Alignment tools for parallel treebanks

Samuelsson, Y ; Volk, Martin

Alignment Tools for Parallel Treebanks

Yvonne Samuelsson and Martin Volk
Department of Linguistics, Stockholm University,
106 91 Stockholm, Sweden
{yvonne.samuelsson, volk}@ling.su.se

Abstract

This paper reports about our efforts in creating a tri-lingual parallel treebank. The focal points are consistency checking and all aspects of sub-sentential alignment. We discuss the alignment guidelines, the importance of quality checks, and special alignment problems. Then we look at alignment algorithms and alignment visualization tools and we compare our own TreeAligner with other alignment tools. Our constituent structure treebanks contain just over 1,000 sentences and around 18,000 tokens in each language.

1 Introduction

The combined research on treebanks and parallel corpora has recently led to parallel treebanks. A parallel treebank consists of syntactically annotated sentences in two or more languages, taken from translated (i.e. parallel) documents. In addition, the syntax trees of two corresponding sentences are aligned on a sub-sentential level (word, phrase and clause level), which we refer to as phrase alignment. Parallel treebanks can serve as training or evaluation material for an example-based machine translation system, as a corpus for the automatic derivation of transfer rules, for the extraction of bilingual dictionaries and for translation studies.

We are developing a German-English-Swedish parallel treebank, SMULTRON (Stockholm MULtilingual TReebank). It contains texts from two domains (a novel and economy texts) and consists of just over 1,000 sentences and around 18,000 tokens in each language. This paper is a report on experiences regarding the alignment. We will look at the alignment guidelines and some problems with semantic equivalence, as well as alignment algorithms and a number of alignment tools that we will compare to our own tool, the Stockholm TreeAligner.

2 Building the parallel treebanks

2.1 Creating the monolingual treebanks

In creating the parallel treebank, we first annotated the monolingual treebanks with the ANNOTATE treebank editor[1]. It includes a statistical part-of-speech tagger and a chunker. We annotated the English treebank according to the Penn Treebank grammar (Bies et al., 1995) while the German follows the TIGER annotation schema (Skut et al., 1997, Brants et al., 2002). For the Swedish treebank we used an adapted version of the German TIGER guidelines. Both the part-of-speech tags and the syntactic structure were manually checked. In addition, we automatically checked for completeness and consistency. Since the TIGER annotation guidelines lead to flat trees, the German and Swedish trees were automatically deepened by inserting unambiguous nodes (see Samuelsson and Volk, 2004).

2.2 Completeness and consistency checking

Completeness and consistency are important characteristics of corpus annotation. Tree completeness means that each token[2] and each node is part of the tree. This can easily be checked and should ideally be part of the annotation tool.[3]

Consistency checking over treebanks is a lot more complicated. Consistent annotation means that the same token sequence (or part-of-speech sequence or node sequence) is annotated in the same way across the treebank. Dickinson (2006), Dickinson and Meurers (2005) have proposed various methods to check treebanks for consistency violations. Their methods boil down to re-engineering the grammar underlying the treebank, and to checking whether the annotation adheres to the grammar.

We have used one of their methods for counting how often a sequence has a certain mother node. For example, if the sequence of Determiner - Noun is mostly annotated as a noun phrase, but only a few times as something else, then these few occurrences are possible annotation errors. The method basically extracts all phrase structure rules from the treebank and sorts and counts them by their right-hand sides. This reduces the manual labour and is very useful.

Since Dickinson (2006) worked mostly with the Penn Treebank, he did not pay attention to functional labels (which are only sparsely used in the Penn Treebank). In our German and Swedish treebanks, however, each edge has a functional label. We have therefore developed our own method for consistency checking of functional triples consisting of *function label - mother node - daughter node*. For example, we count all different function labels of noun phrases that are daughter nodes to a sentence. According to the TIGER guidelines such noun phrases can only be of certain functions. If we find other function labels, then this indicates a possible annotation error.

