Presentation and representation of parallel treebanks

Samuelsson, Y ; Volk, Martin

Presentation and Representation of Parallel Treebanks

Yvonne Samuelsson and Martin Volk
Department of Linguistics
Stockholm University

Abstract

We have created a small German-Swedish-French parallel treebank. The German and Swedish sentences were Part-of-Speech tagged and parsed with phrase structure in a similar way and then aligned on the sentence and phrase level. The French sentences were annotated with an external system combining constituency and dependency analysis and then aligned to the Swedish sentences.

The treebanks are represented with TIGER-XML and the alignment information is contained in a separate XML document. The parallel trees can be viewed with help of the Stockholm Alignment Viewer, based on SVG-files. The program allows for many possibilities in marking and visualizing special traits of the parallel treebank.

1 Introduction

The combination of 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 (phrase and clause level). Parallel treebanks can be used as training or evaluation corpus for word and phrase alignment, as input for example-based machine translation (EBMT), as a training corpus for transfer rules or a corpus for translation studies, to name some applications.

We have developed and aligned a small German-Swedish parallel treebank. In this paper we will report on the representation of the alignment and the tools that we have developed for its presentation.

2 Building the treebanks

Our parallel treebanks contain the first chapter of Jostein Gaarder's novel "Sofie's World" (the original is the Norwegian [Gaarder 1991]). The first parallel treebank contains the first chapter of the novel in German and Swedish (see e.g. [Samuelsson 2004]). Later, the French version was added and aligned

to the Swedish treebank (see [Tidström 2005]). The first chapter contains about 225 sentences (there is some variation between the different language versions). The first 100 sentences were then aligned on the phrase level, the German-Swedish version by Yvonne Samuelsson and the Swedish-French by Frida Tidström[1]. This work has been part of the Nordic Treebank Network[2].

In creating the German-Swedish parallel treebank, we have annotated both our German and our Swedish treebank with the Annotate tool[3]. It includes Thorsten Brants' statistical Part-of-Speech Tagger and Chunker. The PoS tagger is trained with the STTS (Stuttgart-Tübingen TagSet [Thielen et al. 1999]) for German.

The chunker follows the NEGRA/TIGER annotation guidelines [Skut et al. 1997, Brants et al. 2002], which gives a flat phrase structure tree. This means, for instance, no unary nodes, no "unnecessary" NPs (noun phrases) within PPs (prepositional phrases) and no finite VPs (verb phrases). Using a flat tree structure for manual treebank annotation has two advantages for the human annotator: fewer annotation decisions, and a better overview of the trees. This comes at the prize of the trees not being complete from a linguistic point of view. In addition to the linguistic drawbacks of the flat syntax trees, they are also problematic for node alignment in a parallel treebank. Our goal is to align sub-sentential units (such as phrases and clauses) to get fine-grained correspondences between languages. We prefer to have "deep trees" to be able to draw the alignment between the German sentences and the parallel Swedish sentences on as many levels as possible; in fact, the more detailed the sentence structure is, the more expressive is our alignment.

Figure 1 shows our work flow for the German-Swedish parallel treebank. We first annotated the German sentences semi-automatically, in the flat manner, according to the TIGER guidelines ([Brants et al. 2000] and [Albert et al. 2003]) and we then automatically deepened the flat syntax trees. This was achieved by a program, which automatically and unambiguously inserts nodes to create the deeper structure. This procedure is described in detail in [Samuelsson and Volk 2004].

We annotated the Swedish sentences by first tagging them with a Part-of-Speech tagger trained on SUC (the Stockholm-Umeå Corpus). Since we did not have a Swedish treebank to train a Swedish chunker, we used a trick to apply the German chunker for Swedish sentences. We mapped the Swedish Part-of-Speech tags in the Swedish sentences to the corresponding German

[1]We would like to thank Jörg Tiedemann, Eckhard Bick, Declan Groves and Andy Way for their help in this process.

