Published in: Bański, Piotr; Biber, Hanno; Breiteneder, Evelyn; Kupietz, Marc; Lüngen, Harald; Witt, Andreas (eds.) (2015): Proceedings of the 3rd Workshop on Challenges in the Management of Large Corpora (CMLC-3). Mannheim: Institut für Deutsche Sprache, pp. 15-20.

Challenges in the Alignment, Management and Exploitation of Large and Richly Annotated Multi-Parallel Corpora

Johannes Graën
Institute of Computational Linguistics
University of Zurich
Zurich, Switzerland
graen@cl.uzh.ch

Simon Clematide
Institute of Computational Linguistics
University of Zurich
Zurich, Switzerland
siclemat@cl.uzh.ch

Abstract

The availability of large multi-parallel corpora offers an enormous wealth of material to contrastive corpus linguists, translators and language learners, if we can exploit the data properly. Necessary preparation steps include sentence and word alignment across multiple languages. Additionally, linguistic annotation such as part-of-speech tagging, lemmatisation, chunking, and dependency parsing facilitate precise querying of linguistic properties and can be used to extend word alignment to sub-sentential groups. Such highly interconnected data is stored in a relational database to allow for efficient retrieval and linguistic data mining, which may include the statistics-based selection of good example sentences. The varying information needs of contrastive linguists require a flexible linguistic query language for ad hoc searches. Such queries in the format of generalised treebank query languages will be automatically translated into *SQL* queries.

1 Introduction

The long-term goal of our project is the creation of a means for empirical linguistic research based on large amounts of multi-parallel texts, i.e. corresponding data for more than two languages.[1] Sample questions we seek to answer are: Which features trigger the use or absence of articles in English? How do other languages differ in their article use? What about languages which do not use the concept of articles?

Though we focus on linguists as end-users who use our system to find evidence to answer research questions, the option of relating several layers of linguistics metadata in the form of annotations and alignments may facilitate other use cases, such as dictionary look-ups for words in context in more than one corresponding target language[2], detecting triggers for translation variants of particular expressions and syntactical structures, and comparing corresponding patterns such as word order preferences across multiple languages.[3]

In this paper, we will discuss three prominent challenges to be addressed in our work. Section 2 deals with the characteristics of multi-parallel alignments and outlines techniques to attain them. Section 3 describes the data structures required for our research questions and how to map them to a database schema. Section 4 discusses the requirements for user-friendly reporting of query results and suggests an approach for an expressive linguistic query language.

2 Multi-parallel Corpus Data Preparation

At present, several large, multi-parallel corpora are freely available. *Europarl* (Koehn 2005) and *MultiUN* (Eisele and Chen 2010), for instance, comprise millions of tokens in 21 and 6 languages, respectively. Östling (2015, p. 6) illustrates some of the multi-parallel corpora available in terms of language count and average number of words per language. These corpora consist of parallel documents corresponding to each other.[4]

Pairwise sentence alignment for a number n of languages covered by the respective corpus re-

[1] The definition of ‘large’ in the context of corpora may well be a controversial one. We argue that counting entities, such as tokens, sentences, etc. does not suffice for measuring the largeness of a corpus, but that the richness of relations described by its data model is equally important.

[2] This particularly addresses language learners who are proficient in other languages.

[3] For a discussion of the needs of different user groups see Volk, Graën, and Callegaro (2014).

[4] More specific alignment is implicitly given by speaker turns in the case of *Europarl* (see Graën, Batinic, and Volk 2014, p. 224).

sults in $\binom{n}{2}$ pairs sets of pairwise alignments since the correspondences of sentences are expressed as bidirectional alignments.

2.1 Multi-parallel Alignments

To address questions that involve more than two languages, pairwise sentence alignments pose a problem since combining several sets of pairwise alignments (again $\binom{n}{2}$ pairs for n languages) yields rather big graphs of sentences, moreover, alignment errors tend to propagate. This is depicted in Fig. 1 for 3 languages and 3 sets of pairwise alignments.

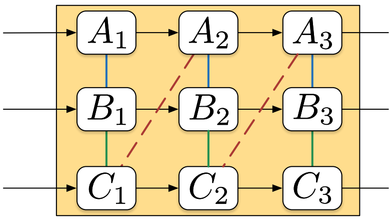

Figure 1: The alignment errors between language A and C (dashed lines) result in overly connected alignment graphs. The yellow box is the closure of all pairwise alignments.

Rather than closures of pairwise alignments, we require sets of corresponding sentences in all languages, denoting that all contained sentences mutually correspond to each other. We call such a set a **multi-parallel alignment** (MPA). MPAs may contain other MPAs, as depicted in Fig. 2, as long as these build a proper subset of the containing MPA.