[1] `www.coli.uni-sb.de/sfb378/negra-corpus/annotate.html`

[2] Different treebanks may take different positions on whether special tokens like punctuation symbols should be part of the tree. For example, the Penn Treebank guidelines require punctuation marks to be part of the tree whereas the German TIGER guidelines leave them unattached.

[3] Problematic cases for completeness are grammatical errors due to superfluous words. For example, one sentence in our corpus contained a separated verb prefix twice. We decided that even such 'superfluous' tokens need to be included in the tree, albeit with a special function label.

Mother category	Function label	Daughter category	Frequency
VP	accusative-object	cat="CNP"	6
VP	accusative-object	cat="NP"	96
VP	accusative-object	cat="PN"	1
VP	accusative-object	pos="ART"	1

Table 1: Consistency checking of functional triples.

We found it to be practical to extract all function triples to an Excel table so that we can sort them by frequency, function label, mother node, or daughter node. Table 1, an excerpt from our German treebank, shows that we have annotated 96 noun phrases as accusative objects in a verb phrase and additionally 6 co-ordinated noun phrases (CNP). We have also annotated one proper noun (PN) and one determiner (ART) as an accusative object. The latter turned out to be an annotation error. A relative pronoun was mistakenly tagged as a determiner (they are homographs in German). By changing the part-of-speech tag to relative pronoun and inserting the appropriate noun phrase node, the error was corrected.

So far we have checked these lists manually. For example, our German economy treebank (around 11,000 tokens in 510 trees) with close to 7100 nodes resulted in a function table with 277 different function triples. In the future we will check the functional constraints automatically, rather than manually going through this list. This can be done by formulating the constraints over functional triples explicitly.

3 Alignment

After creating the monolingual treebanks, we convert the trees into TIGER-XML, a powerful database-oriented representation for graph structures[4]. In a TIGER-XML graph each leaf (= token) and each node (= linguistic constituent) has a unique identifier. We use these unique identifiers for the phrase and word alignment across trees in corresponding translation units. We also use an XML representation for storing this alignment.

3.1 Alignment guidelines

Phrase alignment can be regarded as an additional layer of information on top of the syntax structure. It shows which part of a sentence in one language is equivalent to which part of a corresponding sentence in another language. We draw alignment lines manually between sentences, phrases and words over parallel trees. This is done with the help of our alignment tool, the Stockholm TreeAligner, a graphical user interface to insert (or correct) alignments between pairs of syntax trees. Figure 1 shows a screenshot with two aligned trees. We want to align as many phrases as possible. The goal is to show translation equivalence. Phrases shall be aligned only if the tokens that they span represent the same meaning, if they could serve as translation units outside the current

[4] `http://www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/doc/html/TigerXML.html`.

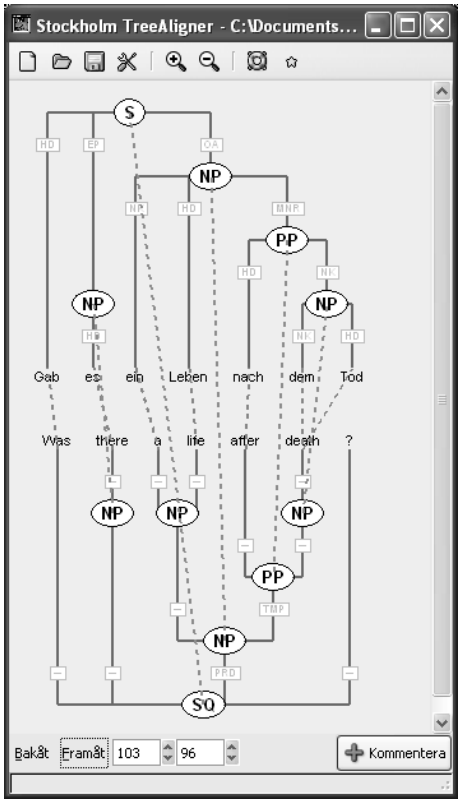

Figure 1: A screenshot of the TreeAligner, with one German and one English aligned tree.

sentence context. The grammatical forms of the phrases need not fit in other contexts, but the meaning has to fit.

