A Context-Sensitive Model for Probabilistic LR Parsing of Spoken Language with Transformation-Based Postprocessing

Tobias Ruland, Siemens AG, ZT IK 5, D-81730 München
Tel.: +49-173-369 30 67, Fax: +49-89-929 54 54, Tobias.Ruland@web.de

Abstract

This paper describes a hybrid approach to spontaneous speech parsing. The implemented parser uses an extended probabilistic LR parsing model with rich context and its output is post-processed by a symbolic tree transformation routine that tries to eliminate systematic errors of the parser. The parser has been trained for three different languages and was successfully integrated in the Verb*mobil* speech-to-speech translation system. The parser achieves more than 90%/90% labeled precision/recall on parsed Verb*mobil* utterances while 3% of German and 5% of all English input cannot be parsed.

1 Introduction

Verb*mobil* (Wahlster, 1993) is a spontaneous speech-to-speech translation system and translates spoken German to English/Japanese and vice versa. The main domains are "appointment scheduling" and "travel planning". There are several parallel analysis and translation modules in Verb*mobil* as described in (Ruland et al., 1998) and one of those analysis modules is the probabilistic parser described in this paper. A schematic diagram of the Verb*mobil* system architecture is shown in figure 1.

The input for the Verb*mobil* speaker independent speech recognizers is spontaneously spoken German (vocabulary 10,254 word forms), English (7,534 word forms) and Japanese (2,848 word forms). The output of the speech recognizers and the prosody module is a prosodically annotated word graph. This word graph is sent to the Integrated Processing module which controls the three parsers (HPSG parser (Kiefer et al., 1999), chunk parser (Abney, 1991) and our probabilistic parser) of the "deep" (semantics based) translation branch of Verb*mobil*. Our probabilistic parser is a shift-reduce parser and uses an A*-search to find the best scored path in the lattice that can be parsed by its context free grammar. The output of the parser is the best scored context free analysis for this path. This syntax tree is passed to a transformation unit that corrects known systematic errors of the probabilistic parser to correct trees. The result of this process is passed to a semantics construction module and processed by the other modules of the deep translation branch as shown in figure 1.

2 Spontaneous Speech Parsing

The Integrated Processing unit uses the acoustic scores of the word hypotheses in the word graph and a statistical trigram model to guide all connected parsers through the lattice using an A*-search algorithm. This is similar to the work presented by (Schmid, 1994) and (Kompe et al., 1997). This A*-search algorithm is used by the probabilistic shift-reduce parser (see section 3) to find the best scored path through the word graph according to acoustic and language model information. If the parser runs into a syntactic "dead end" in the word graph (that is a path that cannot be analyzed by the context-free grammar of the shift-reduce parser), the parser searches the best scored alternative path in the word graph, that can be parsed using the context-free grammar.

We extracted context free grammars for German, English and Japanese from the Verb*mobil* treebank (German: 25,881 trees; English: 23,140 trees; Japanese: 4,534 trees) to be able to parse spontaneous utterances. The treebanks consist of annotated transliterations of face-to-face dialogs in the Verb*mobil* domains and contain utterances like

- *and then well you you you have hotel information*
- *no I am not how about what about Tuesday the sixteenth*
- *actually it yeah so seven hour flight*

The grammar of the parser covers only spontaneous speech phenomenas that are contained in the treebanks.

During the development of the parser we encountered severe problems with the size of the context-free grammar extracted from the treebanks. The German grammar extracted from a treebank containing 20,000 trees resulted in a LALR parsing table with more than 3,000,000 entries, which cannot be trained on only 20,000 utterances. The reason was that there are many rules in the treebank, which occur only once or twice but inflate the context-free grammar and thus the size of the