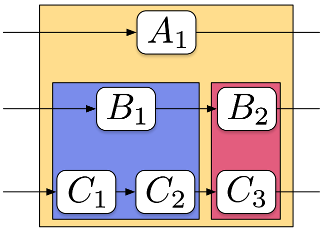

Figure 2: MPAs (coloured boxes) designate the elements of each language they extend over as corresponding.

The same problem applies to word alignment, once a multi-parallel sentence alignment has been found, where correspondence is usually calculated unidirectionally[5], which results in a set of 2 × $\binom{n}{2}$ unidirectional pairwise alignments. Several well-known algorithms exist to deduce a bidirectional word alignment from a pair of unidirectional ones[6], but they may result in a loss of valuable information for linguistic questions (Lehner, Graën, and Clematide 2015).

Analogous to the MPAs of sentences, different granularities of word correspondences can be expressed by nested MPAs as shown in Fig. 3.

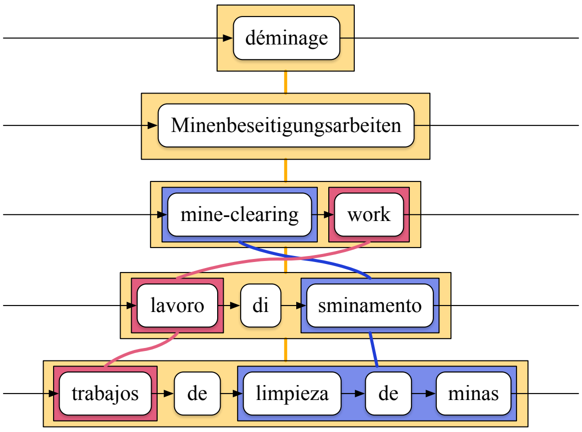

Figure 3: MPAs on a sub-sentential level, ranging from word to phrase alignment. Two MPAs with elements in three languages (red and blue) are contained by a broader MPA (yellow) which covers five languages.

2.2 Approaches for Attaining Multi-parallel Alignments

In order to obtain MPAs on a sentence level, we calculated the respective pairwise alignments for a total of five languages with *hunalign* (Varga, Halácsy, Kornai, Nagy, Németh, and Trón 2005) and combined the respective alignments in a graph as shown in Fig. 1. We then removed improbable links, i.e. those receiving less support from the other language pairs, by applying different heuristics which performed well for highly parallel texts. As soon as the translations became loose, our algorithms were unable to make good decisions.

In our opinion, this problem arises because, after the pairwise alignment step, alternative alignment scores get lost, and only the solution maximising the overall alignment score of the particular pair of texts is returned. A multi-parallel alignment performed on a joint alignment score is supposed to yield better, and a priori consistent, results. As the

[5] That means that a token t_a of language A being aligned with a token t_b of language B does not imply a reverse alignment between t_b and t_a.

[6] This process is called symmetrisation (Liang, Taskar, and Klein 2006; Tiedemann 2011, pp. 75–77).

costs of calculating scores for all possible alignment options grows quadratically, both time-wise and memory-wise, with the number of languages involved, an exhaustive search is not feasible. Instead, we are working on an approximate dynamic programming approach (Powell 2007).

To compute the MPAs on a word level, we plan to implement an algorithm similar to the one used for multilingual sentence alignment. We expect the complexity of this task to be considerably higher mainly since **(a)** sentences comprise more words than a textual unit contains sentences[7], **(b)** the constraint of sequentiality does not hold for words between multi-parallel aligned sentences, and **(c)** based on our previous investigations, we expect the word alignment ratio to vary strongly across languages.[8]

Bilingual alignment algorithms for phrase-structure parses have been reported by Zhechev and Way (2008). We plan to adapt their approach to our multi-lingual dependency parses.

3 Efficient Representation of Multi-parallel Corpora in an RDBMS

We expect our corpus compilation to be an aid in answering complex cross-linguistic questions by means of correlating different linguistically motivated data layers on a large scale of data. We identified the following eligible layers: sentence segmentation, tokenisation, lemmatisation, part-of-speech tagging, chunking, syntactical dependency parsing, coreference resolution (based on parse trees), sentence alignment, word alignment and sub-sentential alignment.

There are several NLP tools available for each layer. We allow for multiple annotation and alignment layers of the same kind, e.g. dependency parses by different parsers, with the exception of sentence and token segmentation where we commit to a single layer of primary data.[9] Apart from the primary data, each of these layers is based on at least one other layer such that the layer dependencies form a directed acyclic graph. In this vein, we know which dependent layers to recreate once a particular layer is rebuilt. In contrast to Bański, Fischer, Frick, Ketzan, Kupietz, Schnober, Schonefeld, and Witt (2012, p. 2906), we do not require query results to be reproducible after such layer rebuilds.

