of German as a foreign language by the Goethe Institute. Among other linguistically motivated data, E-VALBU contains information about obligatory and facultative complements by providing (mostly corpus-based) example sentences for each verb (see table 1). We extract these sentences and augment all words with part-of-speech (POS) annotations and lemmatizations. For this task, we choose the TreeTagger tool (Schmid, 2009) with the german-utf-8 tagset, which uses the Stuttgart-Tübingen-Tagset (STTS).

The dataset is then split up into a training file and a test file. The training file contains 80% of the data set, while the test file includes 20%. These values are pitted against the number of sentences (28,659) contained in the data set.

Complement Class	Example Sentences
Subject	**Ich** halte seinen Vorschlag für sehr vernünftig. **Die neue Bluse** steht dir gut.
Accusative	Sie liebt **diesen Mann**. **Ihre Hilfe** wird er annehmen müssen.
Dative	Ich konnte **seinen Worten** nicht immer folgen. Paul hat **mir** ein Buch geschenkt.
Genitive	Er versicherte den Präsidenten **seiner Freundschaft**. Der Zeuge hat sich **seiner** erinnert.
Adverbal	Das Inhaltsverzeichnis steht **am Anfang des Buchs**. Sie fährt **nach Heidelberg**. Die Sitzung beginnt **um drei Uhr**.
Prepositional	Ich denke **an dich**. Das liegt **an dir**. Er hält nichts **von diesem Vorschlag**.
Predicative	Das Wetter ist **schön**. Mein Vater ist **Arzt**. Man nannte ihn **einen Idioten**. Wir hielten ihn für **originell**.
Verbal	Er bedeutet ihr **zu kommen**. Das Medikament beginnt **zu wirken**. Die Untertanen finden, **dass die Steuern gesenkt werden müssen**.

Table 1: German example sentences from E-VALBU.

3.2. Training algorithm

For the model training with Conditional Random Fields, the open source software tool CRF++ is chosen, cf. (Kudo, 2005 2013). The tool's training algorithm is based on a limited-memory BFGS (LM-BFGS).

3.2.1. The feature template

As a first step, a so-called feature template file is created. The template file describes relations between the tokens and therefore determines the number of features. A template file does not only consist of one template but of many templates. This means that every line in a template file is a template by itself and consists of macros. A macro is specified like %x[i,j], whereas *i* is the row and *j* is the column. Table 2 illustrates, how to determine a macro for the sentence *Ich kaufe dir ein Buch.* (*I buy you a book.*) with its respective POS-tags and lemmata. In each row in the first column there is the word itself, the second column presents the POS-tag and in the third column there is the lemma of the original word. Thus, a first macro for a unigram template U1:%x[-2,1] is the POS-tag *PPER* while the second macro U2:%x[-1,1] is *VVFIN* and so on. The relation between these two neighboring tokens can be described in an own macro as U3:%x[-2,1]%x[-1,1]. Each of these macros represents one feature. Almost all tokens in that template file need to be specified as a macro. The only tokens which are not defined are the output tags.

3.2.2. Parameter settings

However, considering too many features overloads the model, so that CRF++ crashes without generating one. To avoid this problem, the parameter settings have to be modified, e.g. the cut-off threshold for the features, the hyperparameter, the shrinking size, the maximum number of

i \ j	0	1	2
-2	Ich	PPER	ich
-1	kaufe	VVFIN	kaufen
0	dir	PRF	du
1	ein	ART	einen
2	Buch	NN	Buch
3	.	$.	.

Table 2: Specifying macros.

iterations during a generating process, the parameter for the termination criterion or the number of used threads. For testing purposes, we generate sixteen disctinct models: (1) Parsedf2, (2) Parsedf2c2, (3) Parsedf3, (4) ParsProc, (5) ParsProcc2, (6) ParsProcc3, (7) ParsProcf2, (8) ParsProcf2c2, (9) Proc, (10) Procc2, (11) Procc3, (12) Procc4, (13) Procc5, (14) Procf2, (15) Procf2c2, (16) Procf3 (please refer to (Fürbacher, 2015) for a detailed description of the models).
For each model, different parameter settings are tested, as shown below. It has to be noted that the number of iterations will not be limited and therefore the parameter -m is not modified. Since the default settings for the shrinking size as well as for the termination criterion work best, these parameters are not changed, too. In addition, four threads are used for all generated models. The only parameters that are changed are the cut-off threshold for the features and the cost-value.
The 16th model utilizes the fewest number of features of all with only 881,352 features. Models number four to six use the most features: 10,771,240. Taking a look at the number of iterations, model number four requires only 250 iterations to be generated. The second, third and eighth models need to iterate most often with more than 400 times. Model number 16 is generated in the shortest time, more precisely within 288.08 seconds. The longest generation time is required for the third model with 826.23 seconds. Nevertheless, all models completed during 5 to15 minutes.

