MAMBA Meets TIGER: Reconstructing a Treebank from Antiquity

Jens Nilsson, Johan Hall and Joakim Nivre
Växjö University
The School of Mathematics and Systems Engineering, MSI
{jni, jha, nivre}@msi.vxu.se

Abstract

In this paper, we will give an overview of the reconstruction process of the Swedish treebank Talbanken, created in the first half of the 70's. Talbanken contains both written and spoken material, both encoded in the MAMBA-format. The goal has been to construct two new versions of the original data, one based on phrase structure and one on dependency structure. The outcome of the reconstruction, i.e. different versions of Talbanken, is available for non-commercial research and educational purposes.

1 Introduction

Treebanks, collections of syntactically annotated sentences, are important resources for data-driven parsers. They have a number of advantages over rule-based parsing techniques, such as fast development time, broad-coverage and robustness. When developing a parser for Swedish one needs a treebank containing Swedish sentences, but currently there is a lack of Swedish treebanks of substantial size. This holds for the other Nordic languages too, with Danish as an exception. The absence of Swedish treebanks is remarkable considering that two corpora of Swedish text augmented with syntactic annotation have been created, one as early as 1974 named Talbanken (Einarsson 1976a; Einarsson 1976b), and another in the 80's named Syntag (Järborg 1986). Unfortunately, the annotation formats of these resources make them cumbersome to use for modern treebank tools and parsers. In a way, Sweden can be regarded as a pioneer in this area, but thereafter the work with creating new treebanks has decreased considerably. As long as there does not exist a large enough Swedish treebank, one can reuse the available resources.

The aim of the project that this report presents is to recycle, or reconstruct, Talbanken to a more convenient format adapted to modern treebank tools and parsers. The motivation for this is primary to make an unwieldy linguistic resource available for research purposes. Talbanken is manually annotated with partial phrase structure and grammatical functions encoded in the MAMBA-format (Teleman 1974), and was a very impressive achievement at the time of its creation. We will in this project raise the question if it is possible to transform Talbanken's original annotation into an encoding format that can be used to extract both dependency trees and phrase structure trees, and if it is possible to leave out certain kinds of information that instead can be acquired by applying transformation rules (i.e. inference). An overview of this idea is shown in figure 1.

The encoding format we use to convey the phrase structure and dependency structure is TIGER-XML (Lezius et al. 2002). It is one attempt to create a more generic format for representing various corpora and treebanks. The format is designed to be theory-independent, but it is especially suited for treebanks using phrase structure annotation schema, where the encoding uses node labels (syntactic categories) and edge labels (grammatical functions) for creating the syntactical structure. The Nordic Treebank Network has agreed to use the TIGER-XML format to exchange syntactic information within the network, and for this purpose there is a special annotation in TIGER-XML for dependency treebanks (Kromann 2005). TIGER-XML documents can easily be imported into the search tool TIGERSearch (König et al. 2003), which is a treebank viewer with a graphical user interface.

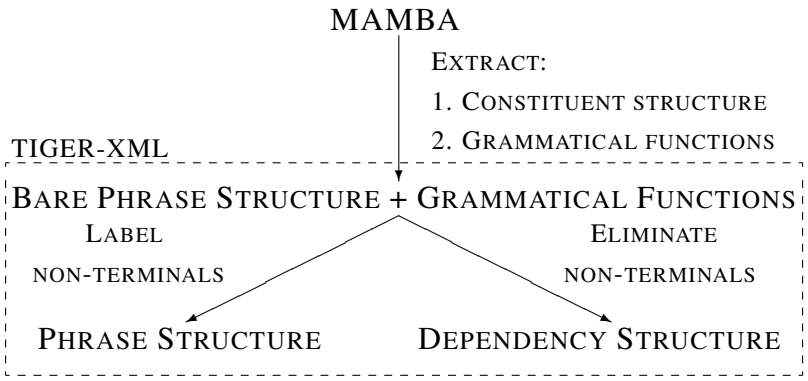

Figure 1: Reconstructing Talbanken

2 Reconstructing Talbanken

This section will deal with the first part of the project, that is, we take the syntactically annotated data in the original format and transform it into a bare phrase structure, mentioned in figure 1. Our definition of a bare phrase structure is a syntactic annotation based on constituency with unspecified phrase labels.

