

©2005 IEEE. Personal use of this material is permitted. However, permission to reprint/republish this material for advertising or promotional purposes or for creating new collective works for resale or redistribution to servers or lists, or to reuse any copyrighted component of this work in other works must be obtained from the IEEE.

# INTEGRATING A NON-PROBABILISTIC GRAMMAR INTO LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

*René Beutler, Tobias Kaufmann and Beat Pfister*

Speech Processing Group  
Computer Engineering and Networks Laboratory  
ETH Zurich, Switzerland

{beutler, kaufmann, pfister}@tik.ee.ethz.ch

## ABSTRACT

We propose a method of incorporating a non-probabilistic grammar into large vocabulary continuous speech recognition (LVCSR). Our basic assumption is that the utterances to be recognized are grammatical to a sufficient degree, which enables us to decrease the word error rate by favouring grammatical phrases. We use a parser and a hand-crafted grammar to identify grammatical phrases in word lattices produced by a speech recognizer. This information is then used to rescore the word lattice. We measured the benefit of our method by extending an LVCSR baseline system (based on hidden Markov models and a 4-gram language model) with our rescoring component. We achieved a statistically significant reduction in word error rate compared to the baseline system.

## 1. INTRODUCTION

To incorporate knowledge about the structure of language above word level, most speech recognizers use simple order statistics like N-grams. Such models are based on a notion of language as a linear sequence of words. However, natural language is more precisely described in terms of hierarchical structures and dependencies between constituents. The German language, for which we conducted our experiments, poses a particular challenge for N-grams. German has a relatively free word order and is highly inflected. The choice of the inflectional morphemes depends on grammatical constraints whose scope is usually not restricted to a local neighborhood of words. In short, it seems to be desirable to include linguistic knowledge into speech recognition [1].

In the past few years research has begun to show progress towards more adequate models of spoken language. The most successful techniques applied in previous work are based on N-best rescoring by statistical parsing. Most of these approaches do incorporate linguistic concepts like lexical heads [2, 3, 4] or even adjuncts, complements

and gaps [5]. However, these models are still rather simple from the linguistic point of view.

Our approach involves the use of a state-of-the-art non-probabilistic formal grammar which accepts grammatical phrases and rejects ungrammatical ones with high precision. The grammar is hand-crafted and incorporates insights of current linguistic research. The main advantage of using a linguistically sound and precise model of a natural language is that it allows us to better exploit syntactic constraints for rejecting incorrect hypotheses.

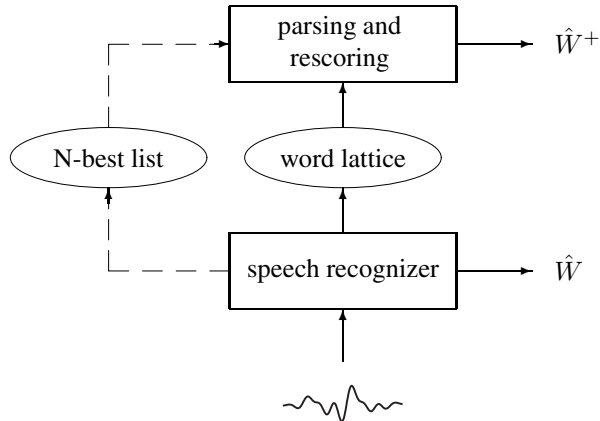
A drawback of using non-probabilistic grammars is that it is not obvious how to integrate the hard decisions made by the parser into a probabilistic speech recognition framework. In particular, there must be some way to deal with the situation where the parser does not accept any of the recognizer hypotheses. This might be the case if none of the hypotheses is correct, or if the correct hypothesis is not grammatical, or if the correct hypothesis is grammatical but not covered by the given grammar. We will present an approach which can deal with these problems.

Our basic assumption is that the utterances to be recognized are grammatical to a sufficient degree, which enables us to decrease the word error rate (WER) by favouring grammatical phrases. However, we do not require that the utterances to be recognized are covered by the grammar: even if some hypothesis cannot be fully parsed, the presence of large grammatical fragments (e.g. noun phrases) may be a strong indicator for its correctness.