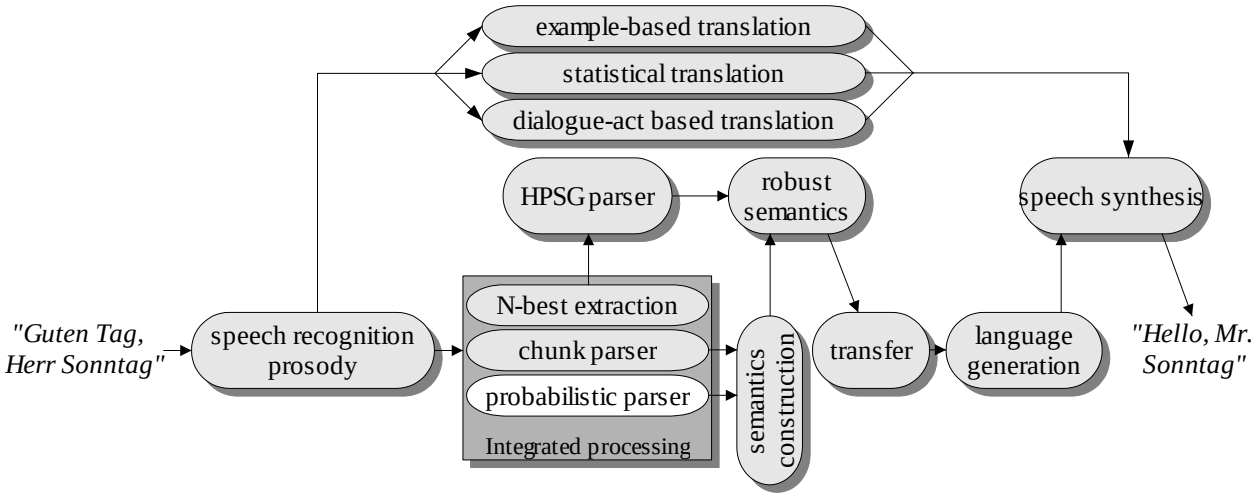

Figure 1

size of the parsing table. For this reason we eliminate trees from our training material containing rules that occur unfrequently in the treebank and use only rules achieving a minimal rule count. This threshold is determined experimentally in our training process.

3 A new context sensitive approach to probabilistic shift-reduce parsing

The work of Siemens in Verb*mobil* phase 1 showed that a combination of shift-reduce and unification-based parsing of word graphs works well on spontaneous speech but is not very robust on low-word-accuracy input (the word error rate of the Verb*mobil* speech recognizers is about 25% today). One way to gain a higher degree of robustness is to use a context-free grammar instead of an unification-based grammar, hence we decided to implement and test a context-free probabilistic LALR parser in Verb*mobil* phase 2.

3.1. Previous approaches

There are several approaches (see for example (Wright & Wrigley, 1991), (Briscoe & Carroll, 1993/1996), (Lavie, 1996) or (Inui et al., 1997)) to probabilistic shift-reduce parsing but only Lavie's parser, whose probabilistic model is very similar to (Briscoe & Carroll, 1993), has been tested on spontaneously spoken utterances.

While the model presented by (Wright & Wrigley, 1991) was equivalent to the standard PCFG (probabilistic context-free grammar, see (Charniak, 1993)) model, which is not context-sensitive and thus has certain limitations in the precision that it can achieve, later work tried to implement slight context-sensitivity (as e.g. the probability of a shift/reduce-action in Briscoe and Carroll's model depends on the current and succeeding LR parser state and the look-ahead symbol).

3.2. Bringing context to probabilistic shift-reduce parsing

Like other work on probabilistic parsing our model is based on the equation

$$P(T|W)=P(T)\cdot P(W|T) \quad , \qquad (2)$$

where T is the analysis of a word sequence W and a widely used approximation for $P(W|T)$ is given by

$$P(W|T)\approx\prod_{w_i\in W} P(w_i|l_i) \quad , \qquad (3)$$

where l_i is the part-of-speech tag for word w_i in analysis T.

Finding a realistic approximation for $P(T)$ is very difficult but important to achieve high parsing accuracy. Supposed we approximate $P(W|T)$ by equation (3). Then $P(W|T)$ is nothing more than $P(W|L)$, where L is the part-of-speech tag sequence for a given utterance W. If our goal is to select the best analysis T for a given tag sequence L we do not necessarily depend on a good approximation of $P(T)$, but simply select the best analysis for a given L by finding a T that maximizes $P(T|L)$ (and not $P(T)$). Hence, in our model we use $P(T|L)$ instead of $P(T)$ so that

$$\sum_k P(T_k|L)=1 \quad , \qquad (4)$$

where T_k is the set of possible analyses for L. Let D be the set of all complete shift-reduce parser action sequences for L, i.e. d_k is the sequence of shift- and reduce-actions that generates analysis T_k. Then we

can define $P(d|L)$ ($=P(T|L)$) as

$$\forall d \in D: \quad P(d|L) = \prod_{j=1}^{|d|} P(a_{d,j}|k_{d,j}) \quad , \quad (5)$$

where $|d|$ is the number of parser actions in d, $a_{d,j}$ is the jth parser action in d and $k_{d,j}$ is the context of the parser while executing $a_{d,j}$.