We differentiate between two types of alignment, displayed by different colours in our alignment tool. Nodes and words representing exactly the same meaning are aligned as exact translation correspondences, like the English noun phrase *the human brain* and the Swedish *den mänskliga hjärnan*. If they represent approximately the same meaning, they are aligned as fuzzy translation correspondences, like the English prepositional phrase *on her way home from school* and the German *auf dem Heimweg von der Schule*, because of the pronoun *her*.

Our alignment guidelines allow phrase alignments within m:n sentence alignments and 1:n phrase alignments. Even though m:n phrase alignments are technically possible, we have only used 1:n phrase alignments (not specifying the direction), for simplicity and clarity reasons. One example of 1:n alignment on the word level is the English multi-word expression *the fruit trees* which is one word in Swedish, *fruktträden*. The 1:n alignment option is not used if a node from one tree is realized twice in the corresponding tree, e.g. a repeated subject in coordinated sentences.

Pronouns should not be aligned to full noun phrases. Nodes that contain required extra information in one language should not be aligned. This means that e.g. a sentence (with the subject) cannot be aligned to a verb phrase (without the sentence subject).

In analogy to quality checks over treebanks, we would like to ensure that the guidelines

are followed and that the alignments are complete and consistent. We check for all aligned single tokens and token sequences whether they are aligned in the same way (i.e. with the predicate 'exact' or 'fuzzy') to the same corresponding tokens. We also check whether the aligned token sequences differ in length and we examine those cases where different types of nodes are aligned across the languages (see Samuelsson and Volk, 2006).

3.2 Problematic alignment

Translation equivalence is a problematic subject. (Some problems due to different annotation schemata were already discussed by Samuelsson and Volk (2006).) Let us look at prepositions, which turned out to be problematic for the word alignment since they vary from being semantically well defined to semantically vague. We had decided that typical correspondences between prepositions are to be marked as exact while others are approximate. The prepositions in English *in the mailbox* and German *in dem Briefkasten* are clearly a case of exact equivalence. But what of the English phrase *a bowl of cat food* and German *eine Schale mit Katzenfutter* (literally a bowl with cat food)? Here it would be easier to look at the NP. If they are exact correspondences, then the prepositions are too.

This still leaves us with a problem, since some prepositions and articles in German can be contracted, e.g. *an dem* becomes *am*, *in das* becomes *ins*. If the contraction equals a preposition and article in the other language, these should be aligned as exact equivalence, e.g. *at the end* and *am Ende*. However, when aligning with Swedish, where the definite article is a suffix, this is problematic. For German *beim Supermarkt* and Swedish *vid snabbköpet* (at the supermarket) we have decided to draw fuzzy alignment between the prepositions (while German has exact 1:2 alignment to English and only the prepositions are aligned for Swedish-English).

3.3 Alignment transfer

We have been looking at alignment transfer, where the alignment between L2 and L3 is derived from the alignment between L1 and L2 and between L1 and L3. The simplest cases are one-to-one mappings. If e.g. the English prepositional phrase *of daffodils* is aligned to the Swedish PP *av påskliljor* and to the German PP *von Osterglocken*, we can automatically create alignment between the Swedish and the German PP. This simple mapping applies to all cases where there is a uniform mapping on both sides.

There are however a few cases where we cannot create the automatic alignment. One example is two English words, *(in) the garden*, that are aligned to one Swedish word, *(i) trädgården*, while only one of the English words, *garden*, is aligned to one German word, *(im) Garten*. This gives a partial mapping between German and Swedish and should be left out. Another problematic case is when one English phrase is aligned to two German and two Swedish phrases. We do not want to create many-to-many alignment and we cannot know which phrase in German belongs to which Swedish phrase.