2.1 Talbanken

This section will give a brief presentation of Talbanken. It is divided into four parts, which together comprise close to 320,000 words. The four parts are: *Professionell prosa* (Professional prose, about 85,000 words), *Gymnasistsvenska* (Swedish by high school students, about 85,000 words), *Samtal och debatt* (Conversation and debate, about 75,000 words), and *Boråsintervjuerna* (The Borås interviews, about 75,000 words).

The first two are written language and the last two are spoken language, all syntactically annotated. Since the part ”Professionell prosa” was the major aim of the project, the reconstruction of the three other parts can be regarded as an additional bonus. Only a few adjustments were needed in the reconstruction program in order to take care of the peculiarities in the three other parts. Especially the two parts containing spoken language required a few minor changes in the program.

```
P11126050001     0000                 <<       GM          046
P11126050002     *MAN                 POZPHH   SS          046
P11126050003     FÄSTER               VVPS     FV          046
P11126050004     STÖRRE               AJKP     OOAT        046
P11126050005     VIKT                 NN       OO          046
P11126050006     VID                  PR       OAPR        046
P11126050007     ELEVERNAS            NNDDHHGGOADT         046
P11126050008     SPONTANA             AJ       OAAT        046
P11126050009     FÖRMÅGA              NN       OA          046
P11126050010     1000                 IF       OAET        046
P1112605001110002ATT                  IM       IM          046
P1112605001210002UTTRYCKA             VVIV     IV          046
P1112605001310002SIG                  POXPHHGGOO           046
P1112605001410002MUNTLIGT             AJ       AA          046
P1112605001510002OCH                  ++OC     ++          046
P1112605001610002SKRIFTLIGT           AJ       AA          046
P11126050017     .                    IP       IP          046
-------------------------------------------------------------------
1   2 3  4  5    6                    7        8           9
```

Figure 2: An example in the MAMBA-format: ”One gives greater weight to the-pupils’ spontaneous ability to express themselves orally and in writing.”

2.2 The MAMBA-format

From a modern point of view, the MAMBA-format may seem a little obscure and odd. But it should be kept in mind that at the time of its creation, Talbanken was originally placed on punch cards having a limited upper line length. This does of course convey limitations on the encoding format. Talbanken was later transformed into a more convenient electronic form, but the limitations remained.

Figure 2 shows a sentence from Talbanken. It consists of two layers: one is a lexical analysis layer, comprising part-of-speech information and morphological features (field 7), and the other is a syntactic analysis layer having grammatical functions (field 8). Field 1-4 and 9 contain non-grammatical data such as id for the current text (1), paragraph (2), macro syntagm (3), word (4) and graphical sentence (9). All fields except the one containing the word-id are associated with one or more so called non-grammatical dummy words, which are used to mark the beginning of a new text, paragraph, macro syntag or graphical sentence. The first line in the figure is a non-grammatical dummy word marking the beginning of a new graphical sentence (*GM*).

An important thing to note here is that there is a notion of phrases, but the phrase labels are unspecified. We can for instance see that *större vikt* is the

object at the lowest level, since both words have the grammatical function OO in the first slot of field 8. This phrase would in a modern treebank have the label NP, but this kind of information does not exist in Talbanken. On the other hand, the grammatical functions marking phrase boundaries, such as object (OO), subject (SS) and adverbial (OA), resemble the arc labels normally found in dependency structure.

Field 8 in the MAMBA-format has a very limited number of slots. Each word can contain at most six grammatical functions. Therefore, the MAMBA-format offers the possibility to introduce additional syntactic hierarchy using hierarchic dummy words. Line 10 with the word form *1000* (field 6) is a hierarchic dummy word and is not part of the original sentence. The idea here is that all lines in the rest of the sentence having the same digit at the first position in field 5 as the first digit in the word form of hierarchic dummy word (i.e. the digit *1*) are affected by this hierarchic dummy word. This is the case for the words from line 11 (*att*) to line 16 (*skriftligt*). This means that the grammatical functions of the dummy word (*OA ET*) really should be placed in front of the grammatical functions of all the words at line 11 to 16. For example, the word *att*, which has one grammatical function (*IM*), will have the grammatical functions *OA ET IM* after the expansion of the hierarchic dummy word.