3.3. Choosing a context

"Context" in equation (5) might be everything. It can be the classical (CurrentParserState; LookAheadSymbol)-tuple, it may also contain information about the following (look-ahead) word(s), elements on the parser stack or the most probable dialogue act of the utterance, even semantical information about roles of the syntactical head of the phrase on the top of the parser stack.

The training procedure of our probabilistic parser is straightforward:

1. Construct complete parser action sequences for each tree in the training set. Save all information (on every action) about the whole "context" we have chosen to use.
2. Count the occurences of all actions in different subcontexts. A subcontext may be the whole context or a (even empty) selection of features of the whole context. Compute the probability of a parser action regarding to the subcontext as the relative frequency of the action within this subcontext.

The reason why we build subcontexts is that there is a relevant sparse-data-problem in Verb*mobil*. A treebank containing between 20,000 and 30,000 trees is too small to give reliable values for larger contexts in a parsing table containing 500,000 entries or more. Hence we use the smoothing technique that is known as backing-off in statistical language modelling (Charniak, 1993) and approximate the probability of an action a with context k using its subcontexts c_j:

$$P(a|k) = \alpha_0 P(a|c_0) + \alpha_1 P(a|c_1) + \ldots \alpha_n P(a|c_n) \quad (6)$$

with α_j summing up to 1. The values for α_j are determined experimentally. We have chosen three contexts for evaluation (K1 and K2 also exist in our model but are irrelevant for this evaluation):

- K3: LR parser state and look-ahead symbol,
- K4: K3 plus phrase head of the top element of the LR parsing stack,
- K5: K4 plus look-ahead word.

Please see section 5.1. for the detailed results of this evaluation.

4 Transformation-based error correction

Parsing spontaneous speech - even in a limited domain - is a quite ambitious task for a context free grammar parser. We have a large set of non-terminals in our grammar that also encode functional information like *Head* or *Modifier*, grammatical information like accusative-complement or verb-prefix besides phrase structure information. Our current grammars contain 240 non-terminals for German, 178 for English and 200 for Japanese and the lexicon is derived automatically from the tree bank and external resources (there were only minor efforts in improving the lexicon manually).

During the development of the parser we observed a constantly declining *Exact Match* rate of the parser from over 80% in the early stages (with just a few hundred trees of training data) to under 50% today. The reason was that the first training samples were simple utterances on "appointment scheduling" only, while the treebank nowadays contains spontaneous utterances from two domains and that there was a growing number of inconsistencies in the treebank due to annotation errors and a growing number of annotators. Hence we had to develop a technique to improve the exact match rate particularly with regard to the following semantics construction process that depends on correct syntactic analyses to produce a correct semantic representation of the utterance.

(Brill, 1993) applied transformation-based learning methods to natural language processing, especially to part-of-speech tagging. He showed that it can be effective to let a system make a first guess that may be improved or corrected by following transformation-based steps. We observed many systematical errors in the output of the probabilistic parser, hence we adopted this idea and took the probabilistic shift-reduce parser as the guesser and tried to learn tree transformations from our training data to improve this first guess. We integrated the learned transformations into Verb*mobil* as shown in figure 2.

