Bootstrapping Parallel Treebanks

Martin VOLK and Yvonne SAMUELSSON
Stockholm University
Department of Linguistics
10691 Stockholm
Sweden
volk@ling.su.se

Abstract

This paper argues for the development of parallel treebanks. It summarizes the work done in this area and reports on experiments for building a Swedish-German treebank. And it describes our approach for reusing resources from one language while annotating another language.

1 Introduction

Treebanks have become valuable resources in natural language processing (NLP) in recent years (Abeillé, 2003). A treebank is a collection of syntactically annotated sentences in which the annotation has been manually checked so that the treebank can serve as training corpus for natural language parsers, as repository for linguistic research, or as evaluation corpus for NLP systems. The current interest in treebanks is documented in international workshop series like “Linguistically Interpreted Corpora (LINC)” or “Treebanks and Linguistic Theories” (TLT). But also the recent international CL conferences have seen a wide variety of papers that involved treebanks. Treebanks have become a necessary resource for many research activities in NLP.

On the other hand recent years have seen an increasing interest in parallel corpora (often called bitexts). See for example (Melamed, 2001) or (Borin, 2002) for a broad picture of this area.

But surprisingly little work has been reported on combining these two areas: parallel treebanks. We define a parallel treebank as a bitext where the sentences of each language are annotated with a syntactic tree, and the sentences are aligned below the clause level. This leaves room for various kinds of tree structure (e.g. dependency structure trees or constituent structure trees) and does not specify a precise requirement for tree alignments but rather for some sort of sub-clausal alignment (e.g. word alignment or phrase alignment).

But why has there been so little work done on parallel treebanks? The benefits of having such a treebank for training statistical machine translation systems, experimenting with example-based translation systems, or evaluating word alignment programs seem so overwhelming. We speculate that this scarcity is mainly due to the expenses necessary for building a parallel treebank (in terms of time and human resources). It is well known that the manual labor involved in building a monolingual treebank is high (For the Penn Treebank (Taylor et al., 2003) report on 750 - 1000 words per hour for an experienced annotator, which translates to 35 - 50 sentences per hour). And the cross-language alignment requires additional work. Therefore every approach to facilitate and speed up this process will be highly welcome.

The goal of this paper is to summarize the (little) work that has been done on parallel treebanks and related areas such as annotation projection. In particular we will report on our experiments for building a Swedish-German parallel treebank. As a side issue we investigated whether the German treebank annotation guidelines (from the NEGRA / TIGER projects) can be applied to Swedish. We have chosen Swedish and German because they are our mother tongues, but also because they are similar and still interestingly different.

2 Previous Work on Parallel Treebanks

The field of parallel treebanks is only now evolving into a research field. (Cmejrek et al., 2003) at the Charles University in Prague have built a treebank for the specific purpose of machine translation, the Czech-English Penn Treebank with tectogrammatical dependency trees. They have asked translators to translate part of the

Penn Treebank into Czech with the clear directive to translate every English sentence with one in Czech and to stay as close as possible to the original.

This directive seems strange at first sight but it makes sense with regard to their objective. Since they specifically construct the treebank for training and evaluating machine translation systems, a close human translation is a valid starting point to get good automatic translations.

At the University of Münster (Germany) (Cyrus et al., 2003) have started working on FuSe, a syntactically analyzed parallel corpus. The goal is a treebank with English and German texts (currently with examples from the Europarl corpus). The annotation is multi-layered in that they use PoS-tags, constituent structure, functional relations, predicate-argument structure and alignment information. However their focus is on the predicate-argument structure.

The Nordic Treebank Network[1] has started an initiative to syntactically annotate the first chapter of "Sophie´s World"[2] in the nordic languages. This text was chosen since it has been translated into a vast number of languages and since it includes interesting linguistic properties such as direct speech. Currently a prototype of this parallel treebank with the first 50 sentences in Swedish, Norwegian, Danish, Estonian and German has been finished. The challenge in this project is that all involved researchers annotate the Sophie sentences of their language in their format of choice (ranging from dependency structures for Danish and Swedish to constituency structures for Estonian and German). In order to make the results exchangeable and comparable all results have been converted into TIGER-XML so that TIGERSearch[3] can be used to display and search the annotated sentences monolingually. The alignment across languages is still open.