Approaches similar to the one described here have been followed in [6] and [7], namely the robust parsing of word graphs. Our work is different from these projects in that our grammar is not domain specific and our goal is to increase word accuracy rather than semantic accuracy.

In this paper we will describe a method of incorporating a non-probabilistic grammar into speech recognition. By applying this method to a dictation task we achieved a relative reduction of the WER of 27% which is statistically significant.

Sections 2 and 3 describe how we integrate syntactic



**Fig. 1.** A speech recognizer is used to measure the baseline recognition accuracy for a given task. We aim at outperforming this baseline system by rescoring the word lattice created by the recognizer by means of linguistic knowledge. The N-best list is used to drive the parser to work on the most promising hypotheses first.

knowledge into our speech recognition system. The grammar formalism is discussed in Sect. 4. We report and discuss our results in Sect. 5 followed by the conclusions in Sect. 6.

## 2. ARCHITECTURE

Our aim is to provide evidence that our approach indeed improves LVCSR accuracy. To this end, we extend a speech recognizer with our rescoring component in a way such that any improvement of recognition accuracy can be clearly attributed to that component. The architecture shown in Fig. 1 fulfills this requirement: a speech recognizer creates word lattices and at the same time provides the baseline word error rate. The word lattices are subsequently processed by a natural language processing module. By comparing the word error rate of the enhanced system with the baseline word error rate we can directly quantify the benefit of our approach.

Initially, a word lattice is produced by the baseline speech recognizer. Due to the uncertainty of word boundaries in continuous speech, the same word sequence may be represented by several lattice paths with different acoustic scores. The parser only considers the word sequence of a path but not its score. Consequently, there is no use in parsing several paths that differ in score only. By ignoring acoustic scores and timing information we create a word graph which represents all word sequences of the lattice in compact form and thus can be processed more efficiently by the parser.

Ideally, the parser processes each path in the word graph, producing all phrases which can be derived from the

corresponding word sequences. As the number of paths in a word graph may be huge, one has to focus on the most promising hypotheses. For this reason only the paths in the baseline system’s N-best list are parsed, starting with the best hypothesis. If the parsing time exceeds some predefined limit, the parsing procedure is terminated. Parsing can potentially lead to a combinatorial explosion of hypotheses resulting in large processing times. The parsing time limit guarantees a worst-case running time of the experiments.

The phrases derived by the parser are used to rescore the recognizer lattices. However, the final solution is not restricted to the N-best paths but can rather be any path in the lattice. The rescoring step will be described in the next section.

## 3. SCORING SYNTACTIC STRUCTURES

A speech recognizer’s aim is to find the word sequence which was most likely uttered given an acoustic observation  $O$ . The *maximum a posteriori* (MAP) criterion chooses the word sequence  $W$  such that the product of the acoustic likelihood  $P(O|W)$  and the language model probability  $P(W)$  is maximized. In practical applications the acoustic likelihood and the language model probability have to be balanced to optimize performance. Also, adding a word insertion penalty has proven to be advisable to get optimal results.

We extend the MAP criterion with an additional parsing score  $f(W)$  which allows us to favour grammatical utterances:

$$\hat{W}^+ = \arg \max_W P(O|W) \cdot P(W)^\lambda \cdot |W|^{ip} \cdot f(W) \quad (1)$$

$\lambda$  denotes the language model weight, the norm  $|\cdot|$  measures the length of a sequence and  $ip$  is the word insertion penalty. As we use a non-probabilistic rule-based grammar the parser does not provide a score but only syntactic structures. In the remainder of this section we explain how the parsing score  $f(W)$  can be computed from syntactic structures.

