Software for Annotating Argument Structure

Wojciech Skut, Brigitte Krenn, Thorsten Brants, Hans Uszkoreit
Universität des Saarlandes
66041 Saarbrücken, Germany
{skut,krenn,brants,uszkoreit}@coli.uni-sb.de

Abstract

We present a tool developed for annotating corpora with argument structure representations. The presentation focuses on the architecture of the annotation scheme and a number of techniques for increasing the efficiency and accuracy of annotation. Among others, we show how the assignment of grammatical functions can be automatised using standard part-of-speech tagging methods.

1 The Annotation Scheme

Several features of the tool have been introduced to suite the requirements imposed by the architecture of the annotation scheme (cf. (Skut et al., 1997)), which can itself be characterised as follows:

- Direct representation of the underlying argument structure in terms of unordered trees;
- Rudimentary, flat representations; uniform treatment of local and non-local dependencies;
- Extensive encoding of linguistic information in grammatical function labels.

Thus the format of the annotations is somewhat different from treebanks relying on a context-free backbone augmented with trace-filler annotations of non-local dependencies. (cf. (Marcus et al., 1994), (Sampson, 1995), (Black et al., 1996)) Nevertheless, such treebanks can also be developed using our tool. To back this claim, the representation of structures from the SUZANNE corpus (cf. (Sampson, 1995)) will be shown in the presentation.

2 User Interface

A screen dump of the tool is shown in fig. 1. The largest part of the window contains the graphical representation of the structure being annotated. The nodes and edges are assigned category and grammatical function labels, respectively. The words are numbered and labelled with part-of-speech tags. Any change into the structure of the sentence being annotated is immediately displayed.

Extra effort has been put into the development of a convenient keyboard interface. Menus are supported as a useful way of getting help on commands and labels. Automatic completion and error check on user input are supported.

Three tagsets have to be defined by the user: part-of-speech tags, phrasal categories and grammatical functions. They are stored together with the corpus, which permits easy modification when needed.

The user interface is implemented in Tcl/Tk Version 4.1. The corpus is stored in an SQL database.

3 Automation

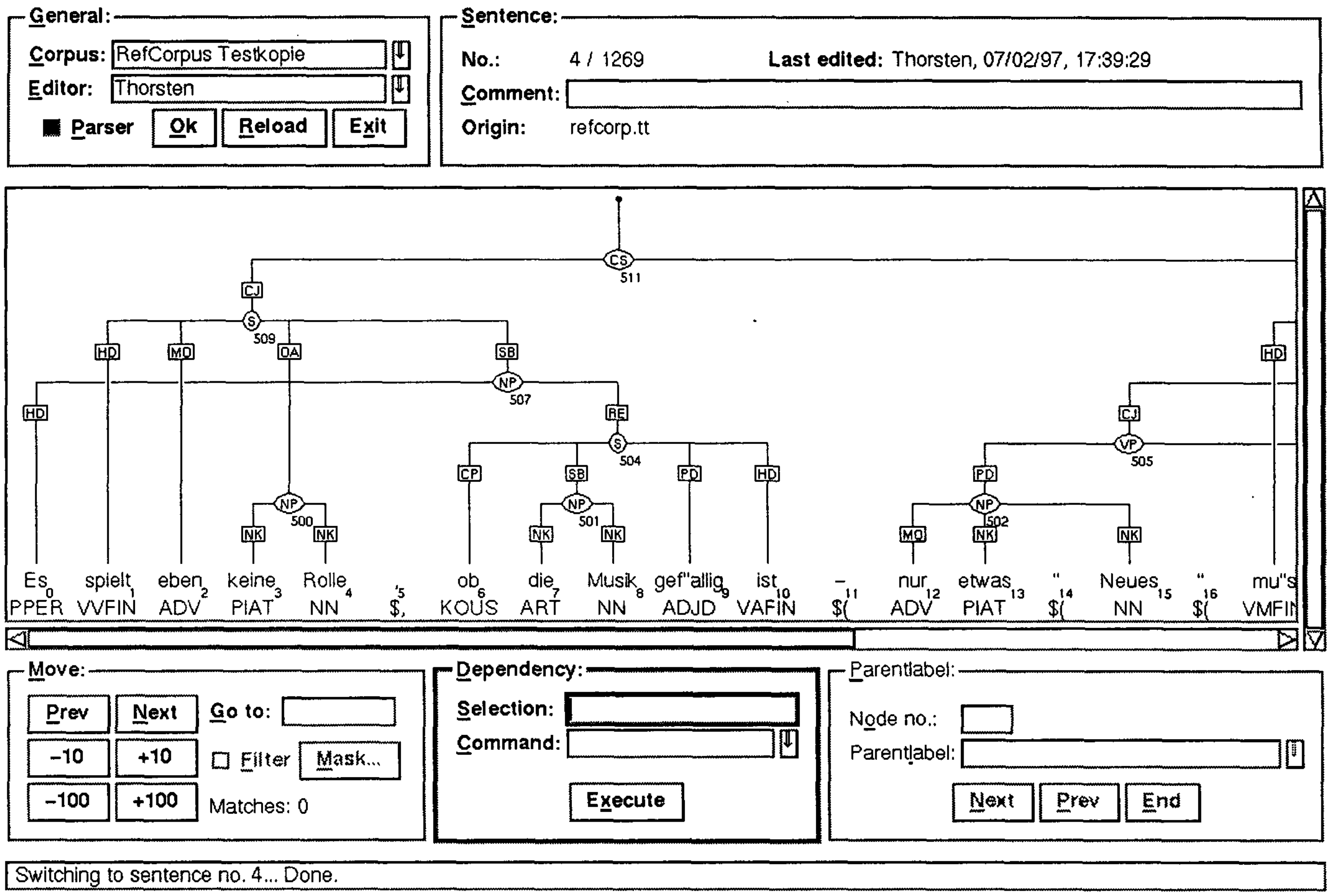
To increase the efficiency of annotation and avoid certain types of errors made by the human annotator, manual and automatic annotation are combined in an interactive way. The automatic component of the tool employs a stochastic tagging model induced from previously annotated sentences. Thus the degree of automation increases with the amount of data available.