[2]http://w3.msi.vxu.se/∼nivre/research/nt.html

[3]http://www.coli.uni-sb.de/sfb378/negra-corpus/annotate.html

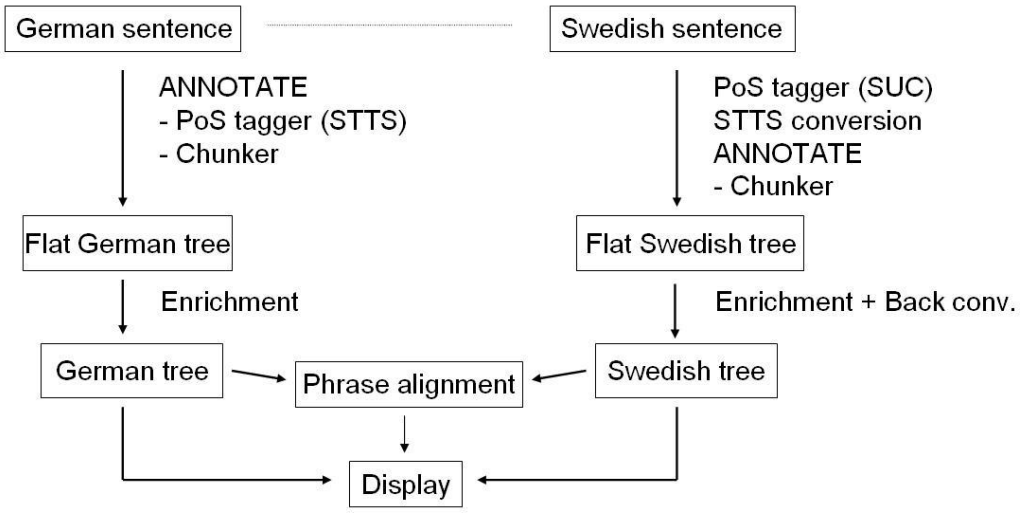

Figure 1: The process of creating the parallel treebank, step-by-step.

tags. Since the German chunker works on these tags, it then suggested constituents for the Swedish sentences, assuming they were German sentences. These experiments and the resulting time gain were reported in [Volk and Samuelsson 2004]. Upon completion of the Swedish treebank with flat syntax trees, we applied the same deepening method as for German and we then converted the Part-of-Speech labels back to the Swedish labels.

A closer look at the treebanks revealed some interesting counts. For example, as can be seen in table 1, the sentences of the German treebank are short, with an average of only 14 words per sentence. We can also see that from the 225 sentences, we extract over 500 different grammar rules. This is a lot, but also means that most rules only have one occurence. However, the most frequent rules occur more than 100 times in the corpus. This means that it is possible to extract interesting information even from small treebanks like ours.

In annotating the French version, the sentences were submitted to a French parser, Eckhard Bick's French Annotation Grammar (FrAG), developed for the VISL ("Visual Interactive Syntax Learning") project at the University of Southern Denmark. FrAG, a Constraint Grammar parser, tags the data with around 20 Part-of-Speech tags, using a probabilistic tagger and morphological rules for Part-of-Speech correction, based on a lexicon of 57,000 lexemes. This is subjected to a syntactic Constraint Grammar pars-

Number of sentences	225
Number of tokens	3146
Average number of tokens / sent	14.0
Number of nodes (= rule tokens)	2278
Number of grammar rules types	513
Most frequent (non-unary) rules:	
PP → P NP	144
NP → Det N	121

Table 1: Some numbers for the German treebank.

ing, which uses 1,200 hand-written rules, and provides a shallow dependency analysis. This in turn is processed by a Phrase Structure Grammar, which uses 200 rules to convert the data into deeper constituent tree structures ([Bick 2003, Bick 2004]). The output then contains Part-of-Speech tags with morphological features, lemmas, constituency and dependency analysis and grammatical functions for phrases and individual words.

FrAG is thus a hybrid system, using both rules and probabilistic methods, and providing information about both constituency and dependency analysis. The FrAG system uses three form levels: clause, group and word. Furthermore, there are three types of clauses: finite clause, non-finite clause and averbal clause. At the group level, there are two main functions: Heads and Dependents, in a dependency grammar perspective. This means that the French syntax annotation is very different from the Swedish annotation which we had modeled after the German annotation guidelines. It results in flatter trees which provide less phrase nodes for cross-language phrase alignment. We envision that a deepening step as outlined above will remedy this problem.

3 Representation of Alignment

After finishing the monolingual treebanks with Annotate, the trees were exported from the accompanying MySQL database and imported into TIGER-Search, a dedicated treebank query tool. This import step creates, as a side effect, a TIGER-XML version of the treebanks. TIGER-XML is a line-based (i.e. database-oriented) representation for graph structures, which includes syntax trees with node labels, edge labels, multiple features on the word level

and even crossing edges[4].

