Automatic node insertion for treebank deepening

Samuelsson, Y; Volk, M

Postprint available at:
http://www.zora.uzh.ch

Posted at the Zurich Open Repository and Archive, University of Zurich.
http://www.zora.uzh.ch

Originally published at:
Automatic Node Insertion for Treebank Deepening

Yvonne Samuelsson and Martin Volk
Stockholm University
Department of Linguistics
106 91 Stockholm
Sweden
yvonnesamuelsson@yahoo.se and volk@ling.su.se

1 Background

Creating a treebank is a time-consuming task. The Part-of-Speech tagging of the sentences is fast and automatic, with only minor corrections afterwards, since there are good taggers out there today. However, parsing is still a task that needs to be done semi-automatically, where the human annotator has to make many decisions manually.

We are working on a German-Swedish parallel treebank, where the data consists of the first chapter of Jostein Gaarder’s novel Sophie’s World (the Norwegian original is [Gaarder 1991]). The German treebank contains 225 sentences and the Swedish one 216 sentences. The initiative for using this text comes from the Nordic Treebank Network1, which has an ongoing project to syntactically annotate the first chapter of this book 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.

For the annotation of the German part we used the treebank editor Annotate2. 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 rather 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).

---

1The Nordic Treebank Network is headed by Joakim Nivre. See www.masda.vxu.se/~nivre/research/nt.html
2Annotate has been developed at the University of Saarbrücken. See www.coli.uni-sb.de/sfb378/negra-corpus/annotate.html
Using a flat tree structure for manual treebank annotation has two big advantages for the human annotator:

1. the annotator needs to make fewer decisions, and
2. the annotator has a better overview of the trees.

This comes at the prize of the trees not being complete from a linguistic point of view. One could ask why an NP that consists of only one daughter is not marked, or why an NP that is part of a PP is not marked, while the same NP outside a PP is explicitly annotated. These restrictions also have practical consequences: If certain phrases (e.g. NPs within PPs) are not explicitly marked, then they can only indirectly be searched for in corpus linguistics studies.

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) so that we get fine-grained correspondences between languages. Our alignment focuses on meaning, rather than sentence structure. This means that sentences can have alignment on a higher level of the tree (for instance if the S-node carries the same meaning in both languages), without necessarily having alignment on lower levels (for instance an NP without correspondence). 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.

2 Building treebanks with automatic node insertion

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 then automatically deepened the flat syntax trees. This was achieved by a Perl-program, which automatically inserts nodes to create the deeper structure. However, these insertions must be totally un-ambiguous, so that no errors are introduced.

2.1 The node insertion program

The input for this program is a tree description in TIGER-XML [König and Lezius 2002], an interface format which can be created and used by the treebank tool TIGERSearch\(^3\). The output is a deepened TIGER-XML tree. Our deepening program can be called with or without the \_i-flag. When the flag is used, the program

\(^3\)See also www.ims.uni-stuttgart.de/projekte/TIGER/TIGERSearch/doc/html/TigerXML.html.
adds the marker _i to every automatically inserted node, enabling checks of the insertions in a tool like TIGERSearch. Figure 1 shows an example tree before and after the automatic insertion of an adjective phrase node and a noun phrase node.

There are basically two sets of rules; rules for the insertion of unary nodes and rules for handling other nodes. The first set of rules insert adjective phrases (APs), adverbial phrases (AVPs), noun phrases (NPs) and verb phrases (VPs). One simple rule is the one which inserts an AP if there is an NP with a direct adjective child (ADJA). More complex rules are e.g. the rules for handling NPs. The main point is that an NP that is the child of a PP should be marked in the same way as any other NP. These rules are listed in table 1.

We have three rules for unary branching nodes, which state that

1. If we have an NP with an AG (genitive attribute), APP (apposition) or GL (prenominal genitive) child, then this child is annotated as NP.

2. If there is an S or CNP (coordinated nominal phrase) with a direct MPN (multi-word proper noun) child, then this child is annotated as NP.

3. If there is an S, VP or CNP with a nominal (noun, pronoun or the like) child, then this child is annotated as NP.

Another rule states that unless there is only an AP, AVP or CNP together with
Unary branching nodes:

The format of the rules: ‘flat structure’ $\implies$ ‘deep structure’, with nodes in parentheses, edge labels in brackets; X standing for any label.