4. Results

The classification result of the models is a binary one, so that the F-measure can be used as a criteria for the model quality. The F-measure is calculated to clarify the following questions:

1. How well can the model distinguish whether a word is a complement or not?
2. How well can the model classify a complement into the correct complement class?

4.1. Results for complement identification

This F-measure also predicates how well the model can identify a word being no complement.
The model with the highest recall, precision as well as the highest F-measure is the first one which was trained with an unreviewed training file. With a recall of 81.3% and a precision of 92.7% it reaches an F-measure of 86.6%. Second best is the 10th model with 79.3%. It also reaches the second best F-measure with 82.4%, even if the precision is 85.7%. The third-best model is the 13th one which has a recall of 79%, a precision of 85.5% and an F-measure of 82.3%.
Furthermore, the cost-value affects the performance of the models. The model result gets worse by increasing the value up to 3.0. Interestingly, by raising the cost-value up to 4.0 and 5.0, the result gets better. However, not all models are optimized by the cost-value. For example, the result of the second model, for which the cost-value is raised up to 2.0, performs significantly worse than the first model with unmodified cost-value. With 77.9%, the recall of the second model is about 3.4 percentage points worse than the first one. Also, the precision reaches only 66.2%, which means it is about 26.5 percentage points lower than the first model. This leads naturally to an F-measure of 69.2%. This implies that the optimal cost-value depends on the training file and thus has to be ascertained separately for each one.

4.2. Results for complement classification

The results of the best models for the task of classifying the complements in their correct complement class are shown in table 3. We state a wide range of recall (.029 - .778) and precision (.1 - .798).
The F-measure shows how well the models can distinguish between certain complement classes. As a consequence, it also indicates which complement classes are difficult to identify.

4.2.1. Subject complement class

The overall results of the recalls for the task of classifying subject complements range from 59.5% to 77.8%, the precision of the models from 67.2% to 79.8%. This leads to an F-measure between 63.1% and 78.7%.
The 11th model has not only the best F-measure of 78.8%, but also the best recall of 77.8% and a precision of 79.8%. Model number nine is the second best with an F-measure of 78.7%, a recall of 77.8% and a precision of 79.6%. The third best one is the 13th model with an F-measure of 78.6%, a recall of 77.6% and a precision of 79.6%.

4.2.2. Accusative complement class

When classifying accusative complements, the recall results range between 45.0% and 64.2%. The lowest precision is 62.0% and the highest value is 66.8%. Thus, the yielded F-measure lies between 52.1% and 64.6%.
The best model is the 12th one. It has a recall of 63.7%, a precision of 65.5% and an F-measure of 64.6%. Second best is model number 10 with a recall of 64.2%, a precision of 64.4% and an F-measure of 64.3%. With a recall of 62.9%, a precision of 65.2% and an F-measure of 64.0%, the 11th model is the third-best one. They all have in common that they comprise all features, and also their cost-value is raised.

4.2.3. Dative complement class

The results of the recall reach from 22.2% to 59.3% for the task of classifying dative complements. The precision results range between 55.0% and 87.2%, and the F-measures vary between 31.6% and 62.7%.
The best model - number (6) - yields an F-measure of 62.7% with a recall of 59.3% and a precision of 66.7%.

For this one, all available features are used and only the cost-value was raised up to 3.0. For the second and third best models, only those features are involved which occur at least twice. The 14th model reaches the second best results with a recall of 41.6%, a precision of 85.1% and an F-measure of 55.9%. The third-best model for classifying dative complement is model number 15, which yields a recall of 43.3%, a precision of 86.5% and an F-measure of 57.7%. Besides the feature parameter, the cost-value is also modified for this model.

4.2.4. Genitive complement class

For the task of classifying genitive complements, the models one to eight fail to detect any genitive.

From the eight models that succesfully identify genitive complements, the recall ranges from 25.3% to 31.3%, while the value for the precision varies between 32.2% and 38.7%. The lowest F-measure is 29.8% and the highest one is 32.7%.

4.2.5. Adverbial complement class

The highest recall result for the task of classifying adverbial complements is 37.8%, while the lowest value is 24.2%. The precision ranges from 41.5% to 51.2%. This leads to an F-measure between 30.8% and 41.2%.

The best model overall is the 14th one, while models 13 and 15 are second and third best. Interestingly, not all features are used by creating the best model, and also the cost-value is not changed. Only while training the 13th model, the cost-value is raised up to 5.0. This implies that in order to create a well-working model for classifying adverbial complements, either not all features should be involved or the cost-value should increase considerably.