In a TIGER-XML graph each leaf (= token) and each node (= linguistic constituent) has a unique identifier which is prefixed with the sentence number. Leaves are numbered from 1 to n and nodes from 500 to m (under the plausible assumption that no sentence will ever have more than 499 tokens). As can be seen in example 1, node 500 in sentence 5 is of the category prepositional phrase. The phrase consists of word number 4, which is the preposition *über*, plus node 503 which in turn is marked as noun phrase (NP).

(1)
```
<s id="s5">
<graph root="s5_502">
<terminals>
  <t id="s5_1" word="Sie" pos="PPER" morph="--" />
  <t id="s5_2" word="hatten" pos="VAFIN" morph="--" />
  <t id="s5_3" word="sich" pos="PRF" morph="--" />
  <t id="s5_4" word="&#x00fc;ber" pos="APPR" morph="--" />
  <t id="s5_5" word="Roboter" pos="NN" morph="--" />
  <t id="s5_6" word="unterhalten" pos="VVPP" morph="--" />
  <t id="s5_7" word="." pos="$." morph="--" />
</terminals>
<nonterminals>
  <nt id="s5_500" cat="PP">
    <edge label="HD" idref="s5_4" />
    <edge label="NK" idref="s5_503" />
  </nt>
[...]
  <nt id="s5_503" cat="NP">
    <edge label="HD" idref="s5_5" />
  </nt>
[...]
</nonterminals>
</graph>
</s>
```

This means that the token identifiers and constituent identifiers are used to represent the nested tree structure. One might wonder why tree nesting is not directly mapped into XML nesting. But the requirement that the representation format must support crossing edges rules out this option.

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

TIGER-XML is a powerful representation format and is typically used with constituent symbols on the nodes and functional information on the edge labels. This constitutes a combination of constituent structure and dependency structure information[5].

The unique node identifiers can be used for the phrase alignment across parallel trees (more precisely: across trees in corresponding translation units). We decided to also use an XML representation for storing the alignment[6]. Thus the entry in this XML file, as in example 2, represents the alignment of node 507 in sentence 11 of language one (German) to node 500 in sentence 12 of language two (Swedish).

(2)
```
<phraseLink xtargets="11_507 ; 12_500"/>
```

This representation allows phrase alignments within m:n sentence alignments, which we have used in our project. One example of this can be seen in example 3. The XML also allows m:n phrase alignments, which we however did not use. The main guidelines for alignment can be seen in table 2.

(3)
```
<sentLink xtargets="30-31 ; 29-30">
   <phraseLink xtargets="30_501 ; 29_500"/>
   [...]
   <phraseLink xtargets="30_500 ; 30_509"/>
   <phraseLink xtargets="31_503 ; 30_506"/>
   [...]
</sentLink>
```

4 Presentation of Parallel Treebanks

Building a parallel treebank is one thing, but there also has to be a way to view the alignment of the parallel treebank. This is important not only after finishing the treebank, but also during the creation of the alignment. One should be able to see the alignment and, if possible, be able to change the alignment and check the result.

To solve the problem we have developed an alignment viewer based on SVG graphics. SVG (Scalable Vector Graphics) describes vector graphics in

[5]Researchers at Växjö University and the Copenhagen Business School have shown that TIGER-XML can also be used to represent pure dependency structures.

[6]The DTD for the alignment file was inspired by the liu-align-DTD, which we have used with kind permission from Lars Ahrenberg at Linköping University.

Two nodes are aligned if the words which they span convey the same meaning and could serve as translation units. Use m:n sentence alignments. The node alignment is deterministic; a node in one language can never be aligned to more than one node in the other language (even if m:n phrase alignment is technically possible). The alignment should be as detailed as possible, i.e. align all nodes except those that do not have any correspondence in the other language.

Table 2: Some guidelines for the alignment (DE-SV).

XML. According to the W3C[7] SVG allows three types of graphic objects: vector graphic shapes, raster graphics and text. SVGs can be static or animated. Some browsers (like Amaya and a version of Mozilla) have (partial) SVG implementations, but mostly a plug-in is needed for viewing SVGs. There are several SVG-viewers available for free download over the web, for instance Corel's[8] and Adobe's[9]. Batik[10] is a Java-based toolkit for viewing, generating and manipulating SVGs. We are mostly using Corel's SVG plugin because it allows for window-optimized zooming (i.e. the graph is automatically zoomed to be fully displayed in the given window), even though this viewer does not support searching through the textual elements of a graph within the SVGs, as some other viewers do.