Let  $W$  be a word sequence in the lattice spanning the whole utterance.  $W$  can be decomposed into a sequence  $U = \langle u_1, u_2, \dots, u_n \rangle$  of so-called parsing units  $u_i$ . A parsing unit  $u_i$  represents a word sequence  $w(u_i)$  which the parser identified as being grammatically correct. The decomposition is such that the concatenation  $w(u_1) \circ w(u_2) \circ \dots \circ w(u_n) = W$ . Note that for a given  $W$  there may exist several different decompositions.

We distinguish three types of parsing units. The smallest unit represents a single word. Units which are larger than one word but do not span a whole utterance are called fragment units. A unit spanning the whole utterance is called an utterance unit. We first define the parsing score  $s(\cdot)$  for a single parsing unit to depend on its unit type:

sentence	parsing score
(Anna and Bob go to school)	$f(W) = c_\alpha$
(Anna and Bob) <sup><math>c_\alpha</math></sup> (so) (to school)	$f(W) = c_\beta^2 \cdot c_\gamma$
(Anna) <sup><math>c_\beta</math></sup> (and) <sup><math>c_\gamma</math></sup> (bobbed) <sup><math>c_\beta</math></sup> (go) <sup><math>c_\beta</math></sup> (two) (school)	$f(W) = (c_\gamma)^6$
<sub><math>c_\gamma</math> <math>c_\gamma</math> <math>c_\gamma</math> <math>c_\gamma</math> <math>c_\gamma</math> <math>c_\gamma</math></sub>	

**Table 1.** Three examples illustrating how the parsing score is computed.

$$s(u, W) = \begin{cases} c_\alpha & \text{if } w(u) = W, \\ c_\beta & \text{if } 1 < |w(u)| < |W|, \\ c_\gamma & \text{else} \end{cases} \quad (2)$$

where  $c_\alpha$ ,  $c_\beta$  and  $c_\gamma$  denote the scores for utterance units, fragment units and single word units, respectively. The score of a decomposition of  $W$  is

$$g(\langle u_1, \dots, u_n \rangle, W) = \prod_{i=1}^n s(u_i, W) . \quad (3)$$

The score of word sequence  $W$  is the maximal score of all its valid decompositions:

$$f(W) = \max_U g(U, W) . \quad (4)$$

Note that  $f(W)$  is always defined, even if the utterance is not fully parsable, because  $W$  can always be decomposed into single word units. Therefore a fall-back mechanism for unparseable sentences is superfluous. Table 1 illustrates how parsing scores are computed.

The parameters  $\lambda$ ,  $ip$ ,  $c_\alpha$ ,  $c_\beta$  and  $c_\gamma$  are optimized on development data to minimize the empirical word error rate (cf. Sect. 5.1).

#### 4. GRAMMAR FORMALISM

A good grammar should accept as many grammatical word sequences as possible and at the same time reject as many ungrammatical word sequences as possible. Precision is the main requirement of a grammar to be used in our architecture: it only makes sense to favour the parsable word sequences if they are very likely to be correct. Note that since our approach can deal with unparseable word sequences, there is no need to artificially weaken the grammar rules, as is sometimes done to achieve a higher robustness.

However, it is also important that the grammar covers a wide range of syntactic constructions. It is necessary that the syntactically analyzable parts of the utterance are as large as possible, since only the words within an analyzable

unit can be constrained. For instance, knowing a German verb’s valency structure allows to constrain the inflectional endings of its objects (case, agreement of subject and finite verb). The disambiguation of inflectional endings is important since such endings are easily confused by the recognizer. In order to favour a given word sequence for obeying the valency constraint, the parser has to be able to derive a unit which contains the verb and all its objects. This in turn requires that each individual object is fully parsable.

As a grammar formalism, we have chosen *Head-Driven Phrase Structure Grammar* (HPSG) [8]. HPSG is a framework for linguistic theories. It uses linguistically motivated abstractions which substantially simplify the task of writing precise large-scale grammars. An important property of HPSG is that words or phrases are represented by complex feature structures which precisely specify their syntactic properties.