3.1 Data Types Required for the Representation of Linguistic Data Layers

In our considerations of the data structure required for building a conceptual data model incorporating those respective layers (and potential future ones), we identified three abstract data types which can be composed in such a way that all our requirements are met:

1. an interval on sequential elements,
2. a directed binary relation between two elements of the same type and
3. an undirected relation between several elements of the same type.

Each of these types, as well as a basic one without further definitions, may comprise any number of attributes such as labels, confidence scores, etc.

Tokens are basic elements and have attributes like their surface form, lemmas, and part-of-speech tags. Chunks are represented as intervals on tokens, dependency relations and unidirectional word alignments as relations between two tokens. Finally, the most complex type, n-ary relations between sets of elements, is needed for modelling MPAs[10], as well as for the modelling of coreference chains for instance.

3.2 Deriving a Database Schema from the Data Model

Corpus query systems are optimised for efficient retrieval rather than for processing new data, as the underlying linguistic data typically does not change. Richly annotated and aligned corpora allow for considerably more sophisticated corpus queries and thus require an efficient way to retrieve data in a less restricted fashion.

In times of freely available, advanced relational database management systems (RDBMS) which target flexible and efficient retrieval of large amounts of arbitrary structured data, building an own storage and retrieval system from scratch seems pointless (Davies 2005).

The limitation to the three described abstract data types allows us to define a translation pattern for the conversion of the data model into a

[7] In *Europarl*, a sentence contains three times more words on average than a textual unit contains sentences.

[8] As Fig. 3 illustrates, a ratio of 1:5 is not uncommon for aligned complex noun phrases, whereas ratios of 1:3 or more in sentence alignments are rare ($< 1\%$).

[9] Chiarcos, Ritz, and Stede (2009) discuss problems that arise with multiple tokenisation layers.

[10] In Fig. 3, these relations are expressed by connecting lines between sets of words in each language.

relational database schema, including normalisation, indices and access functions as stored procedures. Moreover, snippets for the retrieval of the particular data types can also be compiled uniformly based on the data model. As a further advantage, our RDBMS, PostgreSQL[11], includes an advanced query optimiser whose goal is defined to determine the most efficient query plan by rewriting a given query (see Momjian 2015).

4 User-friendly Reporting and Flexible Querying

Our third challenge involves two aspects:

1. How can we flexibly report user-friendly query results?
2. What is needed to enable contrastive corpus linguists, who are generally non-experts in *SQL*, to formulate their information needs more naturally in an expressive linguistic query language?[12]

4.1 User-friendly Reporting of Query Results

For the use case of cross-lingual frequency distributions of translations illustrated by example sentences, a simple form-based query menu is probably adequate. The user input, for instance, word or base forms including part-of-speech filters, can be easily interpolated into handcrafted *SQL* templates.

Applying such queries to large corpora is likely to yield large amounts of search hits. A practical challenge for the usability of such a system lies in the proper selection of sentences that are delivered to the end user as relevant and informative examples. This is an instance of the *Good Dictionary Example Extractor* problem (Kilgarriff, Husák, McAdam, Rundell, and Rychlỳ 2008), termed *GDEX* in the context of the *Sketch Engine* (Kilgarriff, Baisa, Bušta, Jakubíček, Kovář, Michelfeit, Rychlỳ, and Suchomel 2014).

Kosem, Husak, and McCarthy (2011) discuss many textual features (sentence lengths, punctuation, frequency thresholds on words, anaphoric expressions, etc.) that must be statistically evaluated for such a task. Our RDBMS includes the option to use *R* as an embedded statistical programming language, which we expect to be sufficient for our needs.

Furthermore, statistical evaluation of result sets, for instance, across different language pairs, could be provided given the ability to statistically analyse query results.

4.2 An Expressive Linguistic Query Language for Our Data Model

SQL allows the user to flexibly query every aspect of our data model, that is, every entity, attribute, relation and Boolean combination thereof. However, native *SQL* queries for our highly interconnected and normalised data structures are not an appropriate abstraction level for linguists; they cannot express their linguistic information needs in a natural way.

Therefore, there is a need for an expressive linguistic query language to flexibly describe the constructions contrastive linguists are interested in. Two important strains of linguistic query systems have been developed in the past:

1. Corpus linguistics tools for text corpora such as *CQP* (Christ 1994) and
2. treebank query tools such as *TIGERSearch* (König, Lezius, and Voormann 2003).