We used TIGERSearch as an intermediary step in creating the SVGs for our viewer, which we call the Stockholm Alignment Viewer. TIGERSearch allows us to export a single tree (or the whole forest of all trees from a treebank) in SVG format. This means that we do not have to bother to program the layout of the tree structure, we simply take the tree layout as computed by TIGERSearch.

The Stockholm Alignment Viewer is a Perl program, which needs three files as input: one SVG-file with the trees from language one, one SVG-file with the trees from language two and the alignment file. From these

[7]See http://www.w3.org/Graphics/SVG/

[8]http://www.smartgraphics.com/Viewer_prod_info.shtml

[9]http://www.adobe.com/svg/

[10]http://xml.apache.org/batik/

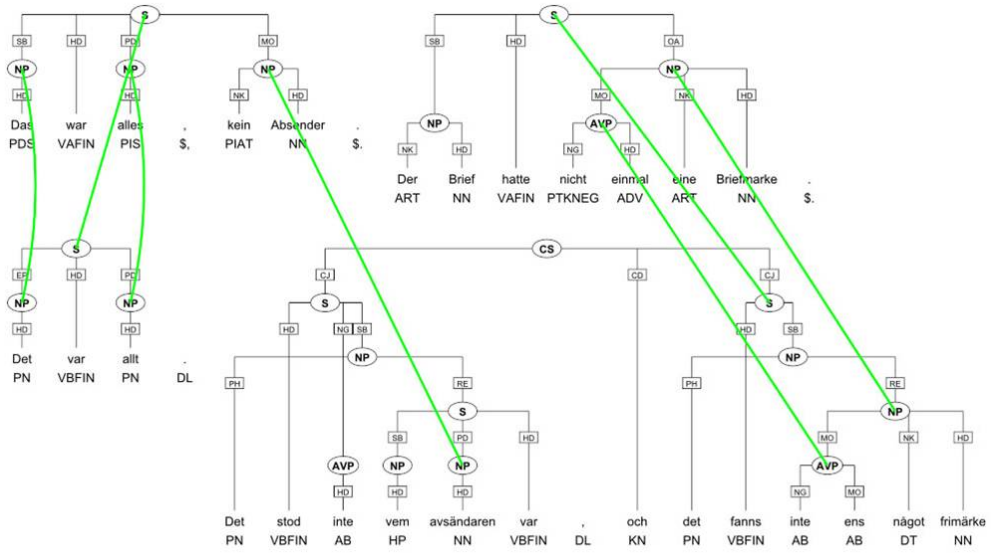

Figure 2: m to n sentence alignment.

three files the program creates new SVG-files, one for each translation unit (with m:n sentences). In the output files the trees of the two languages are placed above each other, with the alignment information shown as colored lines between the nodes. This creates the visual representation, which can be displayed in a browser with the help of an SVG-viewer.

In figure 2 we see two German sentences aligned to two Swedish sentences. We need to be able to view all four sentences together, since one phrase of the first German sentence corresponds to a phrase in the second Swedish sentence.

The latest version of the Stockholm Alignment Viewer uses different colors depending on whether the aligned nodes have the same name or not. For example, if an NP node of the German tree is aligned to an NP node in the Swedish tree, then the alignment is displayed in green. But if a prepositional phrase in tree one is aligned to an adverbial phrase in tree two, then the alignment is displayed in a different color, so that interesting translation variations can easily be spotted.

One problem with the alignment comes from the fact that the sentence structure might change in the translation. One example of this is given as example 4.

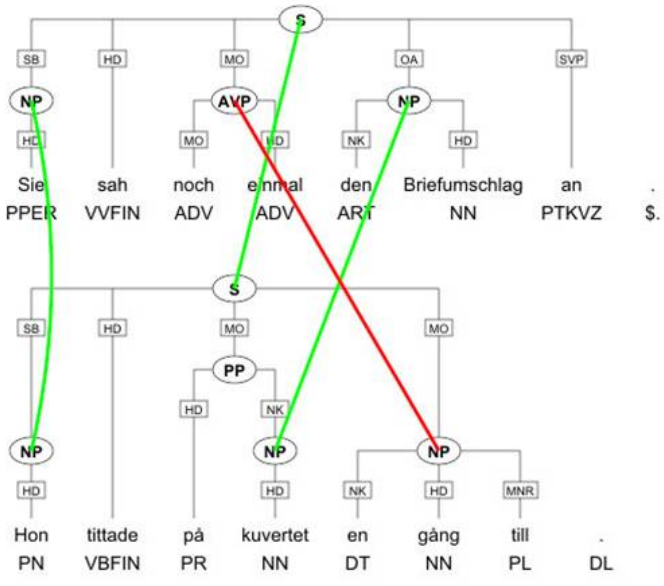

Figure 3: Marking nodes of different type in a different colour.