3 Bootstrapping a German-Swedish parallel treebank

We have built a small German-Swedish parallel treebank with 25 sentence pairs taken from the Europarl corpus. First, the German sentences were tokenized and loaded into the Annotate treebank editor[4]. Annotate includes Thorsten Brants' Part-of-Speech Tagger and Chunker for German. The PoS tagger employs the STTS, a set of around 50 PoS-tags for German. The set is so large because it incorporates some morpho-syntactic features (e.g. it distinguishes between finite and non-finite verb forms). The chunker assigns a flat constituent structure with the usual node labels (e.g. AP, NP, PP, S, VP), but also special labels for coordinated phrases (e.g. CAP, CNP, CPP, CS, CVP). In addition the chunker suggests syntactic functions (like subject, object, head or modifier) as edge labels. The human treebank annotator controls the suggestions made by the tagger and the chunker and modifies them where necessary. Tagger and chunker help to speed up the annotation process for German sentences enormously. The upper tree in figure 1 shows the structure for the following sentence (taken from Europarl):

(1) *Doch sind Bürger einiger unserer Mitgliedstaaten Opfer von schrecklichen Naturkatastrophen geworden.*
(EN: But citizens of some of our member states have become victims of terrible natural disasters.)

Now let us look at the resources available for Swedish. First there is SUC (the Stockholm-Umeå-Corpus), a 1 million word corpus of written Swedish designed as a representative corpus along the lines of the Brown corpus. SUC contains PoS-tags, morphological tags and lemmas for all tokens as well as proper name classes. All the information is hand-checked. So this is proper training material for a PoS tagger. Compared to the 50 tags of the STTS, the 22 SUC PoS-tags (e.g. only one verb tag) are rather coarse-grained, but of course we can use the combination of PoS-tags and morphological information to automatically derive a richer tag set.

Training material for a Swedish chunker is harder to come by. There are two early Swedish treebanks, Mamba and SynTag (dating back to the 1970s (!) and 1980s respectively), but they are rather small (about 5000 sentences each), very heterogeneously annotated and somewhat faulty (cf. (Nivre, 2002)). Therefore, the most serious attempt at training a

[1] The Nordic Treebank Network is headed by Joakim Nivre. See www.masda.vxu.se/∼nivre/research/nt.html

[2] The Norwegian original is: Jostein Gaarder (1991): Sofies verden: roman om filosofiens historie. Aschehoug.

[3] TIGERSearch is a treebank query tool developed at the University of Stuttgart. See also section 5.2.

[4] Annotate is a treebank editor developed at the University of Saarbrücken. See www.coli.uni-sb.de/sfb378/negra-corpus/annotate.html