If necessary, hierarchic dummy words can also be nested. A dummy word can be placed inside another dummy word in order to introduce additional hierarchy. The depth can be no deeper than four, since field 5 admits no more than four digits (each digit corresponds to a specific level of nesting). See Teleman (1974) for more details concerning the MAMBA-format. There you will also find detailed descriptions about the lexical and grammatical categories of the MAMBA-format.

One interesting feature of the MAMBA-format is that it allows discontinuous construction. Words having the same grammatical function at a specific level can be interrupted by other grammatical functions, which the reconstruction program can handle. Related to discontinuity in the MAMBA-format is what is known as alphabetical displacement. The MAMBA-format uses alphabetical displacement in order to distinguish two phrases from each other, with the same grammatical function within the same phrase.

2.3 From the MAMBA-format to bare phrase structure

Here we will explain the process of creating the bare phrase structure version. In principle, the reconstruction consists of four phases. First, the input module takes a file containing texts encoded in the MAMBA-format, and extracts the necessary information described above, line by line. It returns a list

of MAMBA-lines. Secondly, another module takes such a list and categorizes each item as either (1) a non-grammatical dummy word, (2) a hierarchic dummy word, or (3) an actual word (first two mentioned above). The third module of the reconstruction program tries to identify phrases in the syntactic hierarchy captured by the grammatical functions, leaving the node labels of the phrases unspecified. Finally, we refine the output from phase three, since it is not well-suited for later transformations. The identification of phrases is the core of the reconstruction program. An important step before this can be done is to divide the data into sentences. The non-grammatical dummy words and their corresponding fields in the MAMBA-format have been used to divide the corpus into sentences, which proved to be a quite complex task to sort out. Thereafter, all non-grammatical dummy words are removed from the data, since they are no longer needed.

The next step in the process is to expand all hierarchic dummy words. In other words, we insert the grammatical functions of the hierarchic dummy words in front of the grammatical functions of the words they affect, according to the description in section 2.2. Figure 2 above contained one hierarchic dummy word, and its grammatical functions (*OA ET*) have been extracted and inserted in front of the words from *att* to *skriftligt*. Thereafter, the hierarchic dummy word is redundant and is consequently removed.

After this expansion, the module for identifying phrases takes over. In principle, the task for this module is quite straightforward. The core of this module is a recursive method that takes two parameters: a list of MAMBA-words with the expanded grammatical functions and the current analysis depth. When the words are analyzed and the inner hierarchy is identified, the function returns a non-terminal node. The node has a right-hand side of non-terminals and terminals comprising the identified hierarchy of expanded MAMBA-words. The current analysis depth is the same as the slot index in the list of grammatical functions.

This function is called once for each sentence, and all words are fed to it together with depth 1 (it starts by looking at all the words' left-most grammatical functions). The main principle that is used to identify phrases within a list of words is to make everything with the same grammatical function at the current analysis depth part of the same phrase. When a list of words has been partitioned with respect to the grammatical function, the function evokes itself with the words in each such partition, and adds one to the current analysis depth. The returned non-terminal is then inserted in the right-hand side. All words lacking a grammatical function at the current analysis depth are treated as terminals and added to the right-hand side. Since they do not have a grammatical function, the terminals are given the unspecified edge label *??*. They will be replaced by more informative edge labels during

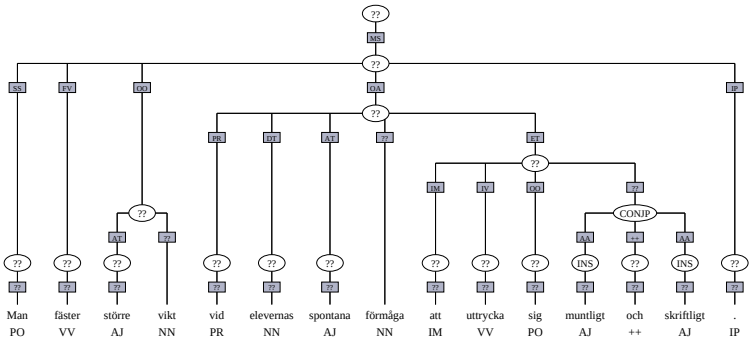

Figure 3: The bare phrase structure tree of the sentence in figure 2.

the refinement phase.