The transformations map a tree to another tree, changing parts that had be identified as incorrect in the learning process. The output of the learning process are simple Prolog clauses of the form

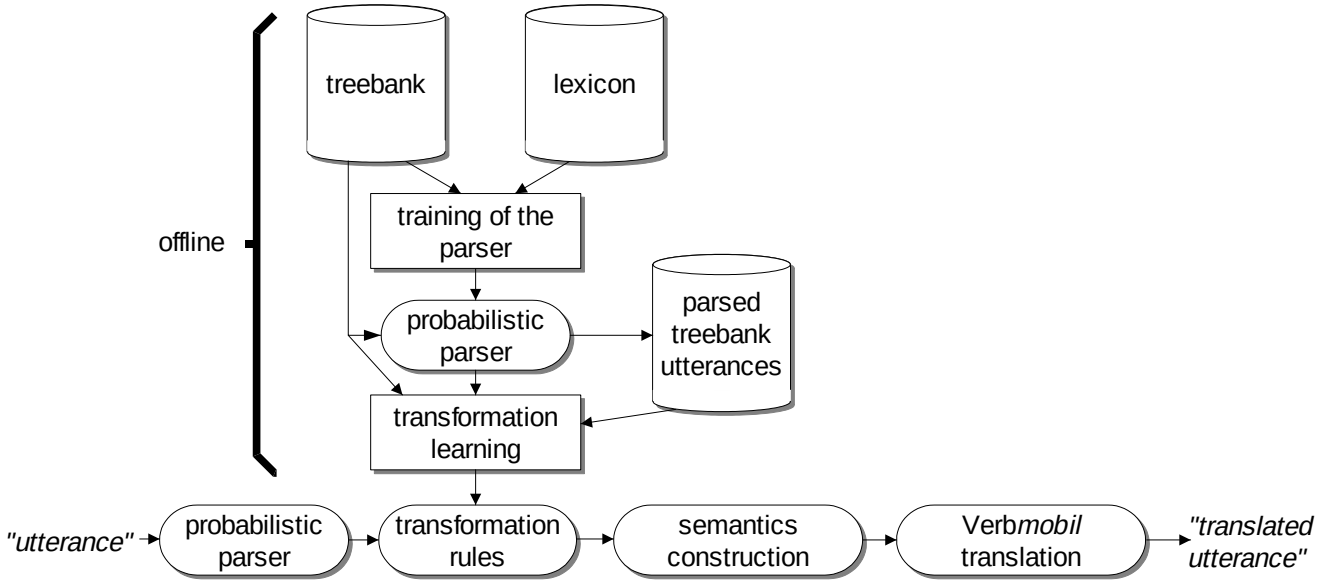

Figure 2

`trans(+InputTree,-OutputTree):-!.`, that are sorted by the number of matches on the training corpus.

4.1 The Problem

The task of learning transformations that are suitable to post-process the output of a probabilistic parser can be implemented as shown in figure 2:

1. train the probabilistic parser on a training set *O* (containing utterances and their human-annotated analyses).
2. parse all utterances of *O* and save the corresponding parser outputs *P*.
3. find the set of as-general-as-possible transformations *T* that map all incorrect trees of *P* into corresponding correct trees in *O* and select the "optimal" transformation from this set.

The first point has been described in section 3.3. and the second point is trivial. The *as-general-as-possible transformation* is the mapping of a tree of *P* into a tree for the same utterance in *O* that achieves a high degree of generalization and fulfils certain conditions, which are explained in section 4.2.

4.2. The Learning Algorithm

The learning algorithm to derive the most general tree transformations for incorrect trees in *O* is straightforward. To find the most general transformation for a source tree $\mathcal{P} \in P$ to be mapped into a destination tree $\mathcal{O} \in O$ do:

1. find the set $\mathcal{G}$ of all common subtrees of $\mathcal{P}$ and $\mathcal{O}$.
2. find the set $\mathcal{T}$ of all potential transformations. A transformation t is formed by substitution (θ_i) of one or more elements of $\mathcal{G}$ by logical variables in $\mathcal{P}$ und $\mathcal{O}$ (i.e. t: $\theta_i(\mathcal{P}) \Rightarrow \theta_i(\mathcal{O})$)
3. choose the "optimal" transformation from $\mathcal{T}$.