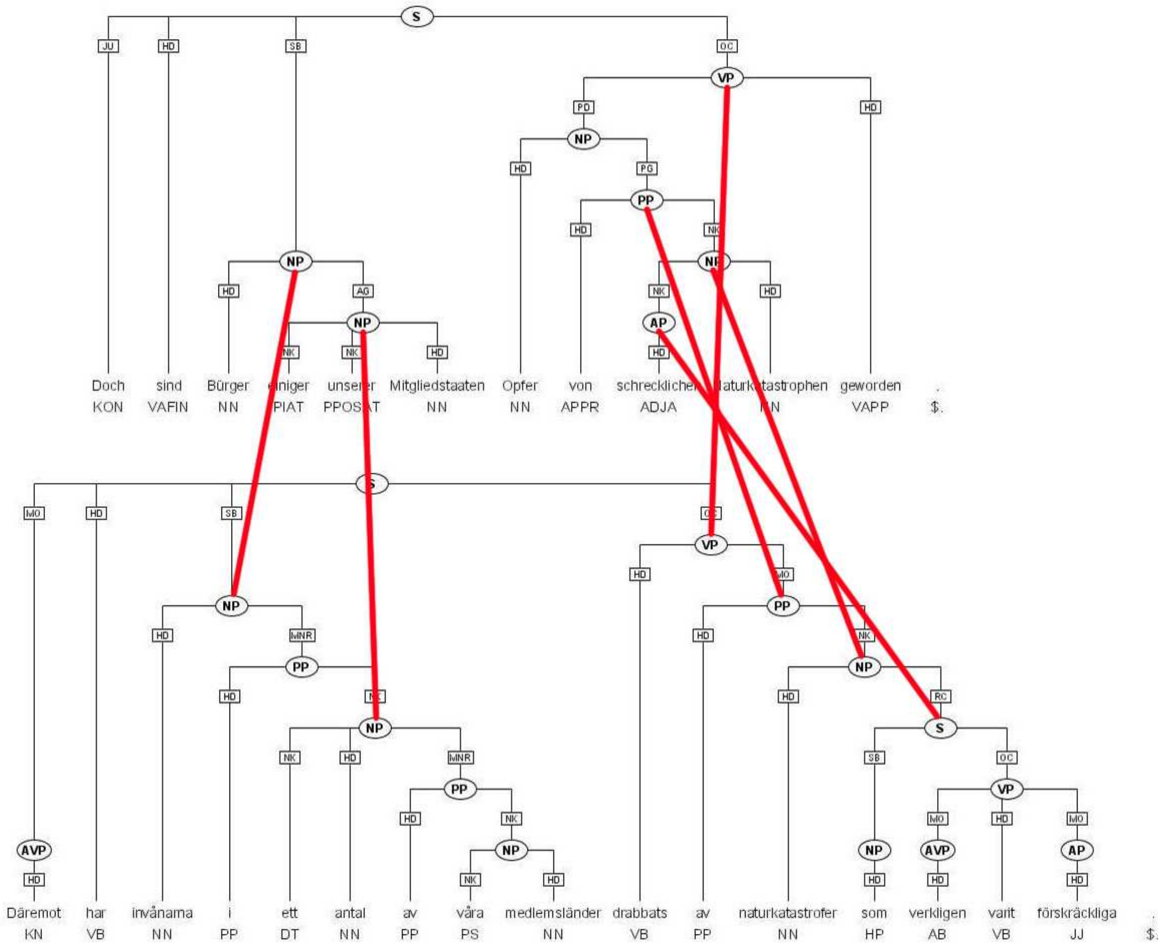

Figure 1: Parallel trees with lines showing the alignment.

chunker for Swedish was based on an automatically created "treebank" which of course contained a certain error rate (Megyesi, 2002). Essentially there exists no constituent structure treebank for Swedish that could be used for training a chunker with resulting structures corresponding to the German sentences.

Therefore we have worked with a different approach (described in detail in (Samuelsson, 2004)). We first trained a PoS tagger on SUC and used it to assign PoS-tags to our Swedish sentences. We then converted the Swedish PoS-tags in these sentences into the corresponding German STTS tags.[5] We loaded the Swedish sentences into Annotate (now with STTS tags), and we were then able to reuse the German chunker to make structural decisions over the Swedish sentences. This worked surprisingly well due to the structural similarities of Swedish and German. After the semi-automatic annotation of the syntactic structure, the PoS-tags were converted back to the usual Swedish tag set. This is a straight-forward example of how resources for one language (in this case German) can be reused to bootstrap linguistic structure in another albeit related language (here Swedish).

The lower tree in figure 1 shows the structure for the Swedish sentence which corresponds to the German sentence in example 1.

(2) *Däremot har invånarna i ett antal av våra medlemsländer drabbats av naturkatastrofer som verkligen varit förskräckliga.*
(EN: However inhabitants of a number of our member states were affected by natural disasters which indeed were terrible.)

[5] An alternative approach could have been to map all tags in the SUC to STTS and then train a Swedish tagger on this converted material.

Since the German STTS is more fine-grained than the SUC tag set, the mapping from the SUC tag set to STTS does not entail loosing any information. When converting in this direction the problem is rather which option to choose. For example, the SUC tag set has one tag for adjectives, but the STTS distinguishes between attributive adjectives (ADJA) and adverbial or predicative adjectives (ADJD). We decided to map all Swedish adjectives to ADJA since the information in SUC does not give us any clue about the usage difference. The human annotator then needs to correct the ADJA tag to ADJD if appropriate, in order to enable the chunker to work as intended.

Other tag mapping problems come with the SUC tags for adverb, determiner, pronoun and possessive all of which are marked as "interrogative or relative" in the guidelines. There is no clear mapping of these tags to STTS. We decided to use the mapping in table 1.

The benefit of using the German chunker for annotating the Swedish sentences is hard to quantify. A precise experiment would require one group of annotators to work with this chunker and another to work without it on the same sentences for a comparison of the time needed.

We performed a small experiment to see how often the German chunker suggests the correct node labels and edge labels for the Swedish sentences (when the children tags/nodes were manually selected). In 100 trials we observed 89 correct node labels and 93% correct edge labels (for 305 edges). If we assume that manual inspection of correct suggestions takes about a third of the time of manual annotation, and if we also assume that the correction of erroneous suggestions takes the same amount of time as manual annotation, then the employment of the German chunker for Swedish saves about 60% of the annotation time.