if NP with a direct genitive or apposition child, e.g. *Sofies Mutter*, then insert an NP:

$$(NP) \rightarrow [AG|APP|GL] \rightarrow NE \implies$$

$$(NP) \rightarrow [AG|APP|GL] \rightarrow (NP) \rightarrow [HD] \rightarrow NE$$

if S or CNP with a direct MPN child, then insert an NP:

$$(S|CNP) \rightarrow [X] \rightarrow MPN \implies$$

$$(S|CNP) \rightarrow [X] \rightarrow (NP) \rightarrow [HD] \rightarrow MPN$$

if S or VP or CNP with a direct noun or pronoun child, then insert an NP:

$$(S|VP|CNP) \rightarrow [X] \rightarrow$$

$$(NN|NE|PPER|PDS|PRF|PPOSS|PIS|PRELS|PWS|TRUNC) \implies$$

$$(S|VP|CNP) \rightarrow [X] \rightarrow (NP) \rightarrow [HD] \rightarrow$$

$$(NN|NE|PPER|PDS|PRF|PPOSS|PIS|PRELS|PWS|TRUNC)$$

Other nodes:

All children of a PP except for the preposition are marked as NP, unless there is only an AP (e.g. *seit längerem*) or an AVP (e.g. *bis morgen*) or a CNP in the NP.

A coordinated noun phrase (CNP) does not get an NP mother if it is the child of an S or a PP. But if the CNP has siblings (typically modifiers) that belong to the same NP, then it gets an NP mother.

| Table 1: Rules for insertion of NP nodes |
the preposition in a PP, everything but the preposition should be made into an NP. This binds the parts of an NP inside the PP together (like in figure 1). Finally the program checks that every NP has a head (this is especially important since the automatic alignment program created for [Samuelsson 2004] is based on the fact that every node has a head).

The node insertion rules should all be reliable because they are un-ambiguous, as long as the manual part of the annotation is correct according to the guidelines. However, there are some problems with the program. Cardinal numbers are still not handled since they can be of different types, e.g. adjective-like in 25 Computer and noun-like in im Jahre 2000. Therefore they are not un-ambiguous and cannot easily be handled automatically. Adverbs in adjective phrases are still not handled (they should have their own AVP) and there are several STTS-tags that do not have their own node label, e.g. PTKNEG for the negation particle. Several of them could probably be made into adverbial phrases (AVPs).

2.2 Creating the German treebank

The main gain of the program is of course to speed up the work in creating treebanks. This worked very well for the German trees. For the 225 German sentences in the first chapter of Sofies Welt, with 3146 tokens, we thus semi-automatically annotated 1426\(^4\) nodes with 4570 edges. The automatic node insertion resulted in a total of 2278 nodes with 5422 edges. This means an increase of almost 60% with regards to the nodes. 548 of the inserted nodes were NPs (of which 420 are unary), 143 were APs (all unary), 160 AVPs (all unary) and 1 VP. Within PPs 189 NPs were inserted and 45 APs.

The semi-manual annotation of the flat structured German sentences took about 5 hours. This means almost 5 nodes per minute, which in turn would mean that we saved almost 3 hours of annotation time due to the automatic node insertion. This is not entirely true, since not all nodes are equally problematic. An annotator needs more time to make a decision for a problematic node and many of the nodes that are automatically inserted are easy to create during semi-manual annotation. But it gives a hint about the possible time gain in the annotation process, when creating large treebanks.

2.3 Creating the Swedish treebank

When aiming for a parallel treebank it is advantageous to handle the annotation of the Swedish sentences similar to the annotation of the German sentences. Tra-

\(^4\)These numbers are taken from TIGERSearch, which also includes a top node for each sentence, with edges to the punctuation.
ditionally, however, a different PoS tagset has been used for Swedish, called the SUC-tagset\(^5\). We trained Brants’ TnT-tagger with the SUC-tagset for automatic PoS tagging. Unfortunately there is no constituent structure treebank for Swedish that could be used for training a chunker with resulting structures corresponding to the German sentences. Therefore we mapped the SUC-tags into the German STTS to be able to re-use the German chunker in Annotate for Swedish. This works nicely. A small experiment, where the children were manually selected, shows that the German chunker suggests 89% correct node labels and 93% correct edge labels for Swedish ([Volk and Samuelsson 2004] and [Samuelsson 2004]).

Still Swedish annotation takes more time than German annotation (over 10 hours), mainly due to the fact that the NEGRA annotation guidelines are written for German and there is no appropriate Swedish annotation manual. We had to adapt the guidelines to Swedish as we went along. Even though the Annotate tool mostly suggests the correct node and edge labels, there are still a number of difficult cases for the annotator to decide.

One example is the difference in Swedish between prepositions and verb particles. In spoken language, prepositions are not stressed, while verb particles are. In some cases the word is a preposition, not a verb particle, but it still is closer to the verb and therefore behaves “strangely”. One example is the relative clause

\begin{equation}
\text{hår som varken gelé eller spray bet på}
\end{equation}

(\textit{hair which neither gel nor spray would work on})

In this relative clause the pronoun som is put in the beginning and is thus separated from the stranded preposition på. The annotator has to establish that this is indeed a preposition and then decide whether the pronoun should be in the prepositional phrase (with crossing branches) or not.