CQP supports annotated words, structural boundaries (sentences, constituents), and sentence-aligned parallel texts right from the beginning. For instance, a query for the word *car* in the English part of *Europarl* may be restricted to the co-occurrence of the German word *Auto* in the aligned sentence using the `within` operator:

```
[word="car"] within europarl7_de: [word="Auto"]
```

Although useful, this is not the level of expressiveness we have in mind.

In recent years, treebank query systems have been generalised in various ways. The *Stockholm Treealigner* (Lundborg, Marek, Mettler, and Volk 2007) introduced an operator for querying alignments between words or phrases of bilingual treebanks, freely combinable with precise monolingual *TIGERSearch*-like queries for syntactic structures. The *ANNIS* platform (Zeldes, Lüdeling, Ritz, and Chiarcos 2009) with its query language *AQL* for multi-level graph-based annotations offers operators for dependency relations, inclusion or overlap of token spans, and namespaces for annotations of the same type produced by different tools (for instance, the output of different dependency parsers, see also section 3). Our proposed

[11] `http://www.postgresql.org/`

[12] These linguists typically have varying research questions and a strong need for flexible and precise queries.

linguistic query language will include these operators and follow the logic-based style of this language family.[13] The next step in our work is therefore a translation of *Treealigner*/*AQL*-style queries into native *SQL* queries for our database. Rosenfeld (2010) describes the translation of *AQL* into *SQL*, which in turn is inspired by the implementation of the *DDDQuery* language (Faulstich, Leser, and Vitt 2006), an extended XPath query language for linguistic data.

Lai and Bird (2010) discuss the formal expressiveness of linguistic query languages and mention the known inherent limitation of *AQL*-style query languages to the fragment of existential first-order logic, which does not support queries for missing constituents. Recently, we proposed an approach where the result sets of several *AQL*-style queries are subtracted in order to identify configurations with missing constituents (Clematide 2015).

5 Conclusions

We identified three of the most prominent issues that we face building a system for querying large multi-parallel corpora with several inter-connected layers of linguistic information.

Typically, alignments have been calculated pairwise. Multi-parallel alignments, as we call the mutual correspondence relation between sets of elements of multiple languages, demand new, innovative approaches. Once the annotation and alignment data has been obtained, we need to store this complex accumulation in a fashion that supports efficient retrieval from multiple layers. Hence, we argue for the use of a relational database. We built a data model upon three abstract data types which incorporates the data structures of the aforementioned layers and allows for a direct translation into a database schema.

Having set up a database comprising multi-parallel corpus data with several layers of annotation and alignment, our intended end user requires a means to access said information in a convenient way. We sketched a flexible, yet user-friendly query language to deal with any kind of data layers defined within the data model whose queries can be mapped to *SQL* queries and thereupon processed by the database. On this basis, we discussed varying requirements regarding the presentation of query results (reporting), ranging from a selection of prototypical exemplars to an automatic statistical evaluation.

Acknowledgment

Thanks to Martin Volk for discussions and Rahel Oppliger for proof reading. 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

Bański, Piotr, Peter M Fischer, Elena Frick, Erik Ketzan, Marc Kupietz, Carsten Schnober, Oliver Schonefeld, and Andreas Witt (2012). “The new IDS corpus analysis platform: Challenges and prospects”. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC)*, pp. 2905–2911.

Chiarcos, Christian, Julia Ritz, and Manfred Stede (2009). “By all these lovely tokens... Merging Conflicting Tokenizations”. In: *Proceedings of the Third Linguistic Annotation Workshop*, pp. 35–43.

Christ, Oliver (1994). “A modular and flexible architecture for an integrated corpus query system”. In: *Proceedings of COMPLEX’94: 3rd Conference on Computational Lexicography and Text Research*. (Budapest), pp. 23–32.

Clematide, Simon (2015). “Reflections and a Proposal for a Query and Reporting Language for Richly Annotated Multiparallel Corpora”. In: *Proceedings of the Workshop on Innovative Corpus Query and Visualization Tools*. (Vilnius). Nordic Conference of Computational Linguistics (NODALIDA), pp. 6–16.

Davies, Mark (2005). “The advantage of using relational databases for large corpora: Speed, advanced queries and unlimited annotation”. In: *International Journal of Corpus Linguistics* 10.3, pp. 307–334.

Eisele, Andreas and Yu Chen (2010). “MultiUN: A Multilingual Corpus from United Nation Documents.” In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*. (Valletta). European Language Resources Association (ELRA), pp. 2868–2872.

[13] We are aware of alternatives, for instance, *XPath*-style query languages such as LPath (Lai and Bird 2010) or PML-TQ (Štěpánek and Pajas 2010).