To make things more concrete, have a look at the sentence in figure 2. When the identification of terminals and non-terminals is done, the complete bare phrase structure tree representing the sentence is depicted in figure 3.

The sentence also contains coordination which is not treated in the same way as the above described phrase identification. Without going into details, the module for handling coordination starts by identifying all the conjunctions in the sentence. All conjunctions have the grammatical function ++. Step two is to identify the conjuncts, which somewhat simplified is everything with the same grammatical function on both sides of the conjunction at the same level as ++. In the sentence, this is the grammatical function *AA* for the words *muntligt* and *skriftligt*. We have chosen to capture coordination in a new non-terminal that we temporary call *CONJP*.

The output of the conversion so far contains a lot of peculiarity and unnecessary constructions inherited from the MAMBA-format. Therefore, we want to refine the output into a more appropriate phrase structure for later transformations. The two most important transformations that we perform here are: (a) removing unary non-terminals, and (b) finding heads and modifiers. We have chosen to remove all non-terminals having just one child. For example, the terminal node *Man* in figure 3 is the only child of its parent node. This non-terminal is discarded and the label above it (*SS*) is instead assigned to the terminal node *Man*, which replaces the non-terminal in the tree. When all such unnecessary non-terminals are removed, the module for finding heads and modifiers takes over. In principle, all edge labels in the tree that still have the unspecified name *??* are heads, and these labels are set

to the name *HD*. This is the case for the words *vikt* and *förmåga* in figure 3. An example of a sentence after the refinement is shown in figure 4 (disregard the node labels, which are assigned below).

3 Labeling bare phrase structure trees

The goal of this step is to transform the bare phrase structure trees encoded in TIGER-XML into phrase structure trees with suitable phrase labels. We have chosen to use a small set of phrase labels. The labeling process is performed by traversing the trees recursively in a bottom-up fashion. For each non-terminal (except coordination nodes, which is handled as a special case described later on) we collect information to apply a labeling rule, which is a quadruple (C, P, L, N), where:

1. C and P are lists of grammatical functions,
2. L is a list of lexical categories,
3. N is a non-terminal node label.

A labeling rule (C, P, L, N) assigns the label N to a node n if the following conditions are satisfied:

1. n has a child with a grammatical function $g \in C$, or C = *,
2. n has a grammatical function $g \in P$, or P = *,
3. n has a child with a lexical category $l \in L$, or L = *.

In table 1 we show the set of rules we use to label the phrases, ordered by decreasing priority. After looking up the appropriate phrase label, the non-terminal is labeled with this label. In figure 4, the phrase node *S* gets its label because there is a finite verb labeled *FV* amongst the children, as well as a subject *SS* (rule 3). Moreover, the modifier *AT*, *individuell*, modifies the head word *beskattning* and this is only found in a noun phrase *NP* (rule 7). The edge label *PR* as a child in a phrase is a strong indicator that it is a prepositional phrase *PP*, and can be found in two places in the example: *Genom skattereformen* and *av arbetsinkomster* (rule 2).

The infinitive marker *att* labeled *IM* and the non-finite *acceptera* labeled *IV* are evidences of a verb phrase *VP* (rule 5). Such an example is shown in figure 5. In the same example, we can also see a clause fragment with the parent edge label *+F* which indicates that some component is missing, but it is nevertheless labeled *S* according to the present set of labels (rule 4).

#	C	P	L	N
1	MS	*	*	ROOT
2	PR	*	*	PP
3	SS, FV	*	*	S
4	*	+F	*	S
5	IV, IM	*	*	VP
6	*	VS, VO	*	VP
7	DT, AT, ET	*	*	NP
8	HD	DT	PN, MN, AN, VN, NN, PO, RO	NP
9	HD	DT, PR	*	XP
10	HD	*	PN, MN, AN, VN, NN, PO, RO	NP
11	HD	SS, OO	*	NP
12	HD	*	AJ, TP, SP	AP
13	HD	*	AB	AVP
14	*	SS, OO	*	NP
15	*	*	*	XP

Table 1: A list of labeling rules. The C-column (Child) enumerates the edge labels from the non-terminal to its children and the P-column (Parent) enumerates the edge labels from the non-terminal to its parent. The L-column (Lexical categories) contains the parts-of-speech for the children to the non-terminal, and the N-column (Non-terminal) contains the resulting phrase label.