(4) Sofie wohnte am Ende eines ausgedehnten Viertels mit Einfamilienhäusern und hatte einen fast doppelt so langen Schulweg wie Jorunn.

Sofie, som bodde i utkanten av ett stort villaområde, hade nästan dubbelt så långt till skolan som Jorunn.

The German sentence consists of two coordinated clauses, where the NP *Sofie* is the subject. The Swedish sentence, however, is not coordinated and the subject not only contains the name *Sofie* but also a relative clause. These two nodes are clearly not direct translation equivalents, our alignment is just not fine-grained enough. The NPs with the name *Sofie* should of course be aligned, but the alignment representation must be extended to express the exclusion of sub-constituents. This would allow us to say: align a node n_1 from tree one to a node n_2 from tree two, but exclude the sub-constituent node n_3 from n_2, in this case excluding the Swedish relative clause from the alignment of the NP's.

This exclusion can be marked as in example 5, where for instance node 506 has been excluded from Swedish node 507, which is aligned to the German node 514.

(5)
```
<sentLink  xtargets="10 ; 10">
    <phraseLink  xtargets="10_507 ; 10_506"/>
    <phraseLink  xtargets="10_514 ; 10_507{-506} "/>
[...]
    <phraseLink  xtargets="10_509 ; 10_508{-507} "/>
</sentLink>
```

This can be displayed in the SVG-representation as shown in figure 4 (part of the tree). The dotted lines indicate exclusion from the alignment. The Swedish tree contains two exclusion markers. The relative clause (RC) is excluded from the noun phrase (NP) alignment and the subject (SB) noun phrase is itself excluded from the sentence alignment. Note that the dotted lines do not represent any tree-internal information.

5 Conclusion

We have shown a straightforward way to tie in XML-based phrase alignment information with syntax trees represented in TIGER-XML. And we have argued for the use of Scalable Vector Graphics as a means to visualize phrase alignment.

An alternative way to display the information captured in the sub-sentential alignment is by listing the respective tokens that are covered by each node in a table side by side. For example the alignment between the prepositional phrases *in den Briefkasten* and *i brevlådan* (in the mail-box) will result in a table as shown in example 6

(6)

in den Briefkasten	i brevlådan
irgendwann	en gång i tiden

The corresponding pairs should be good translation units. One way to spot alignment problems is to measure the length (as number of characters) of the translation units. First experiments show that the alignment is often not precise if the length of the two translation units differs by more than 30 percent (relative to the longer unit).

Now that we have phrase alignments between German and Swedish trees as well as between (the same) Swedish and corresponding French trees, we would like to check if the alignment relation is transitive. Can we automatically infer phrase alignment between the German and French trees?

We used automatically computed word alignments (which was kindly provided by Jörg Tiedemann, details about the word alignment can be found in [Tiedemann 2003]) to predict phrase alignment. The automatic phrase