At the current stage of automation, the annotator determines the substructures to be grouped into a new phrase and assigns it a syntactic category. The assignment of grammatical functions is performed automatically. To do this, we adapted a standard part-of-speech tagging algorithm (the best sequence of grammatical functions is to be determined for a sequence of syntactic categories, cf. (Skut et al., 1997))

The annotator supervises the automatic assignment of function tags. In order to keep him from missing tagging errors, the grammatical function tagger is equipped with a function measuring the reliability of its output. On the basis of the difference between the best and second-best assignment, the prediction is classified as belonging to one of the following certainty intervals:

Reliable: the most probable tag is assigned,

Less reliable: the tagger suggests a function tag; the annotator is asked to confirm the choice,



Corpus: RefCorpus Testkopie
Editor: Thorsten
No.: 4 / 1269 Last edited: Thorsten, 07/02/97, 17:39:29
Comment:
Origin: refcorp.tt
Matches: 0
Switching to sentence no. 4... Done.

Figure 1: Screen dump of the annotation tool

Unreliable: the annotator has to determine the function himself.

The annotator always has the option of altering already assigned tags.

The tagger rates 90% of all assignments as reliable. Accuracy for these cases is 97%. Most errors are due to wrong identification of the subject and different kinds of objects in S's and VP's. Accuracy of the unreliable 10% of assignments is 75%, i.e., the annotator has to alter the choice in 1 of 4 cases when asked for confirmation. Overall accuracy of the tagger is 95%.

In several cases, the tagger has been able to abstract from annotation errors in training material, which has proved very helpful in detecting inconsistencies and wrong structures.

This first automation step has considerably increased the efficiency of annotation. The average annotation time per sentence improved by 25%.

References

Ezra Black et al. 1996. Beyond Skeleton Parsing: Producing a Comprehensive Large-Scale General-English Treebank With Full Grammatical Analysis. In *The 16th International Conference on Computational Linguistics*, pages 107 – 113, Copenhagen, Denmark.

Mitchell Marcus et al. 1994. The Penn Treebank: Annotating Predicate Argument Structure. In *Proceedings of the Human Language Technology Workshop*, San Francisco. Morgan Kaufmann.

Geoffrey Sampson. 1995. *English for the Computer. The SUSANNE Corpus and Analytic Scheme.*

Wojciech Skut et al. 1997. *An Annotation Scheme For Free Word Order Languages.* In *The 7th Conference on Applied Natural Language Processing*, Washington, DC.