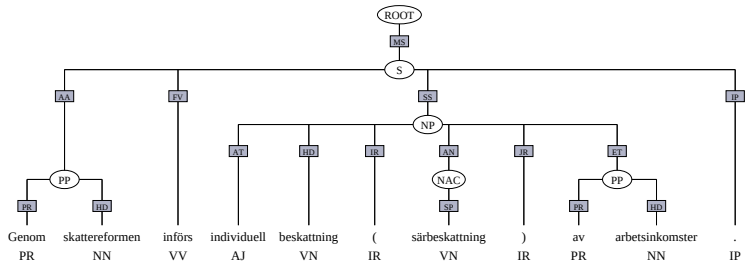

Figure 4: After the labeling of the phrases: ”By the tax reform individual taxation (separate taxation) of work income is introduced.”

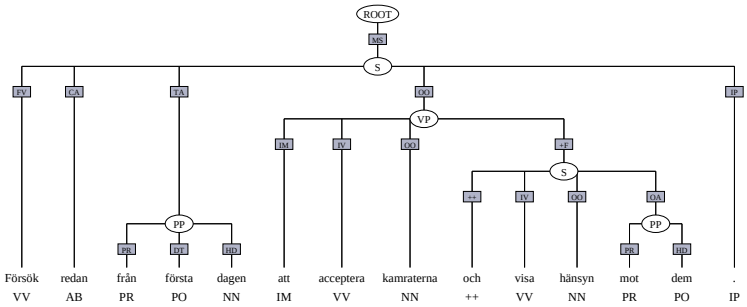

Figure 5: After the labeling of the phrases: "Try already from day one to accept the fiends and show consideration for them."

We introduce a phrase labels *XP* (rule 9), which can be viewed as a phrase label containing all phrases we do not want to discern. An example where we used this dummy phrase *XP* is what one can call determiner phrases and another example is multiword unit.

Finally, there is a special case for labeling the coordination phrases. A special list of labeling rules is used to determine the phrase label for coordination. A labeling rule for coordination is a triple (L, P, N), where:

1. L is a list of lexical categories,
2. P is a list of non-terminal labels,
3. N is a non-terminal node label.

A labeling rule (L, P, N) assigns the label N to a node n if the conjuncts have the lexical category $l \in L$ or the non-terminal label $p \in P$. If there exist different conjuncts which satisfy two or more labeling rules the coordination phrase will be labeled with the default coordination label *CONJP*.

In the example shown in figure 6 the phrase label *CNP* is used because the conjuncts' lexical categories are *NN* and *VN* (both are mapped to *CNP*). The phrase label *CONJP* is used to indicate that there are two suitable phrase labels, *CPP* and *CAVP*.

4 Extracting dependency trees

In the transformation from bare phrase structure to phrase structure, the reconstruction program was forced to augment nodes with node labels. The

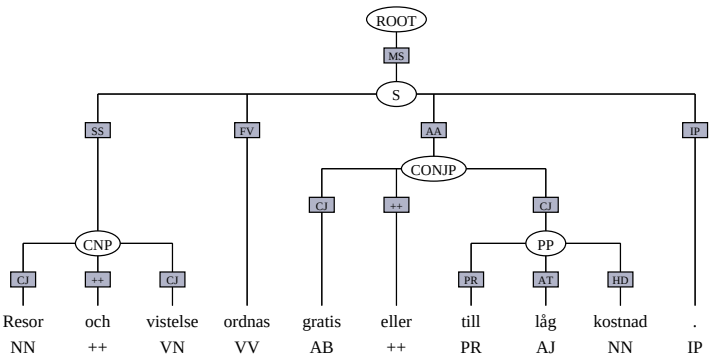

Figure 6: After the labeling of the phrases: ”Travels and stays are arranged for free or at a low cost.”

transformation to dependency grammar is on the other hand not a matter of adding information. Instead, it performs a selection among the available and perhaps ambiguous information in order to construct dependency trees.

As input to this step, we use the output from the first step described in the section 2, i.e. bare phrase structure trees. One might consider using the labeled phrase structure as input to this step, since the non-terminal labels, such as *NP* and *VP*, can be useful in the construction of the dependency structure. We have decided to use only the original information instead, since the non-terminal labels are extracted from the same information source.