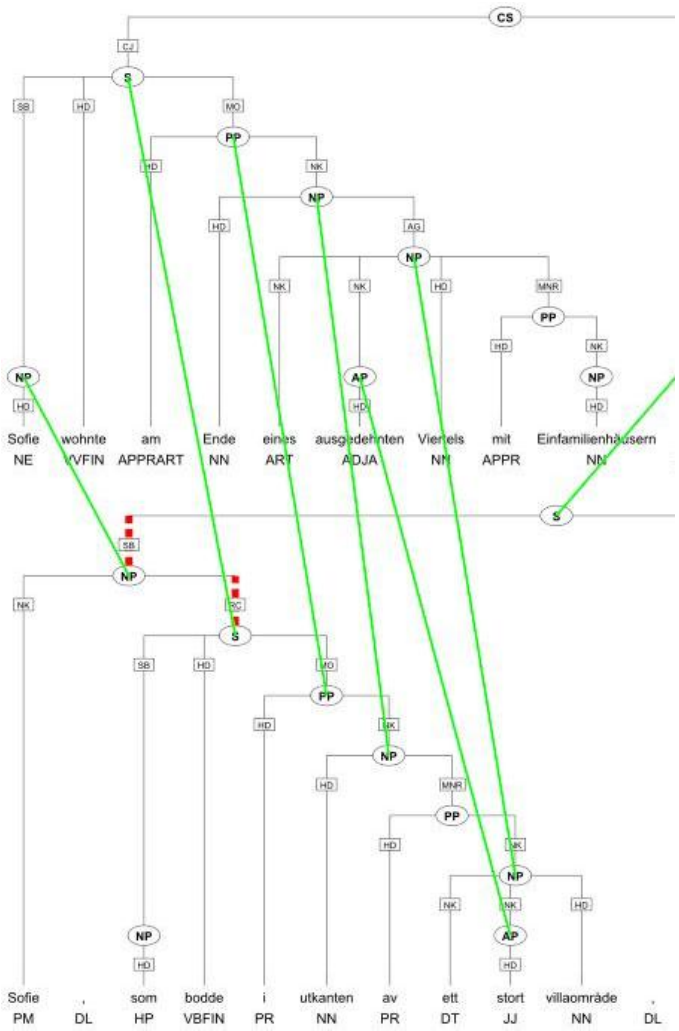

Figure 4: Excluding nodes that are not part of the alignment.

alignment had to be manually checked, and we also created a gold standard to compare the automatic alignment to. This was done by manually checking the entries in the alignment file. A future step is to have a graphical user interface (preferably coupled with an automatic phrase alignment tool) that allows us to manipulate the alignment directly.

References

[Albert et al. 2003] S. Albert, J. Anderssen, R. Bader, S. Becker, T. Bracht, S. Brants, T. Brants, V. Demberg, S. Dipper, P. Eisenberg, S. Hansen, H. Hirschmann, J. Janitzek, C. Kirstein, R. Langner, L. Michelbacher, O. Plaehn, C. Preis, M. Pussel, M. Rower, B. Schrader, A. Schwartz, G. Smith, and H. Uszkoreit. 2003. TIGER Annotationsschema. July.

[Bick 2003] Eckhard Bick. 2003. A CG & PSG hybrid approach to automatic corpus annotation. In *Proceedings of SProLaC2003. Corpus Linguistics 2003*, Lancaster.

[Bick 2004] Eckhard Bick. 2004. Parsing and evaluating the French Europarl corpus. In *Méthodes et Outils Pour L'evaluation Des Analyseurs Syntaxiques (Journée ATALA, May 15, 2004)*, pages 4–9, Paris. ATALA.

[Brants et al. 2000] T. Brants, S. Dipper, P. Eisenberg, S. Kramp, C. Preis, M. Pussel, A. Schwartz, G. Smith, and H. Uszkoreit. 2000. TIGER Annotationsschema. May.

[Brants et al. 2002] S. Brants, S. Dipper, S. Hansen, W. Lezius, and G. Smith. 2002. The TIGER treebank. In *Proceedings of the Workshop on Treebanks and Linguistic Theories*, Sozopol.

[Gaarder 1991] Jostein Gaarder. 1991. *Sofies verden: Roman om filosofiens historie*. Aschehoug.

[Samuelsson and Volk 2004] Yvonne Samuelsson and Martin Volk. 2004. Automatic node insertion for treebank deepening. Manuscript submitted to TLT2004.

[Samuelsson 2004] Yvonne Samuelsson. 2004. Parallel Phrases - Going Automatic - Experiments towards a German-Swedish parallel treebank. D-uppsats, Stockholm University, http://ling16.ling.su.se:8080/PubDB/doc_repository/ samuelssonautomatic2004.pdf.

[Skut et al. 1997] W. Skut, B. Krenn, T. Brants, and H. Uszkoreit. 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.

[Thielen et al. 1999] C. Thielen, A. Schiller, S. Teufel, and C. Stöckert. 1999. Guidelines für das Tagging Deutscher Textkorpora mit STTS. Technical report, IMS and SfS.

[Tidström 2005] Frida Tidström. 2005. Extending a Parallel Treebank with Data in French. C-uppsats, Stockholm University, http://ling16.ling.su.se:8080/ new_PubDB/doc_repository/ 212_frida-tidstrom-2005.pdf.

[Tiedemann 2003] Jörg Tiedemann. 2003. *Recycling Translations - Extraction of Lexical Data from Parallel Corpora and their Application in Natural Language Processing*. Ph.D. thesis, Uppsala university.

[Volk and Samuelsson 2004] Martin Volk and Yvonne Samuelsson. 2004. Bootstrapping parallel treebanks. In *Proceedings of the 5th International Workshop on linguistically Interpreted Corpora (COLING 2004)*, pages 63–69, Geneva, Switzerland, August.