4.2.6. Prepositional complement class

The recall results for the classification of prepositional complements range from 33.9% to 63.5%, while the results for the precision range from 53.8% to 60.2%. The resulting F-measures vary between 42.7% and 60.9%.

The 6th model yields the highest F-measure with 60.9% by a recall of 61.6% and a precision of 60.2%. The 11th model shows a higher recall (63.5%) than the 6th one, but also a lower precision (58.3%). The 12th model has the same F-measure as the 11th one, but a recall of 63.1% and a precision of 58.7%. These three models are trained including all available features, but with an increased cost-value of 3.0 respectively 4.0.

4.2.7. Predicative complement class

For the task of classifying predicative complements, the lowest recall result is 32.3%, while the highest one is 39.6%. The worst result for the precision is 65.3% and the best is 81.9%. Hence, the F-measure varies between 38.0% and 51.7%.

With a recall of 38.8%, a precision of 75.5% and an F-measure of 51.7%, model number five is the best one in classifying predicative complements. Second best is model number six, which has a recall of 38.2%, a precision of 76.2% and an F-measure of 50.9%. The 4th model is the third-best with a recall of 36.5%, a precision of 81.9% and an F-measure of 50.5%. These models share the fact of using all available features. Moreover, the cost-value is raised up to 2.0 for the best model and up to 3.0 for the second best.

Complement Class	Recall	Precision	F-score
Subject	.778	.798	.788
Accusative	.637	.655	.646
Dative	.593	.667	.627
Genitive	.283	.387	.327
Adverbal	.341	.512	.409
Prepositional	.616	.602	.609
Predicative	.388	.755	.517
Verbal	.029	.100	.056

Table 3: Best results for each complement class.

4.2.8. Verbal complement class

Due to the fact that only six models are able to classify verbal complement phrases, this task seems to be one of the most difficult ones. Moreover, the six models that classified verbal complements at all yield bad results. The poorest recall is 1.4%, the best one only 2.9%. However, the precision for one model is 1.0% while the other five models have a precision of 100%. This leads to an F-measure between 4.1% and 5.6%.

The two best performing models include all available features and have an increased cost-value of 2.0 respectively 3.0. The third-best model is generated using only features which occur at least twice, but has also a raised cost-value of 2.0.

To sum up, it comes out that our complement classifying algorithms perform well in principle, but would very probably benefit from more training data for specific complement classes.

5. Discussion

All models yield reliable results for the task of classifying whether a word within a freely entered natural language sentence is a complement or not. However, when the models should distinguish between the given complement classes, the F-measure decreases. The corresponding results clearly suggest that the training set should contain more example sentences for the underrepresented verbal complements and genitive complements.

Since the manual tagging of these complement classes is both time consuming and indispensable, the further extension of our gold standard training corpus will probably be beneficial for the linguistic community. We are firmly convinced that this effort will result in even better recall/precision values for the automatic assignment of yet underrepresented complement classes.

6. Concluding Remarks

We presented a well-working ML-based algorithm for the identification of verb complements within any German sentence as well as for the annotating of complement classes. For that purpose, a specific data set was created, with POS-tagged and lemmatized example sentences for different complement types. Applying Conditional Random Fields

Figure 1: *GeCoTagger* Output

with the CRF++ toolkit, various models were generated with respect to feature-dependent parameter settings (see also (Fürbacher, 2015)).

We demonstrated that even relatively small amounts of natural language data – the valency dictionary used for our training runs contains about 700 verb lemmata with an example corpus of only some thousand annotated sentences – can constitute a sound basis for machine learning, given that they contain reliable, scientifically grounded, fine-grained information. Fur the future, we think of enhancing our application with a user feedback function, so that complement classifications that are manually evaluated as correct or wrong would contribute to further improvements of the classification model.

Furthermore, our mostly positive evaluation results led to the development of a web interface prototype, which is called *GeCoTagger* (=*German Complement Tagger*). It will be freely available for the linguistic community within the GRAMMIS information system (IDS-Mannheim, 2018) (Schneider and Schwinn, 2014) and allows users to enter natural language sentences, and to receive an analysis of its verb complements.

Figure 1 presents an online example for the classification output of the sentence *Drei Affen schenken dir eine Banane.* (*Three monkeys give you a banana.*), where each word of the sentence is coloured according to its complement class in E-VALBU. Since the user input is always pre-processed by Treetagger, POS-tags and lemmata can be added easily to the output.

7. Bibliographical References

Bussmann, H. (2008). *Lexikon der Sprachwissenschaft*. Kröner, Stuttgart.