After annotation, the Swedish trees are automatically deepened in the same manner as the German trees. Since Swedish and German are similar languages, there are only minor differences between the insertion programs. For instance, pre-noun genitives in German are always assumed to be proper names (NE) but in Swedish they could also be regular nouns (NN) (e.g. \textit{vid världens ände} (literally ‘at the world’s end’)).

One difference between Swedish and German is that a PP in Swedish can consist of a preposition and a sentence or verb phrase. A phrase like

\begin{equation}
göra min av att svara
\end{equation}

(\textit{make as if to answer, literally ‘make an expression of answering’})

\(^5\)SUC, the Stockholm-Umeå Corpus, is a 1 million word representative Swedish corpus which is annotated with Part-of-Speech tags, morphological tags, lemmas and name classes. All of SUC is manually checked.
should contain this structure: [PP av[VP att svara]]. This means that we have to add the possibility of having an S or VP in the PP for Swedish.

Some problems with the node insertion for Swedish actually resulted from errors in the mapping from STTS-tags back into SUC-tags. For instance, the STTS-tag KOKOM (comparison particle, without sentence) is translated into PR (preposition). According to the German annotation guidelines the KOKOM is part of the NP, while of course a preposition should have a PP mother node. Since the translation of the STTS-tags back into SUC-tags is done after the node insertion, this creates erroneous tree structures. A look through the SUC-database showed that the correct translation of KOKOM should have been KN (conjunction).

We also experimented with a rule-based Swedish chunker, to pre-process the sentences before loading them into Annotate for the semi-manual parsing. The chunker produces deep structures for NPs and PPs, which makes part of our later deepening obsolete. But it turned out that this type of deep pre-chunking thwarts the advantages of the flat tree annotation. In fact it leads to more decisions and worse overview for the human annotator. This is mostly due to the fact that our chunker builds all the nodes it can find, which gives trees of “uneven depth”. The solution would be a pre-chunker, which computes only flat and “safe” structures so that the human annotator can concentrate on deciding ambiguous attachments.

3 How flat can a tree be?

The successful automatic node insertion gave rise to the question of whether the flat annotation according to the NEGRA guidelines could be made even flatter. In other words, is it possible and feasible to define a minimal set of human annotation decisions with a maximum number of automatically inserted nodes and labels? A minimal form for the manual annotation, where the rest of the nodes are automatically inserted later, could save a lot of time. The deepening of course still has to be totally safe, i.e. un-ambiguous.

The problem is defining a minimal form that is still linguistically plausible. If the form is too minimalistic and skeletal, it will be hard for the human annotator to still maintain an overview and to see what is correct in the annotation. The advantages of the flat annotation should not be renounced.

It is rather complex to determine the nodes that are always un-ambiguous. There are mainly two difficulties. The first one is that a node has to be manually inserted (i.e. a manual decision has to be made) if any of its edges are ambiguous. This means for instance that we cannot automatically insert a missing top node S (sentence), since the distinction between subject and object is often ambiguous (for the computer).
The second difficulty in finding the un-ambiguous nodes lies in language differences. One example of this is the node VZ (zu-marked infinitive) which in German un-ambiguously groups the infinitive marker immediately followed by an infinitive verb. But the Swedish equivalent (marked with att) is ambiguous. For instance the marker and the verb do not have to be adjacent, there might be an intermediary adverb. Also the word att might be something other than an infinitive marker.

One example of nodes that could be left out in the manual annotation for both German and Swedish (to be automatically inserted afterwards) is nested VPs in verb chains. According to the NEGRA guidelines we need nested VPs for every verb in a chain (see figure 2). But it would facilitate the manual annotation if all verbs in such a chain could be entered into one VP (i.e. as sister nodes on the same level) in the flat tree structure, and the nesting could be done automatically afterwards. This nesting is unambiguous in both German and Swedish since the order of the verbs within the verb group is fixed (with few exceptions like the German Oberfeldumstellung which need to be handled manually).
4 Conclusions

In creating a parallel treebank we investigated how much of the manual labour involved can be carried out automatically. We built flat phrase structure trees according to the NEGRA guidelines. Then we had a program insert un-ambiguous nodes to get a deeper and more detailed structure. This insertion step rendered about 60% more nodes.

Our ultimate goal is to make the tree structure even flatter. A minimal tree is a tree where all and only the ambiguous attachments have been decided by the human annotator. We found that some more nodes could be automatically inserted (in addition to the ones that are already left out according to the NEGRA guidelines). This means that creating treebanks in the future can be made into a less time consuming task.

References