Reusing a chunker for bootstrapping a parallel treebank between closely related languages like German and Swedish is only a first step towards reusing annotation (be it automatic or manual) in one language for another language. But it points to a promising research direction. (Yarowsky et al., 2001) have reported interesting results of an annotation-projection technique for PoS tagging, named entities and morphology. And (Cabezas et al., 2001) have explored projecting syntactic dependency relations from English to Basque. This idea was followed by (Hwa et al., 2002) who investigated English to Chinese projections based on the direct correspondence assumption. They conclude that annotation projections are nearly 70% accurate (in terms of unlabelled dependencies) when some linguistic knowledge is used. We believe that annotation projection is a difficult field but even if we only succeed in a limited number of cases, it will be valuable for increased speed in the development of parallel treebanks.

3.1 Alignment

The alignment in our experimental treebank is based on the nodes, not the edge labels. Figure 1 shows the phrase alignment as thick lines across the trees. All of the alignment mapping was done by hand.

We decided to make the alignment deterministic, i.e. a node in one language can only be aligned with one node in the other language. There are, of course, a lot of problems with the alignment. We have looked at the meaning, rather than the exact wording. Sometimes different words are used in an S or VP, but we still feel that the meaning is the same, and therefore we have aligned them. We might have alignment on one constituent level, while there are differences (i.e. no alignment) on lower levels of the tree. Therefore we consider it important to make the parse trees sufficiently deep. We need to be able to draw the alignment on as many levels as possible.

Another problem arises when the sentences are constructed in different ways, due to e.g. passivisation or topicalisation. Although German and Swedish are structurally close, there are some clear differences.

- German separable prefix verbs (e.g. *fangen an* = begin) do not have a direct correspondence in Swedish. However, Swedish has frequent particle verbs (e.g. *ta upp* = bring up). But whereas the German separated verb prefix occupies a specific position at the end of a clause ("Rechte Satzklammer"), the Swedish verb particle occurs at the end of the verb group.

- The general word order in Swedish subordinate clauses is the same as in main clauses. Unlike in German there is no verb-final order in subordinate clauses.

- German uses accusative and dative case endings to mark direct and indirect objects. This is reflected in the German function labels for accusative object (OA) and for

SUC tag		STTS tag	
HA	int. or rel. adverb	PWAV	adverbial interrog. or relative pronoun
HD	int. or rel. determiner	PWS	(stand-alone) interrog. pronoun
HP	int. or rel. pronoun	PRELS	(stand-alone) relative pronoun
HS	int. or rel. possessive	PPOS	(stand-alone) possessive pronoun

Table 1: Mapping of SUC tags to STTS

dative object (DO). Swedish has lost these case endings and the labels therefore need not reflect case but rather object function.

Our overall conclusion is that applying the German treebank annotation guidelines to Swedish works well when the few peculiarities of Swedish are taken care of.

4 Corpus representation

After annotating the sentences in both languages with the Annotate treebank editor, the tree structures were exported in the NEGRA export format from the MySQL database. The file in NEGRA format is easily loaded into TIGERSearch via the TIGERRegistry which provides an import filter for this format. This import process creates a TIGER-XML file which contains the same information as the NEGRA file. The difference is that the pointers in the NEGRA format go from the tokens to the pre-terminal nodes (and from nodes to parent nodes) in a bottom-up fashion, whereas in the TIGER-XML file the nodes point to their children by listing their id numbers (idref) and their edge label (in a top-down perspective).

In this file the tokens of the sentence (terminals) are listed beneath each other with their corresponding PoS-tag (PPER for personal pronoun, VVFIN for finite verb, APPRART for contracted preposition etc.). The nodes (nonterminals) are listed with their name and their outgoing edges with labels such as HD for head, NK for noun kernel, SB for subject etc.

```
<s id="s1">
<graph root="522">
<terminals>
 <t id="1" word="Ich"     pos="PPER" />
 <t id="2" word="erkläre" pos="VVFIN"/>
 <t id="3" word="die"     pos="ART"  />
 <t id="4" word="am"    pos="APPRART"/>
 <t id="5" word="Freitag" pos="NN"   />
[...]
</terminals>
```

```
<nonterminals>
  <nt id="500" cat="NP">
    <edge label="HD" idref="1" />
  </nt>
[...]
  <nt id="522" cat="S">
    <edge label="HD" idref="2" />
    <edge label="SB" idref="500" />
    <edge label="MO" idref="511" />
    <edge label="OA" idref="521" />
  </nt>
</nonterminals>
</graph>
</s>
```