The transformation going from the source into dependency annotation relies on a number of premises, and much of the ideas are inspired by the technique of head-finding rules in Collins (1996) and Magerman (1995). The idea is that each phrase contains one head word, and consequently, all other words in that phrase will be non-head words. Collins' goal was also to extract dependency trees from the Penn Treebank by using head-finding rules, and the problem with the Penn Treebank (and treebanks based on constituency in general) is that it does not specify which word in the subtree of each phrase is the head. Collins uses a list of priority rules by searching among the children of each phrase to find an appropriate head. The selected head depends on the phrase label of the parent and the phrase label and parts-of-speech of the children.

We apply an algorithm that resembles the one described above in the sense that we are trying to find one head for each phrase. However, the

information we have is different from the information in Penn, since the information source in this step is the edge labels. In the original MAMBA-annotation, phrase heads were identified by leaving the grammatical function unspecified. These correspond to the words in the phrase structure having the edge label *HD*. In the case where a phrase contains exactly one such head word, the task of choosing the head of the phrase is trivial. Unfortunately, not all phrases have this nice property. It can be the case that a phrase lacks such a word, which is most notable at the main clause level. A phrase can also have more than one head. The problem here is to choose a tenable solution for the two non-trivial cases.

First, the algorithm traverses each tree bottom-up and with the current design only takes the children's edge labels for each phrase into consideration. Compared to Collins, the transformation makes use of head-finding rules, but it uses only the edge label of the children and disregards the information given by the parent node, such as phrase label and edge label. Table 2 lists the head-finding rules.

EDGE LABEL
Head (HD)
Finite verb (FV)
Non-finite verb (IV)
Predicative complement (SP)
Subjects and objects (SS, ES, FS, VS, OO, EO, FO, IO, VO)
Clause adverbials (AA, KA, RA, OA, TA)
Phrase adverbials (+A, CA, MA, NA, VA, XA)
Nominal pre-modifier (AT)
Nominal post-modifier (ET)
Other noun dependents (DT, XT)
Unclassifiable dependent (XX)
Other functions not involved in coordination (e.g. +F, EF, XF, IM, AN)
Conjunct (CJ)
Coordinator, conjunction (++)
Punctuation (I?, IC, IG, IK, IP, IQ, IR, IS, IT, IU, JC, JG, JR, ST)

Table 2: The priority list of head-finding rules

The list of head-finding rules is also a priority list, where the first rule of the list has the highest priority and the last rule the lowest. In principle, for each phrase, the algorithm looks at the children's edge labels. Then one of two things happens:

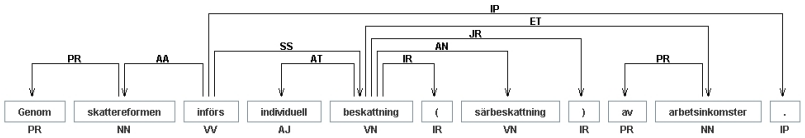

Figure 7: The dependency tree of the sentence in figure 4

- If it finds exactly one edge label with higher priority than all the other, it makes the child with that edge label the head of the phrase.
- If it finds two or more edge labels with the same priority and with higher priority than all the other, then it lets the left-most of these children be the head of the phrase.

When a head has been identified, the second task for the algorithm is to make all other words dependents to the head. In the simple case, when the head is an individual word (terminal node), that word will attach to all its children. In the other case, when the head in a phrase (non-terminal node), the word that was identified as the head of that phrase will attach to all the children.

The corresponding procedure holds for the children too. If a child is an individual word, then that word is the dependent of the head. For example, this is the case for the dependent *Genom* and the head *skattereformen* in figure 4. Also, the dependency relation between the head word and the dependent is set to the edge label of the child, which here will be *PR*. If the child on the other hand is a phrase, the head word of that phrase is the dependent to the head. This is the kind of relation between the words *skattereformen* and *införs* in the same figure. The dependency relation between them is *AA*, since that is the edge label of the phrase that is dependent on *införs*. Figure 7 depicts the corresponding dependency trees. Note also that it is always the highest positioned edge label that is used to label the dependency relation, which becomes apparent in the relation between *beskattning* and the apposition *särbeskattning*. Here the edge label *AN* is picked, not the lower positioned *SP*.