4 Alignment tools and algorithms

4.1 Sentence alignment

There are well-established algorithms for aligning sentences (which represent translation correspondences) across parallel corpora. These algorithms are based on features such as sentence length (in terms of number of characters), word correspondences (as taken from bilingual dictionaries, or automatically found cognates), or distance factors. A well-known example is the algorithm presented by Gale and Church (1993) which uses length comparisons. It allows 1:m sentence alignments and optimizes the sentence mapping across paragraphs. Such sentence alignment algorithms have found their way into commercial translation tools as a means to fill translation memories with previously translated texts (Groß, 1998).

It should be remembered, however, that these algorithms work only as long as the translation units occur in the same order in the parallel texts. They fail to capture order variations that are more than direct pairwise permutations which can be matched as one block. In our texts we found an itemized listing consisting of full sentences where one translator had decided to render the items in a different order. Such variations cannot be handled correctly with the standard sentence alignment algorithms.

4.2 Word alignment algorithms

Word and phrase alignment goes beyond sentence alignment in that it captures subsentential correspondences: Which part of a sentence in one language corresponds to which part of a translated sentence in a parallel text? Word alignment algorithms are usually based on prior sentence-aligned corpora. However, the features for computing word alignment cannot be the same as for sentence alignment. The word order is different across languages and length comparisons may help but are not indicative. Instead co-occurrence statistics can be used. If two words frequently co-occur in corresponding sentences, they are good candidates for translation correspondences.

Considering the different compounding dynamics in languages like English versus German (or Swedish) it is obvious that 1:m word correspondences must be applied. Often a German compound noun corresponds to a complex noun phrase in English (e.g. in our corpus *Zertifizierungsverfahren* corresponds to *process of certification*).

Sometimes it is not even possible to establish correspondences on the word level; there are rather meaning equivalences on larger units. We capture this by using phrase alignments. For example, the co-ordinated phrase *die Papierindustrie und der Bausektor* certainly corresponds as a whole to *the paper and construction sector*, but we would not want to align *Papierindustrie* to *paper* alone.

Following Tiedemann (2003) we may distinguish two types of word alignment approaches. He calls them association approaches and estimation approaches (they are also called heuristic models and statistical models by e.g. Och and Ney (2003)). Association approaches use string similarity measures, word order heuristics, or co-occurrence measures (e.g. mutual information scores). For the latter, the idea is to find out if a cross-language word pair co-occurs more often than could be expected from chance.

Estimation approaches use probabilities estimated from parallel corpora, inspired from

statistical machine translation, where the computation of word alignments is part of the computation of the translation model. The word correspondences computed by the freely available GIZA++ system (Och and Ney, 2000) have constantly scored high in evaluations. All these methods include multi-word units as alignment targets, these units sometimes being pre-computed and sometimes being determined during the alignment process.

4.3 Word and phrase alignment tools

There are a number of tools that visualize word/phrase alignment. We have looked at several of them (even though the list is not exhaustive). We will not discuss tools that are only used for viewing alignment (like e.g. the Cairo tool (Smith and Jahr, 2000) or our own Stockholm Alignment Viewer (Samuelsson and Volk, 2005)). Instead we focus on tools that allow manipulation of the alignments, either automatically or manually by the user. We have looked at the following tools for word alignment.

- Alpaco (`http://www.d.umn.edu/~tpederse/parallel.html`), created by Brian Rassier and Ted Pedersen at the University of Minnesota
- Blinker (`http://www.cs.nyu.edu/~melamed/ftp/papers/styleguide.ps.gz`), created by Dan Melamed at the University of Pennsylvania (Melamed, 1998a,b)
- HandAlign (`http://www.cs.utah.edu/~hal/HandAlign/`), created by Hal Daume III at the University of Utah, originally designed for aligning articles and their summaries
- UMIACS Word Alignment Interface (`http://www.umiacs.umd.edu/~nmadnani/alignment/forclip.htm`), created by Nitin Madnani (based on a tool by Rebecca Hwa) at the University of Maryland
- I*Link (`http://www.ida.liu.se/~nlplab/ILink/`), created by Lars Ahrenberg and his group at Linköping University (Ahrenberg et al., 2002, 2003)

All of these tools are available for download, except for Blinker and I*Link. Most of them are written in Java, except for Alpaco, which is written in Perl/Tk. Alpaco, Blinker, HandAlign and UMIACS do not have built-in automatic alignment, but allow for manual editing of alignment. I*Link, however, has a built-in interactive alignment module which proposes alignments that the user can accept or reject. The user input is then used as learning material by the system. It is also possible to do manual word alignment.

