Annotated Clauses and Flat Phrase Structures for Swedish

Johnny Bigert, Ola Knutsson, Viggo Kann and **Jonas Sjöbergh**
Numerical Analysis and Computer Science
Royal Institute of Technology, Sweden
{johnny, knutsson, viggo, jsh}@nada.kth.se

1 Introduction

In this paper we focus on tools for grammatical analysis, corpus development and how the corpus must be designed to be useful in machine learning and automatic evaluation.

2 What We Have

At the Department of numerical analysis and computer science, KTH, research in language engineering has been conducted for more than ten years. The research started with the development of a fast spell checker for Swedish, Stava (Domeij et al., 1994) and computer support for writing (Cedergren and Severinson-Eklundh, 1992). Today, the research group has grown and involves about ten researchers and consist of several branches, for example spell- and grammar checking, text summarization, information retrieval and an increasing interest and need for general and robust methods for grammatical analysis. Many of the applications developed have their origin in the grammar checker Granska (Domeij et al., 2000).

The research group is working both with statistical and rule-based methods, and combines them when appropriate. In the field of grammatical analysis the research is currently focused on how to increase the performance of POS tagging and shallow parsing. We are developing a tool that combines different kinds of taggers for increased accuracy (Sjöbergh, 2002). The following five types of taggers have been adopted to Swedish using the SUC corpus: HMM (Brants (2000), Carlberger and Kann (1999)), Maximum entropy (Ratnaparkhi, 1996), Memory based (Daelemans et al., 2001), Tree-tagger (Schmid, 1994), Transformation based (Brill, 1992). We have developed a rule-based shallow parser (Knutsson et al., forthcoming) based on Granska. It has been successfully used in an application for statistical context-sensitive spelling error detection, ProbGranska (Bigert and Knutsson, 2002).

3 What We Want

At present, there is no publicly available treebank for Swedish. On the other hand, several chunking tools exist (e.g. Megyesi (2002), Knutsson et al. (forthcoming) which clearly indicate a need for an evaluation resource.

Our main objective is the development of an environment for POS tag and flat phrase structure experiments and automatic evaluation. To this end, we will create an annotated resource for shallow phrase structure based on existing corpora (e.g. SUC). The annotation format will be designed for easy data collection and use with automatic evaluation tools. In essence, the short-term objectives are to:

- Annotate phrases and the corresponding heads (see Section 4.3).
- Annotate clauses (see Section 4.4).
- Implement the corpus format (see Section 4.1).

The long-term objectives are to:

- Annotate the syntactic function of the phrases (see Section 4.3).
- Annotate the nested clauses (see Section 4.4).
- Develop tools for corpus maintenance (see Section 5.3).

4 Annotation

4.1 Format

We plan to use an existing annotation schema (e.g. TEI or XCES) to obtain a layered design separating different levels of analysis. The

annotations will be isolated from the underlying corpus or text by use of indirect reference to the tokens via pointers. This enables us to freely distribute the resources without having to distribute the underlying corpus and thus, we avoid the problems with copyright and contract issues as well as version control.

The aim is to design the corpus to supply a good support for experiments and automatic evaluation. Without such a resource, thorough evaluation is always labour intensive. Small changes to the software often remain without validation. The corpus format will be designed to allow partial and multiple annotations for ill-formed text, e.g. text written by second language learners.

The corpus format will assign each phrase and clause a unique identifier. From this, we can build higher-level analysis annotations, such as phrase structure and dependency trees.

4.2 Extent

We have the ambition to annotate at least 100000 words from SUC (random files). The overall goal is to create sufficient annotations for the evaluation of phrase chunkers and other related tools. To accomplish this, collaboration with other research groups is required since all annotations need to be manually validated.

One interesting aspect is the way the annotation is conducted. Evidently, an automatic annotation followed by a manual validation is biased towards the tool used. More impartial approaches involve an increasing amount of manual work. For example, an interactive use of the same tool would be much more appropriate. Clearly, manual annotation is preferred but involves an unrealistic amount of work.

4.3 Phrase Annotation

We will consider annotating noun phrases, prepositional phrases, verb chains, adverbial phrases and adjective phrases. Annotations of the phrases will contain the following (with examples for NPs):

- Identification of the tokens constituting the beginning and end of the phrase.
- Annotated features of the phrase (e.g. indefinite, definite).
- Fine-grained classification of phrases (e.g. relative, minimal).
- The head of the phrase as a pointer.
- The most suitable minimal NP replacing the phrase (NP).
- Internal structure, such as phrases in phrases.
- Syntactic function for further development of the grammar checker (long-term work).

