Improving Syntactic Analysis by Parse Reranking

Heike Zinsmeister
Universität Heidelberg
zinsmeister@cl.uni-heidelberg.de

1 Introduction

Linguistic corpora are a widely accepted source of linguistic data. Manually annotated corpora preserve linguistic interpretation in their annotation and thereby make it available for qualitative and quantitative exploitation.

In this paper we report on computational linguistics experiments which use a treebank - a corpus with syntactic annotation - as the basis for deriving a context-free grammar as well as grammar rule frequencies which are employed by a probabilistic parser. Syntactic parsing is one of the core modules in natural language processing and an important preprocessing step in applications such as question answering.

The treebank is used for training the parser and might thus be seen as first-level evidence. For improving the parsing results, we go one step beyond and also use 'secondary evidence': a set of most-probable alternative parses of the same sentence one of which ideally presents the linguistically correct analysis.

The set of alternatives is created by the parser which ranks them according to their probabilites. A preliminary study showed that if the correct analysis is not assigned the highest probability there is room for improvement if the set of 20 or 50 most probable alternatives can be taken into account. The goal is to train a secondary tool which reranks the parser's output such that the linguistically better analyses surface as the most probable parses.

2 Parsing first-level evidence: the treebank

Our parser is derived from the Tübingen Treebank of Written German *TüBa-D/Z*. It consists of a collection of daily newspaper articles published by *die tageszeitung*. In our experiments we used data from release 2 (2005). The annotation of the treebank comprises the following levels: inflectional morphology, parts of speech (using the STTS tagset), syntactic constituents (including a rudimentary marking of clause types), grammatical functions, and topological fields. Local subtrees of the treebank are directly mapped onto context-free production rules which means that context effects such as the relative position in the tree, function of the substructure, or lexical preferences are not taken into account. It is well documented in the literature that enriching local trees with non-local information improves the parser's performance (see e.g. Schiehlen (2004) for an overview).

In order to enrich local subtrees, we adopted Versley (2005)'s tree transformations (his *sclass4* version) which result in more specific labels including: a classification of verbal complexes and of (partial) clauses, a classification of nominal projections according to case, a classification of verbal projections according to their valency, enriching topological field categories with information about included arguments, and binarisation of the verbal complex as well as of coordinate structures.

We use the BitPar parser (Schmid, 2004, release 12/2006) together with a probabilistic grammar model which we derived from the transformed treebank. All unary rules that lead to recursive structures had to be excluded to avoid infinite recursion in lower probable analyses. The parser was trained on 14,726 sentences with an average length of 19.5 words, keeping 300 sentences for evaluation. Example (1) shows rules of topological field nodes (*Mittelfeld, MF*) together with their observed frequencies. The nodes dominate an accusative object (NCX_a) and a dative object (NCX_d) among other daughter constituents (e.g. ADVX).

```
(1)    15      MF_OA_OD   ->    NCX_a NCX_d
       14      MF_OD_OA   ->    NCX_d ADVX NCX_a
       95      MF_OD_OA   ->    NCX_d NCX_a
```

Example (2) exemplifies lexicon rules relating word forms to terminal categories and their (smoothed) frequencies.

```
(2)    kommt       VVFIN_ 103 VVFIN_a 1 VVIMP_ 1
```

```
Krise       NN_sf 19
```

3 Reranking second-level evidence: 50-best parses

BitPar generated up to 50 analyses per sentence (48 analyses on average). For the reranking task we employ BACT (Kudo and Matsumoto, 2004) a tool that reduces the task of reranking to a binary classification task. It discriminates between the best analysis and the rest. BACT extracts all possible subtrees from the parses and determines which subtrees discriminate best between the correct analysis and the incorrect ones.

We tested different settings of the tool's parameters (subtree size, frequency cutoffs for subtrees, training iterations) and also different versions of the training data (with or without functional information and head daughter information). The best results were obtained using full-fledged category information, a subtree size of 6 and a subtree frequency cutoff of 3. Table 1 gives a short summary of the evaluation (using EVALB which employs the PARSEVAL measures). Note that we avoided the problem of guessing unknown words by inclduding all word forms in the lexicon.

	number of sentences	labeled precision	labeled recall	F-score
positive hits of the reranker	15	99.10	98.21	98.56
corresponding parser preferences	15	94.01	91.48	92.73
parser's most probable parses	300	87.24	85.79	86.51
parser + reranker combined	300	87.44	86.06	**86.75**

Table 1: evaluation results using PARSEVAL measures

4 Discussion

The recall of the reranker is low. It selects a candidate in only 5% of the test sentences. Evaluating these 15 candidates with PARSEVAL measures shows that the the parsing quality improves from 92.73 % F-score given the most probable BitPar parses to 98.56 % F-score given the reranker's choices, see Table 1. Combining the reranker's choice with the parser's most probable analyses results in an overall quality improvement of 0.24 % F-score from 86.51 % to 86.75 % on 300 test sentences compared to the original parser performance.

Finally, a short comment on the motivation for our experiments. We started out to investigate parsing performance with a special emphasis on coordinate structures hoping that the global grasp of a reranker would allow us to improve the performance on the notoriously difficult field of coordinate structure parsing. It turned out that the parser very often did not provide the correct analysis in the set of the most probable analyses. As a consequence the reranker picked just once an alternative in the test sentences which included coordination. Reranking is a promising way to improve the overall parsing quality but in the case of coordinate structures we still need to improve the first-level parsers to ensure that the correct analyses are created as candidates in the sets of alternatives of which the second-level reranker is to pick its choice.

References

Kudo, T. and Y. Matsumoto (2004). A Boosting Algorithm for Classification of Semi-Structured Text. In Proceedings of EMNLP 2004.

Schiehlen, M. (2004). Annotation Strategies for Probabilistic Parsing in German. In Proceedings of COLING 2004.

Schmid, H. (2004). Efficient Parsing of Highly Ambiguous Context-Free Grammars with Bit Vectors. In Proceedings of COLING 2004.

Versley, Y. (2005). Parser Evaluation across Text Types. In Proceedings of TLT 2005.