Syntactical trees are represented as Prolog terms in our learning process. Since the transformation should be able to map large correct structures in $\mathcal{P}$ to their (correct) counterparts in $\mathcal{O}$ the first point of the algorithm is done by setting $\mathcal{G}$ equal to the set of all (Prolog) subterms that are common in $\mathcal{P}$ and $\mathcal{O}$ (i.e. $\mathcal{G}$=`subterms(`$\mathcal{P}$`)`$\cap$`subterms(`$\mathcal{O}$`)`).[1]

It is crucial here to attach a unique identifier to each word (like "1-hi","2-Mr.","3-Smith") because one word (like the article "the") could occur several times in one sentence and it is important to keep those occurences separated for the second step of the learning algorithm.

The second step computes all potential tree transformations by substituting one or more elements of $\mathcal{G}$ in $\mathcal{P}$ and $\mathcal{O}$ by identical (Prolog) variables. In this regard "substitution" is an operation, that is inverse to the substitution known

1 `subtrees(+Tree,-SubTrees)` could simply be defined (in Prolog) as

```
subtrees(+T,-S):- findall(X,subtree(X,T),S).
subtree(S,S).
subtree(S,_:L) :- member(M,L),subtree(S,M).
```

Trees are represented as terms like a:[b,c], for example.

from predicate logic.

Choosing the "optimal" transformation from the space of all transformations in the third step is a multi-dimensional problem. The dimensions are:

- fault tolerance
- coverage of the training corpus
- degree of generalization

Fault tolerance is a parameter that indicates how many correction errors on the training corpus the human supervisor is willing to tolerate, i.e. how many of the correct parser trees may be transformed into incorrect ones. Accepting transformation errors may improve the grade of generalization of the transformation but for Verb*mobil* we decided not to be fault tolerant. A correct analysis should be kept correct in our point of view.

Coverage of the training corpus means that if step 2 of the learning algorithm has found several possible transformations for a $\mathcal{P}$-$\mathcal{O}$-pair the transformation t$\in\mathcal{T}$ that covers the most examples in *P*/*O* should be preferred because this transformation is likely to occur more often in the running system or test situation.

Besides the heuristical generalization criterion of coverage of the training corpus we also introduced a formal one. If there are several transformations that do not generate errors on the training corpus and have exactly the same maximum coverage, we select the transformation which has the smallest mean distance of its logical variables to the root of the tree, because we expect the most general transformation to have its variable parts "near the root" of the trees. Distance is measured in levels from the root. For example, the transformation in figure 3 has a mean root distance of the variables of ((1+2) + (1+3)) / 4 = 1.75.

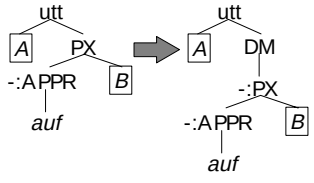

Figure 3

Using this learning algorithm we generate a set of optimal transformations for many errors the parser produced on the set of training utterances. There are still some utterances for which no valid transformation can be found because all potential transformations would generate errors on the training corpus, what we are not willing to accept.

5 Evaluation results

At the time this paper is written we have done several experiments on different aspects of our work, some of which are published here.

5.1. Experiments on context sensitivity

The question of this experiment was: "We have developed a probabilistic parsing model using more context information. Does it generate any benefit?" To answer this question we trained the parser on 19,750 german trees and tested on 1,000 (unseen) utterances with contexts of different sizes (the contexts K3, K4 and K5 are explained in section 3.3). As shown in figure 4 (the x-axis is a weight that controls the influence of the context in the backing-off process) labeled precision of the K5-parser performs always better than the parsers using less context. Labeled recall of the K5-parser is superior as long as the large context is not overweighted. Higher weights increase some kind of "memory effect" so that the trained model does not generalize well on (unseen) test data. The optimal K5 weight is around 0.1 and 0.2 as you can see in figure 4.