4.4 Clause Annotation

Annotations of the clauses will contain the following:

- Identification of the beginning and end.
- Identify clause type (e.g. subordinate, complete vs. incomplete).
- Nested clauses (long-term work).

5 Using and Maintaining the Annotated Corpus

5.1 Extraction of Statistics

From the annotated resource, one can obtain the necessary data for the construction of a probabilistic phrase chunker. Furthermore, it may be interesting to extract statistics of the frequency of individual phrase constructions.

5.2 Automatic Evaluation

At the least, we will implement tools for automatic evaluation of the existing rule-based phrase chunker. This will enable us to assess the usefulness of the individual rules and the impact of minor changes. We will also consider evaluating the differences in performance between the rule-based phrase chunker and a probabilistic implementation.

5.3 Corpus Maintenance

Because of the isolation of the corpus annotations and the underlying corpus, we can correct errors without making changes to the original text. Thus, the underlying corpus is static. We accomplish this by using pointers into the underlying corpus or into a table of corrections. Thus, we can distribute the corrections independently of the original corpus. Tools will ensure consistency and will collect data for export purposes.

To detect errors, we can apply existing software such as Stava, Granska and ProbGranska.

The correction may be performed in an interactive mode. We may also detect tagging errors using ProbGranska and combinations of taggers.

6 Ending Remarks

It is becoming increasingly apparent that the research community has developed a need for annotated resources for use with automatic evaluation. Not only the resulting application has a need for evaluation, but also the components of the application. In effect, the ambition of the research community should be to eliminate as much as possible of the need for manual work.

References

J. Bigert and O. Knutsson. 2002. Robust error detection: A hybrid approach combining unsupervised error detection and linguistic knowledge. In *Proc. 2nd Workshop Robust Methods in Analysis of Natural language Data (ROMAND'02), Frascati, Italy*, pages 10–19.

T. Brants. 2000. Tnt – a statistical part-of-speech tagger. In *Proc. 6th Applied NLP Conference, ANLP-2000, April 29 – May 3, 2000, Seattle, WA.*

E. Brill. 1992. A simple rule-based part-of-speech tagger. In *Proceedings of ANLP-92, 3rd Conference on Applied Natural Language Processing*, pages 152–155, Trento, IT.

J. Carlberger and V. Kann. 1999. Implementing an efficient part-of-speech tagger. *Software–Practice and experience*, 29(9):815–832.

M. Cedergren and K. Severinson-Eklundh. 1992. Språkliga datorstöd för skrivande: förutsättningar och behov. Technical report, Department of Numerical Analysis and Computing Science, Royal Institute of Stockholm, Stockholm, Sweden.

W. Daelemans, J. Zavrel, K. van der Sloot, and A. van den Bosch. 2001. Timbl: Tilburg memory-based learner - version 4.0 reference guide.

R. Domeij, J. Hollman, and V. Kann. 1994. Detection of spelling errors in swedish not using a word list en clair. *Journal of Quantitative Linguistics*, 1(3):195–201.

R. Domeij, O. Knutsson, J. Carlberger, and V. Kann. 2000. Granska – an efficient hybrid system for Swedish grammar checking. In T. Nordgård, editor, *Nodalida '99 Proceedings from the 12th Nordiske datalingvistikkdager*, pages 28–40. Department of Linguistics, University of Trondheim.

O. Knutsson, J. Bigert, and V. Kann. forthcoming. Glass box evaluation of a robust shallow parser for Swedish. *Manuscript.*

B. Megyesi. 2002. Shallow parsing with pos taggers and linguistic features. *Journal of Machine Learning Research*, Special Issue on Shallow Parsing(2):639–668.

A. Ratnaparkhi. 1996. A maximum entropy part-of-speech tagger. In *Empirical Methods in Natural Language Processing Conference*, University of Pennsylvania.

H. Schmid. 1994. Probabilistic part-of-speech tagging using decision trees. In *International Conference on New Methods in Language Processing*, Manchester, UK.

J. Sjöbergh. 2002. Combination of pos-taggers for improved accuracy. *Manuscript.*