In addition to its linguistic adequacy, HPSG is well-suited for natural language processing applications. Existing systems like [9] demonstrate that parsing efficiency can be reasonably high, even for large HPSG grammars which cover a substantial fragment of a natural language.

To carry out this work, we have developed an efficient bottom-up chart parser for HPSG. The parser was especially designed for the application on word graphs. Typically, there is much overlap between the paths in a word graph. To prevent identical word sequences from being analyzed several times, hypotheses from previously parsed paths are reused.

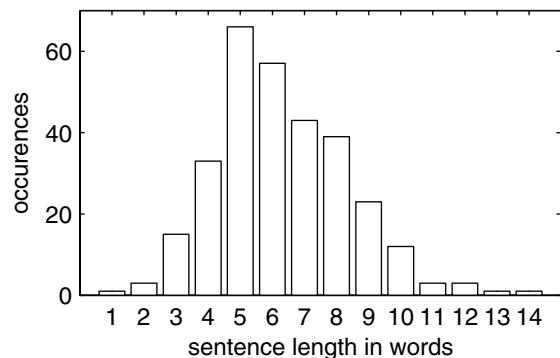
## 5. EXPERIMENTS

### 5.1. Task and Experimental Setup

To initiate our work we looked for a task with a manageable complexity. We identified dictation texts for pupils as a suitable recognition task. The first 300 sentences (1892 words) from a dictation exercise book [10] were read aloud by a single male speaker and recorded with a headset microphone sampled at 16 kHz in an office environment with low background noise.

Although these sentences are rather simple, they comprise a wide variety of grammatical constructions. The sentence lengths range from a single word up to 14 words (cf. Fig. 2). Note that these sentences occur neither in the acoustic training corpus nor in the text corpus used for the estimation of the statistical language models.

Additionally, 30 minutes of speech were recorded from the same speaker under the same conditions. This data was used to adapt the speaker-independent acoustic models. We applied maximum likelihood linear regression (MLLR) and the maximum a posteriori approach (MAP) in a supervised manner.



**Fig. 2.** Distribution of sentence lengths for the given task.

In order to rescore lattices by means of syntactic knowledge as described in Sect. 3, the parameters  $\lambda$ ,  $ip$ ,  $c_\alpha$ ,  $c_\beta$  and  $c_\gamma$  must be optimized on development data. Therefore the 300 sentences were divided into two sets: two-thirds were used as development set on which the parameters were optimized and one-third was used for testing. Cyclic permutation of the sets allowed us to use all 300 sentences for testing. The parameters of the baseline recognizer and the rescoring parameters were optimized separately to minimize the empirical word error rate. Because the WER is not a continuous objective function, gradient descent methods cannot be applied directly. The downhill simplex method known as amoeba search (a multidimensional unconstrained nonlinear minimization algorithm) was applied instead [11].

The decoder of the baseline system performed two passes. In the first pass speaker-adapted monophone models and a 2-gram language model were used. The HTK [12] decoder was used to create word lattices by time-synchronous Viterbi beam search. At each HMM state, the 5 best tokens were taken into account. In the second pass, the resulting lattices were rescored with cross-word triphones and a 4-gram language model.

The 20 best scored recognizer hypotheses were processed by the parser as described in Sect. 2. The parsing timeout was set to one minute on a 1 GHz UltraSPARC III processor. Finally, the optimal word sequence was extracted by combining acoustic, language model and parsing scores.