Most of the systems draw the alignments as lines. This is an easy graphical representation of alignment, but it can be problematic when the texts or languages are very different and require many crossing lines. I*Link instead marks the alignment by different colours.

UMIACS seems to only support one type of alignment links. Alpaco, Blinker and I*Link allow for one type of alignment and null alignment (i.e. it is possible to mark a word explicitly as not aligned). HandAlign has two possible types of alignment (certain and possible), marked by different colours. None of the tools, except for I*Link, seem to

have a query function or allow the user to edit base data and neither of the tools have a built-in consistency checker for alignment.

Our own tool, the Stockholm TreeAligner handles alignment of tree structures, in addition to word alignment, which - to our knowledge - is unique. The most similar tool is probably the MTV, Multi Tree Viewer (`http://nlp.cs.nyu.edu/GenPar/mtv.html`), which was created by a number of people in the Language Engineering Workshop at Johns-Hopkins University in 2005 (Burbank et al., 2005). At the moment this tool is only for viewing the structures, but not for creating the alignment, neither automatically nor manually. It would, however, be a very useful tool if it allowed for manipulating alignment since there are several different possible views, where alignment can be seen as lines or boxes in a matrix.

The Stockholm TreeAligner (`http://www.ling.su.se/dali/downloads/treealigner/index.htm`) was created by Joakim Lundborg at Stockholm University in 2006 (see Volk et al., 2006). It was written in Python and is available for download with the source code. Currently the tool does not have built-in automatic alignment, but it allows for manual editing of alignment, which is drawn as lines. There are two possible types of alignment (exact and approximate), marked by different colours.

Some additional features are planned for the future. First, we would like to add automatic consistency checks for the alignment. None of the tools we have looked at have that function. Additionally we would like to have the possibility of searching for aligned structures, similar to the analysis tools in I*Link, with the expressive power of a Treebank search tool like TIGERSearch.

Finally, of course, we would like to incorporate algorithms for automatic alignment. This could be done by automatically generating the most probable word alignments and manually adding the rest. Another possibility is the interactive alignment approach used in I*Link, where the user can accept or reject the suggestions given by the system.

5 Conclusions

Creating a parallel treebank is a laborious and time-consuming task. One of our major lessons from the project is the importance of quality checks along the way. Each step in the creation process needs to be inspected before the next step is carried out. Automatic completeness and consistency checks are essential tools to handle this work.

Having elaborate and clear guidelines with many examples is also necessary. Creating the alignment is difficult since this is a type of semantic annotation and there is always a discussion about how much similarity is needed for two expressions to be translation equivalents.

The tools used for creating the parallel treebanks are also of utter importance. We have compared a number of tools for visualization and manipulation of word alignment. However, there does not seem to be a tool available like our own TreeAligner, which visualizes tree structures and allows for manual manipulation of the alignments.

In the future we would like to extend the TreeAligner, by adding automatic consistency checks of the alignment and a query function to search for aligned structures. We would also like to incorporate algorithms for automatic or interactive alignment.

References

Ahrenberg, L., Andersson, M., and Merkel, M. (2002). A System for Incremental and Interactive Word Linking. In *Third International Conference on Language Resources and Evaluation (LREC 2002)*, Las Palmas.

Ahrenberg, L., Merkel, M., and Petterstedt, M. (2003). Interactive word alignment for language engineering. In *EACL '03: Proceedings of the 10th conference on European chapter of the Association for Computational Linguistics (EACL03)*, pages 49–52, Morristown, NJ. Association for Computational Linguistics.