5.2. Evaluation of the probabilistic parser

We evaluated the parser on German, English and Japanese Verb*mobil* data. The results of this evaluation are given in the following table:

	German	***English***	***Japan.***
Training set [trees]	19.750	17.793	3.218
Test set [utterances]	1.000	1.000	300
Exact Match	46,3%	55,4%	67,7%
Incorrect parses	50,3%	39,3%	21,3%
Not parsed	3,4%	5,3%	11,0%
context-free rules	988	2.205	932
Labeled Precision	90,2%	90,6%	84,9%
Labeled Recall (all utterances)	83,5%	78,5%	63,1%
Labeled Recall (parsed utterances)	91,0%	90,9%	86,3%

It is quite interesting that despite of the low exact match rate our parser achieves high precision/recall values on parsed utterances. The reason is that we have - for the semantics construction process - a large number of non-terminal symbols in our context-free grammars and the parser often chooses

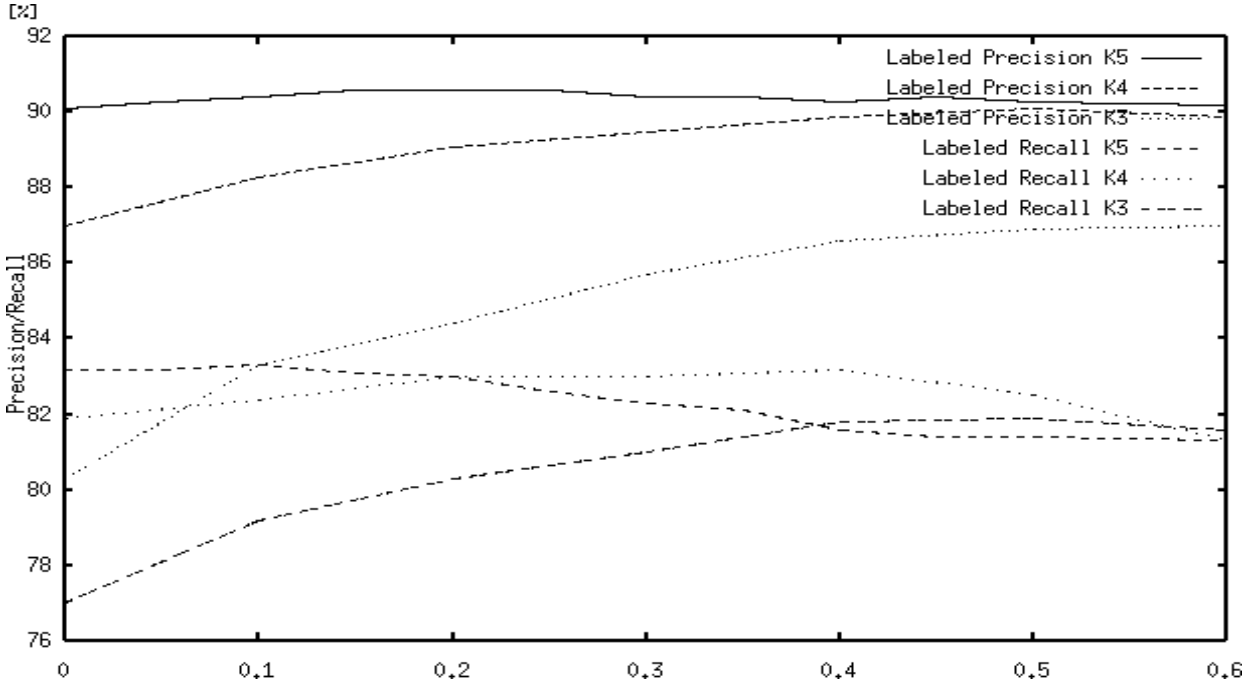

Figure 4

only one or two slightly incorrect symbols per parse. The mean parsing time per utterance was about 400ms for German and English and about 30ms for Japanese on a 166-Mhz Sun Ultra-1 workstation.

5.3. Influence of transformation-based error correction

It is important to have a very high exact match rate for the semantics construction process. As shown in the table of section 5.2. the exact match rates are quite low thus we have learned transformations from the training data to improve the output of the German and English parser (there was not enough training data to do so for Japanese) and evaluated the results shown in the following table (*TT* is an abbreviation for *Tree Transformations*).

As shown in this table the tree transformations improve the exact match rate relatively by 16% for German and 10% for English.