Engel, U. and Schumacher, H. (1978). *Kleines Valenzlexikon deutscher Verben*. Number 31 in Forschungsbericht des Instituts für deutsche Sprache. Gunter Narr Verlag, Tübingen.

Fischer, K. (1997). *German-English Verb Valency: A Contrastive Analysis*. Tübinger Beiträge zur Linguistik. Narr.

Fürbacher, M. (2015). *Developing of an ML-Based Algorithm for Typing of Complement Phrases*. Master Thesis, University of Trier, Computational Linguistics.

IDS-Mannheim. (2010). *Das elektronische Valenzwörterbuch deutscher Verben*. http://www.ids-mannheim.de/e-valbu/.

IDS-Mannheim. (2018). *Grammis - Grammatisches Informationssystem*. https://grammis.ids-mannheim.de, DOI: 10.14618/grammis.

Kubczak, J. (2009). Hier wird Ihnen geholfen! Das elektronische Valenzwörterbuch deutscher Verben. In *Sprachreport 04/2009*, pages 17–23.

Kudo, T. (2005 – 2013). *CRF++: Yet Another CRF Tool Kit*. Version 0.58, https://taku910.github.io/crfpp.

Kupietz, M., Belica, C., Keibel, H., and Witt, A. (2010). The German reference corpus DeReKo: A primordial sample for linguistic research. In Nicoletta Calzolari (Conference Chair), et al., editors, *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta, may. European Language Resources Association (ELRA).

Lafferty, J. D., McCallum, A., and Pereira, F. C. (2001). Conditional random fields: Probabilistic models for segmenting and labeling sequence data. In *Proceedings of the Eighteenth International Conference on Machine Learning (ICML)*, pages 282–289.

Liu, Y., Stolcke, A., Shriberg, E., and Harper, M. (2005). Using conditional random fields for sentence boundary detection in speech. In *Proceedings of the 43rd Annual Meeting of the ACL*, pages 451–458.

McCallum, A. and Li, W. (2003). Early results for named entity recognition with conditional random fields, feature induction and web-enhanced lexicons. In *Proceedings of the Seventh Conference on Natural Language Learning (CoNLL)*, pages 188–191.

Müller-Spitzer, C. and Schneider, R. (2009). Ein XML-basiertes Datenbanksystem für digitale Wörterbücher - Ein Werkstattbericht aus dem Institut für Deutsche Sprache. *it-Information Technology*, 51(4):197 – 206.

Patel, C. and Gali, K. (2008). Part-of-speech tagging for gujarati using conditional random. In *Proceedings of the IJCNLP-08 Workshop on NLP for Less Privileged Languages*, pages 117–122.

Schmid, H. (2009). *TreeTagger - a part-of-speech tagger for many languages*. http://www.ims.uni-stuttgart.de/projekte/corplex/TreeTagger/.

Schneider, R. and Schwinn, H. (2014). Hypertext, Wissensnetz und Datenbank: Die Web-Informationssysteme grammis und Progr@mm. In Franz Josef Berens et al., editors, *Ansichten und Einsichten. 50 Jahre Institut für Deutsche Sprache*, pages 337–346. IDS, Mannheim. http://nbn-resolving.de/urn:nbn:de:bsz:mh39-24719.

Schumacher, H., Kubczak, J., Schmidt, R., and de Ruiter, V. (2004). *VALBU - Valenzwörterbuch deutscher Verben*. Number 31 in Studien zur Deutschen Sprache. Narr, Tübingen.

Schumacher, H. (1996). Satzbaupläne und Belegungsregeln im Valenzwörterbuch deutscher Verben. In *Lucien Tesnière - syntaxe structurale et opérations mentales: Akten des deutsch-französischen Kolloquiums anläßlich der 100. Wiederkehr seines Geburtstages, Strasbourg 1993*, pages 281–294. Niemeyer, Tübingen.

Sha, F. and Pereira, F. (2003). Shallow parsing with conditional random fields. In *Proceedings of HLT-NAACL 2003*, pages 134–141.

Tesnière, L. and Engel, U. (1980). *Grundzüge der strukturalen Syntax*. Klett-Cotta, Stuttgart.

Watrin, P., de Viron, L., Lebailly, D., Constant, M., and Weiser, S. (2014). Named entity recognition for German using conditional random fields and linguistic resources. In *Proceedings of the 12th KONVENS (Konferenz zur Verarbeitung natürlicher Sprache)*, pages 153–156.

Zifonun, G., Hoffmann, L., and Strecker, B. (1997). *Grammatik der deutschen Sprache: Bd. 1-3*. de Gruyter, Berlin.