Bies, A., Ferguson, M., Katz, K., and MacIntyre, R. (1995). Bracketing guidelines for treebank II style, Penn treebank project. Technical report, University of Pennsylvania.

Brants, S., Dipper, S., Hansen, S., Lezius, W., and Smith, G. (2002). The TIGER Treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, pages 24–41, Sozopol, Bulgaria.

Burbank, A., Carpuat, M., Clark, S., Dreyer, M., Fox, P., Groves, D., Hall, K., Hearne, M., Melamed, D., Shen, Y., Way, A., Wellington, B., and Wu, D. (2005). Statistical machine translation by parsing: Final report of the 2005 language engineering workshop. Technical report, Center for Language and Speech Processing, Johns Hopkins University, Baltimore, MD.

Dickinson, M. (2006). Rule equivalence for error detection. In *Proceedings of the Fifth Workshop on Treebanks and Linguistic Theories (TLT 2006)*, Prague, Czech Republic.

Dickinson, M. and Meurers, W. D. (2005). Prune Diseased Branches to Get Healthy Trees! How to Find Erroneous Local Trees in a Treebank and Why It Matters. In *Proceedings of the Fourth Workshop on Treebanks and Linguistic Theories (TLT 2005)*, Barcelona, Spain.

Gale, W. A. and Church, K. W. (1993). A program for aligning sentences in bilingual corpora. *Computational Linguistics*, 19(1):75–102.

Groß, B. (1998). Vergleichende Untersuchung von Alignment-Tools. Saarbrücker Studien zu Sprachdatenverarbeitung und Übersetzen 15, Fachrichtung 8.6 - Angewandte Sprachwissenschaft sowie Übersetzen und Dolmetschen.

Melamed, I. (1998a). Manual annotation of translational equivalence: The Blinker project. Technical Report IRCS 98-07, Department of Computer and Information Science, University of Pennsylvania.

Melamed, I. D. (1998b). Annotation style guide for the blinker project. *CoRR*, cmp-lg/9805004.

Och, F. J. and Ney, H. (2000). Improved statistical alignment models. In *Proc. Of the 38th Annual Meeting of the Association for Computational Linguistics*, pages 440–447, Hongkong, China.

Och, F. J. and Ney, H. (2003). A systematic comparison of various statistical alignment models. *Computational Linguistics*, 29(1):19–51.

Samuelsson, Y. and Volk, M. (2004). Automatic node insertion for treebank deepening. In *Proceedings of the 3rd Workshop on Treebanks and Linguistic Theories (TLT2004)*, Tübingen, Germany.

Samuelsson, Y. and Volk, M. (2005). Presentation and representation of parallel treebanks. In *Proceedings of the Treebank-Workshop at Nodalida*, Joensuu, Finland.

Samuelsson, Y. and Volk, M. (2006). Phrase alignment in parallel treebanks. In *Proceedings of 5th Workshop on Treebanks and Linguistic Theories*, Prague, Czech Republik.

Skut, W., Krenn, B., Brants, T., and Uszkoreit, H. (1997). An annotation scheme for free word order languages. In *Proceedings of the 5th Conference on Applied Natural Language Processing*, pages 88–95, Washington, DC.

Smith, N. A. and Jahr, M. E. (2000). Cairo: An alignment visualization tool. In *Proceedings of the 2nd Conference on Language Resources and Evaluation (LREC)*, pages 549–551, Athens, Greece.

Tiedemann, J. (2003). *Recycling Translations - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing.* PhD thesis, Uppsala university.

Volk, M., Gustafson-Capková, S., Lundborg, J., Marek, T., Samuelsson, Y., and Tidström, F. (2006). XML-based Phrase Alignment in Parallel Treebanks. In *Proceedings of Workshop on Multi-dimensional Markup in Natural Language Processing, EACL2006*, Trento, Italy.