Since all tokens and all nodes are uniquely numbered, these numbers can be used for the phrase alignment. For the representation of the alignment we adapted a DTD that was developed for the Linköping Word Aligner (Ahrenberg et al., 2002). The XML-file with the alignment information then looks like this. The sentLink-tags each contain one sentence pair, while each phraseLink represents one aligned node pair.

```
<!DOCTYPE DeSv SYSTEM "align.dtd">
<DeSv fromDoc="De.xml" toDoc="Sv.xml">
  <linkList>
    <sentLink xtargets="1 ; 1">
      <phraseLink xtargets="500; 500"/>
      <phraseLink xtargets="501; 503"/>
      [...]
    </sentLink>
  </linkList>
</DeSv>
```

This fragment first specifies the two involved XML files for German (De.xml) and Swedish (Sv.xml). It then states the phrase pairs for the sentence pair 1 - 1 from these files. For example, phrase number 501 from the German sentence 1 is aligned with phrase number 503 of the Swedish sentence.

5 Tools for Parallel Treebanks

Treebank tools are usually of two types. First there are tools for producing the treebank, i.e. for automatically adding information (taggers, chunkers, parsers) and for manual inspection and correction (treebank editors). On the other hand we need tools for viewing and searching a treebank.

5.1 Treebank Editors

Of course the tools for monolingual treebank production can also be used for building the language-specific parts of a parallel treebank. Thus a treebank editor such as Annotate with built-in PoS tagger and chunker is an invaluable resource. But such a tool should include or be complemented with a completeness and consistency checker.

In addition the parallel treebank needs to be aligned on the sub-sentence level. Automatic word alignment systems will help ((Tiedemann, 2003) discusses some interesting approaches). But tools for checking and correcting this alignment will be needed. For example the I*Link system (Ahrenberg et al., 2002) could be used for this task. I*Link comes with a graphical user interface for creating and storing associations between segments in a bitext. I*Link is aimed at word and phrase associations and requires bitexts that are pre-aligned at the sentence level.

5.2 Treebank Search Tools

With the announcement of the Penn Treebank, some 10 years ago, came a search tool called tgrep. It is a UNIX-based program that allows querying a treebank specifying dominance and precedence relations over trees (plus regular expressions and boolean operators). The search results are bracketed trees in line-based or indented format catering for the needs of different users. For example, the following tgrep query searches for a VP that dominates (not necessarily directly) an NP which immediately precedes a PP.

```
VP << (NP . PP)
```

More recently TIGERSearch was launched. It is a Java-based program that comes with a graphical user interface and a powerful feature-value-oriented query language. The output are graphical tree representations in which the matched part of the tree is highlighted and focused. TIGERSearch's ease of installation and friendly user interface have made it the tool of choice for many treebank researchers.

According to our knowledge no specific search tools for parallel treebanks exist. In addition to the above sketched search options of tgrep and TIGERSearch a search tool for parallel treebanks will have to allow queries that combine constraints over two trees. For example one wants to issue queries such as "Find a tree in language 1 with a relative clause where the parallel tree in language 2 uses a prepositional phrase for the same content."

5.3 Displaying Parallel Trees

There is currently no off-the-shelf tool that can display parallel trees so that one could view two phrase structure trees at the same time with their alignment. Therefore we discuss possible display options of such a future program.

One alternative is to show the two trees above each other (as in figure 1). And there are many ways to visualize the alignment: Either by drawing lines between the nodes (as we did), or by color marking the nodes, or by opening another window where only chosen parallel nodes are shown. The latter case corresponds to a zoom function, but this also entails that the user has to click on a node to view the alignment.

Another alternative would be a mirror imaging. One language would have its tree with the root at the top and the tree of the other language would be below with the root at the bottom. The alignment could be portrayed in the same ways as above.

But then the display problem is mainly a problem concerning the computer screens of today, where a large picture partly lands outside of the screen, while a smaller scale picture might result in words that are too small to be readable. One solution could be to use two screens (as is done in complex layout tasks), but then we cannot have a solution with the trees above each other, but rather next to each other, possibly with some kind of color marking of the nodes.