	German	***English***
Exact Match (w/o TT)	46,3%	55,4%
Incorrect parses	50,3%	39,3%
Not parsed	3,4%	5,3%
Exact Match (after TT)	53,8%	61,2%
Incorrect parses (after TT)	42,8%	33,5%
Labeled Precision (w/o TT)	90,2%	90,6%
Labeled Precision (after TT)	90,8%	91,4%
Labeled Recall (all utterances, w/o TT)	83,5%	78,5%
Labeled Recall (all utterances, after TT)	84,0%	79,2%
Labeled Recall (parsed utterances, w/o TT)	91,0%	90,9%
Labeled Recall (parsed utterances, after TT)	91,6%	91,7%

6 Conclusion

In this article we have extended probabilistic shift-reduce parsing to be more context-sensitive than previous works and have demonstrated that a bigger context improves the performance of a probabilistic shift-reduce parser. It was shown that our model is suitable to parse utterances of the Verb*mobil* domain in three different languages. It was also shown that the exact match rate of a probabilistic parser can be improved significantly using a symbolic transformation-based post-processing step.

Our method of learning tree transformations has generated first promising results but it is based on the mapping of whole trees to whole trees. It could be a direction of further research to extend this process of learning transformations on smaller

(sub-)structures like single phrases. That should improve generalization and help improving the exact match rate on the difficult domain of parsing spontaneously spoken utterances.

Acknowledgements

This research was supported by the German Federal Ministry for Education, Science, Research and Technology under grant no. 01IV701A3. I would like to thank all Verb*mobil* colleagues, especially the colleagues of IMS Stuttgart and University of Tübingen, who supported this work by their cooperation. I would also like to thank the anonymous reviewers for their valuable comments.

References

Abney, S. P. *Parsing by Chunks.* In: Berwick, R. C., Abney, S. P., Tenny, C. (eds.) *Principle-Based Parsing: Computation and Psycholinguistics.* Kluwer Academic Publishers, 1991.

Brill, E. *A Corpus-Based Approach To Language Learning*. PhD Thesis, Department of Computer and Information Science, University of Pennsylvania, 1993.

Briscoe, T., Carroll, J. *Generalized Probabilistic LR Parsing of Natural Language (Corpora) with Unification-Based Grammars*. In: Computational Linguistics, Vol. 19, No. 1, 1993.

Briscoe, T., Carroll, J. *Apportioning Development Effort in a Probabilistic LR-Parsing System through Evaluation.* In: *Proceedings of the ACL SIGDAT Conference on Empirical Methods in Natural Language Processing*, Philadelphia, PA. 92-100, May 1996.

Charniak, E. *Statistical Language Learning.* MIT Press, Cambridge, Mass., 1993.

Inui, K., Sornlertlamvanich, V., Tanaka, H., Tokunaga, T. *A New Formalization of Probabilistic GLR Parsing.* In: *Proceedings of the International Workshop on Parsing Technologies*, 1997.

Kiefer, B., Krieger, H.-U., Carroll, J., Malouf, R. *A Bag of Useful Techniques for Efficient and Robust Parsing.* In: *Proceedings of the 37th Annual Meeting of the Association for Computational Linguistics*, ACL-99, pp. 473-480, 1999.

Kompe, R., Batliner, A., Block, H.-U., Kießling, A., Niemann, H., Nöth, E., Ruland, T., Schachtl, S. *Improving Parsing of Spontaneous Speech with the Help of Prosodic Boundaries.* In: *Proceedings of the ICASSP*, pp. 75-78, München, 1997.

Lavie, A. GLR*: *A Robust Grammar-Focused Parser for Spontaneously Spoken Language*. PhD Thesis, Carnegie Mellon University, Pittsburgh, 1996.

Ruland, T., Rupp, C.J., Spilker, J., Weber, H., Worm, K. *Making the Most of Multiplicity: A Multi-Parser Multi-Strategy Architecture for the Robust Processing of Spoken Language.* In: *Proceedings of the ICSLP*, Sidney, 1998.

Schmid, L. *Parsing Word Graphs Using a Linguistic Grammar and a Statistical Language Model.* In: *Proceedings of the IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP '94)*, Adelaide, 1994.

Wahlster, W. *Translation of face-to-face dialogs.* In: *Proceedings of MT Summit IV*, Kobe, Japan, pp. 127-135, July 1993.

Wright, J. H., Wrigley, E. N. *GLR Parsing with Probability.* In: Tomita, M. (ed.) *Generalised LR Parsing.* Kluwer Academic Publishers, Boston, 1991.