Faulstich, Lukas C., Ulf Leser, and Thorsten Vitt (2006). “Implementing a linguistic query language for historic texts”. In: *Current Trends in Database Technology – EDBT 2006*. Springer, pp. 601–612.

Graën, Johannes, Dolores Batinic, and Martin Volk (2014). “Cleaning the Europarl Corpus for Linguistic Applications”. In: *Proceedings of the 12th KONVENS*. (Hildesheim), pp. 222–227.

Kilgarriff, Adam, Vít Baisa, Jan Bušta, Miloš Jakubíček, Vojtěch Kovář, Jan Michelfeit, Pavel Rychlỳ, and Vít Suchomel (2014). “The Sketch Engine: ten years on”. In: *Lexicography* 1.1, pp. 7–36.

Kilgarriff, Adam, Milos Husák, Katy McAdam, Michael Rundell, and Pavel Rychlỳ (2008). “GDEX: Automatically finding good dictionary examples in a corpus”. In: *Proceedings of the 13th EURALEX International Congress*. (Barcelona).

Koehn, Philipp (2005). “Europarl: A parallel corpus for statistical machine translation”. In: *Machine Translation Summit*. (Phuket). Asia-Pacific Association for Machine Translation (AAMT), pp. 79–86.

König, Esther, Wolfgang Lezius, and Holger Voormann (2003). *TIGERSearch 2.1 – User’s Manual*. Institute for Natural Language Processing, University of Stuttgart.

Kosem, Iztok, Milos Husak, and Diana McCarthy (2011). “GDEX for Slovene”. In: *Electronic lexicography in the 21st century: New Applications for New Users: Proceedings of eLex 2011*. (Bled), pp. 151–159.

Lai, Catherine and Steven Bird (2010). “Querying linguistic trees”. In: *Journal of Logic, Language and Information* 19.1, pp. 53–73.

Lehner, Stéphanie, Johannes Graën, and Simon Clematide (2015). *Compound Alignment Gold Standard (COMPAL GS)*. URL: `http://pub.cl.uzh.ch/purl/compal_gs` (visited on May 29, 2015).

Liang, Percy, Ben Taskar, and Dan Klein (2006). “Alignment by agreement”. In: *Proceedings of the Main Conference on Human Language Technology Conference (HLT-NAACL)*. (New York). Association for Computational Linguistics (ACL), pp. 104–111.

Lundborg, Joakim, Torsten Marek, Maël Mettler, and Martin Volk (2007). “Using the Stockholm TreeAligner”. In: *Proceedings of the 6th Workshop on Treebanks and Linguistic Theories*. (Bergen), pp. 73–78.

Momjian, Bruce (2015). *Explaining the Postgres Query Optimizer*. URL: `http://momjian.us/main/writings/pgsql/optimizer.pdf` (visited on May 29, 2015).

Östling, Robert (2015). *Bayesian Models for Multilingual Word Alignment*. Department of Linguistics, Stockholm University.

Powell, Warren B (2007). *Approximate Dynamic Programming: Solving the curses of dimensionality*. Vol. 703. John Wiley & Sons.

Rosenfeld, Viktor (2010). *An implementation of the Annis 2 query language*. Tech. rep.

Štěpánek, Jan and Petr Pajas (2010). “Querying Diverse Treebanks in a Uniform Way”. In: *Proceedings of the 7th International Conference on Language Resources and Evaluation (LREC)*.

Tiedemann, Jörg (2011). “Bitext Alignment”. In: *Synthesis Lectures on Human Language Technologies* 4.2, pp. 1–165.

Varga, Dániel, Péter Halácsy, András Kornai, Viktor Nagy, László Németh, and Viktor Trón (2005). “Parallel corpora for medium density languages”. In: *Proceedings of the Recent Advances in Natural Language Processing (RANLP)*. (Borovets), pp. 590–596.

Volk, Martin, Johannes Graën, and Elena Callegaro (2014). “Innovations in Parallel Corpus Search Tools”. In: *Proceedings of the 9th International Conference on Language Resources and Evaluation (LREC)*. (Reykjavik), pp. 3172–3178.

Zeldes, Amir, Anke Lüdeling, Julia Ritz, and Christian Chiarcos (2009). “ANNIS: A search tool for multi-layer annotated corpora”. In: *Proceedings of Corpus Linguistics*. (Liverpool).

Zhechev, Ventsislav and Andy Way (2008). “Automatic generation of parallel treebanks”. In: *Proceedings of the 22nd International Conference on Computational Linguistics (Coling)*. Vol. 1, pp. 1105–1112.