A last alternative is to use vertical trees, where the words are listed below each other, showing phrase depth horizontally. Then the alignment could be shown by having the nodes side by side instead of above each other. This is the least space consuming alternative, but it is also the least intuitive one. Furthermore, this is not a viable alternative if the trees contain crossing branches.

We currently favor the first approach with two trees above each other, and we have written a program that takes the SVG (scalable vector graphics) representation of two trees (as exported from TIGERSearch), merges the two graphs into a single graph and adds the phrase alignment lines based on the information in the alignment file.

6 Conclusions

We have reported on our experiments for building a German-Swedish parallel treebank. We have shown that by mapping the German PoS tag set to the Swedish tag set we were able to reuse the German chunker for the semi-automatic annotation of the Swedish sentences. Our experiments have also shown that the German annotation guidelines with minor adaptations are well-suited for Swedish.

We have argued that tools for building monolingual treebanks can be used for parallel treebanks as well, and that tools for sub-sentence alignment are available but they are not enough evaluated yet for aligning tree structures. Tools for viewing and searching through parallel treebanks are missing.

7 Acknowledgements

We would like to thank the anonymous reviewers for useful comments, the members of the Nordic Treebank Network for many interesting discussions, and David Hagstrand for handling our annotation databases.

References

Anne Abeillé, editor. 2003. *Building and Using Parsed Corpora*, volume 20 of *Text, Speech and Language Technology*. Kluwer, Dordrecht.

Lars Ahrenberg, Magnus Merkel, and Mikael Andersson. 2002. A system for incremental and interactive word linking. In *Proceedings from The Third International Conference on Language Resources and Evaluation (LREC-2002)*, pages 485–490, Las Palmas.

Lars Borin, editor. 2002. *Parallel Corpora, Parallel Worlds. Selected Papers from a Symposium on Parallel and Comparable Corpora at Uppsala University, Sweden, 22-23 April, 1999.*, volume 43 of *Language and Computers*. Rodopi, Amsterdam.

Clara Cabezas, Bonnie Dorr, and Philip Resnik. 2001. Spanish language processing at University of Maryland: Building infrastructure for multilingual applications. In *Proceedings of the Second International Workshop on Spanish Language Processing and Language Technologies (SLPLT-2)*, Jaen, Spain, September.

Martin Cmejrek, Jan Curin, and Jiri Havelka. 2003. Treebanks in machine translation. In *Proc. Of the 2nd Workshop on Treebanks and Linguistic Theories*, Växjö, Sweden.

Lea Cyrus, Hendrik Feddes, and Frank Schumacher. 2003. FuSe - a multi-layered parallel treebank. In *Proc. Of the 2nd Workshop on Treebanks and Linguistic Theories*, Växjö, Sweden.

Rebecca Hwa, Philip Resnik, Amy Weinberg, and Okan Kolak. 2002. Evaluating translational correspondence using annotation projection. In *Proceedings of the 40th Annual Meeting of the ACL*, Philadelphia.

Beáta Megyesi. 2002. *Data-Driven Syntactic Analysis. Methods and Applications for Swedish.* Doctoral dissertation, Kungl. Tekniska Högskolan. Department of Speech, Music and Hearing, Stockholm.

I. Dan Melamed. 2001. *Empirical Methods for Exploiting Parallel Texts.* MIT Press, Cambridge, MA.

Joakim Nivre. 2002. What kinds of trees grow in Swedish soil? A comparison of four annotation schemes for Swedish. In *Proc. Of First Workshop on Treebanks and Linguistic Theory*, Sozopol, Bulgaria.

Yvonne Samuelsson. 2004. Parallel phrases. Experiments towards a German-Swedish parallel treebank. C-uppsats, Stockholms Universitet.

Ann Taylor, Mitchell Marcus, and Beatrice Santorini. 2003. The Penn Treebank: An overview. In Anne Abeillé, editor, *Building and Using Parsed Corpora*, volume 20 of *Text, Speech and Language Technology.* Kluwer, Dordrecht.

Jörg Tiedemann. 2003. *Recycling Translations. Extraction of Lexical Data from Parallel Corpora and Their Application in Natural Language Processing.* Acta universitatis upsaliensis, Uppsala University.

D. Yarowsky, G. Ngai, and R. Wicentowski. 2001. Inducing multilingual text analysis tools via robust projection across aligned corpora. In *Proceedings of HLT 2001, First International Conference on Human Language Technology Research.*