Among the list of edge labels, it is important to note that *CJ* has precedence over ++, and a consequence of the algorithm described above is that the left-most child with the edge label *CJ* will be the head of all other children in all conjunction phrases, similar to Mel'cuk (1988). Since conjunctions in dependency grammar are cumbersome (as well as for other theories), this solution can be regarded as good as any other solution. We will therefore stick to this for now. An example of coordination in a dependency tree given

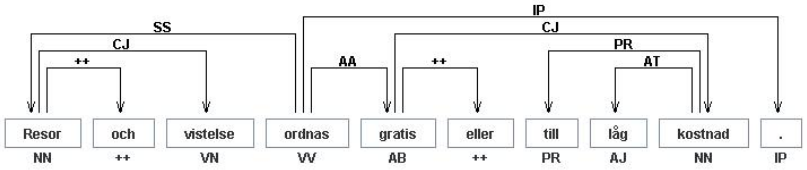

Figure 8: The dependency tree of the sentence in figure 6

this solution is shown in figure 8, having figure 6 as input (disregarding the node labels).

5 Conclusions

In this paper, we have discussed the reconstruction of the Swedish treebank Talbanken, encoded in the MAMBA-format. At a reasonable cost, it is now more accessible after the transformation into a modern representation, i.e. the new encoding format TIGER-XML. The most important result is that we got two new versions of Talbanken, one based on phrase structure and the other on dependency structure. For the phrase structure version, we have created a set of labeling rules in order to infer the phrase labels, since the original format contained a notion of phrases, but no actual phrase labels. On the other hand, for the dependency version, we have instead applied rules for finding heads and removing the non-terminals. Since not all phrases in the bare phrase structure have exactly one head child (edge label *HD*), the problem is not completely trivial. We adopted the strategy to use a priority list and choose the left-most child in case two edge labels have the same priority.

The main focus of this project has been to reconstruct the part ”Professionell prosa”. However, since the spoken parts are encoded in more of less the same way, we could reuse the reconstruction program with only a few adjustments. Most notably, the spoken parts contain no graphical sentences and macro syntagms, which occur only in written language. They are instead divided into utterances, which are marked with dummy words having the grammatical function $<<$, and are not present in the written parts. To keep the division simple, we chose to treat these dummy words in much the same way as *GM*. Thus, the two parts containing spoken language are by and large divided into utterances according to the original MAMBA-annotation.

At *http://w3.msi.vxu.se/users/nivre/research/talbanken.html*, the converted versions of Talbanken can be found. They are freely available for non-

commercial research and educational purposes. A more detailed description of Talbanken, the MAMBA-format, and the reconstruction process is found in Nilsson and Hall (2005).

References

Collins, M. (1996). A new statistical parser based on bigram lexical dependencies. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 184–191.

Einarsson, J. (1976a). *Talbankens skriftspråkskonkordans*. Lund University, Department of Scandinavian Languages.

Einarsson, J. (1976b). *Talbankens talspråkskonkordans*. Lund University, Department of Scandinavian Languages.

Järborg, J. (1986). Manual för syntaggning. Technical report, Göteborgs universitet: Institutionen för svenska språket (Språkdata).

Kromann, M. T. (2005, March). Proposals for extensions and conventions in Tiger-XML within the nordic treebank network. http://www.id.cbs.dk/˜mtk/ntn/tiger-xml.html.

König, E., W. Lezius, and H. Voormann (2003). TIGERSearch user's manual. Technical report, University of Stuttgart, IMS.

Lezius, W., H. Biesinger, and C. Gerstenberger (2002). TIGER-XML quick reference guide. Technical report, University of Stuttgart, IMS.

Magerman, D. M. (1995). Statistical decision-tree models for parsing. In *Proceedings of the 33rd Annual Meeting of the Association for Computational Linguistics*, pp. 276–283.

Mel'cuk, I. (1988). *Dependency Syntax: Theory and Practice*. State University of New York Press.

Nilsson, J. and J. Hall (2005). Reconstruction of the Swedish Treebank Talbanken. Technical Report MSI-05067, Växjö University: School of Mathematics and Systems Engineering.

Teleman, U. (1974). *Manual för grammatisk beskrivning av talad och skriven svenska*. Lund: Studentlitteratur.