## 5.2. Data and Models

Speaker-independent continuous density HMMs have been trained by means of HTK on the PhonDat 1 corpus which contains about 21 hours of clean continuous German speech of 200 speakers. The sampling rate was 16 kHz. The 39-dimensional feature vector consisted of 13 mel-frequency cepstral coefficients (MFCCs) including the 0th coefficient, the delta and the delta-delta coefficients. The HMMs were three-state left-to-right models. For each of the 40 phones a context-independent monophone model with 32 Gaussian

basic language models	data	PPL
class-based 2-gram (500 classes)	70M	566
2-gram <sub>30k</sub>	30k	389
2-gram <sub>70M</sub>	70M	291
4-gram <sub>70M</sub>	70M	222

interpolated models	PPL
<i>decoding:</i>	
2-gram <sub>70M</sub> + 2-gram <sub>30k</sub> + class-2-gram	251
<i>rescoring:</i>	
4-gram <sub>70M</sub> + 2-gram <sub>30k</sub> + class-2-gram	198

**Table 2.** Perplexities (PPL) measured on the test data.

mixtures was trained. Context-dependent cross-word triphone models with 16 Gaussians were trained as well. The states have been tied using a decision-tree-based clustering according to yes/no questions regarding phonetic context, resulting in 875 unique states and 2540 triphone models.

Interpolated back-off N-grams serve as statistical language models (LM). Two text corpora were used: a 70 million words corpus (German newspaper text and literature) and a 35 thousand words corpus (texts from various dictation exercise books). The first one allows smoother estimates but the latter is closer to our task. The interpolation weights and the discounting strategy were optimized for low perplexity on the test data in order to get a competitive LM as our baseline.<sup>1</sup> The N-gram probabilities were estimated for a vocabulary of 7k words with the SRI language modeling toolkit [13] using modified Kneser-Ney discounting. There are no out-of-vocabulary words. The details are given in Table 2.

We have developed a German HPSG grammar. The grammar is largely based on the one proposed by [14], which covers a large range of linguistic phenomena. The semantic component of HPSG was not taken into account, as we are only concerned with the grammaticality of utterances. We added some nominal phrase constructions which occur frequently in real-world data, such as prenominal and postnominal genitives, and expressions of quantity. There are a number of rather specialized constructions whose relevance largely depends on the given domain, such as expressions of date and time, forms of address and numerical expressions. As it would have been to big an effort to incorporate all of these, we have restricted ourselves to a few constructions which we observed to be most relevant for our experimental data. Therefore, our current grammar is slightly tailored to the given task. However, the grammar covers many more phenomena than those which actually occur in the experimental data. Out of the 300 sentences in our task 278 (93%) are accepted by the grammar.

<sup>1</sup>This is valid because the parameters optimized on the test data are part of the baseline system and not of our extension.

	WER	$\Delta$ WER
– 4-gram	7.24	
baseline	5.87	–19.0%
baseline	5.87	
+ parsing	4.28	–27.0%

**Table 3.** Word error rates in percent for different language models: the baseline system (2-gram during decoding and 4-gram lattice rescoring), the baseline without 4-gram rescoring and the enhanced system.  $\Delta$ WER denotes the relative improvement.

### 5.3. Results and Discussion

The relative reduction of the word error rate due to the parser is 27.0%. We tested the statistical significance of our results with the Matched Pairs Sentence-Segment Word Error Test (MAPSSWE) and the McNemar Test on sentence level [15]. The improvement over the baseline is significant on a 0.001 level for both tests. The detailed results are given in Table 3.

We have carried out further experiments (not shown in Table 3) to explore the influence of the baseline word error rate on the relative improvement. The results indicate that the relative improvement increases with the quality of the input word lattices: The lower the baseline word error rate, the higher the relative improvement. For a setup with only 30 seconds of adaptation data the baseline word error rate of 17.3% could be reduced by 20.2% relative. A speaker-dependent system which was trained on 7 hours of speech has been tested on the same task in a similar experimental setup. Even though the word error rate of the baseline system was less than 2%, about one third of the errors were corrected.

## 6. CONCLUSIONS

We have presented an approach of incorporating a non-probabilistic grammar into large vocabulary continuous speech recognition. We have shown that this approach is both feasible and advantageous: we have gained a statistically significant improvement of the word error rate compared to a baseline system with cross-word triphones and a 4-gram language model.

These results were produced for a task which is rather forthcoming in that it consists of grammatical sentences most of which are accepted by our grammar. As the presented results are convincing, we have turned to a more difficult real-world task, namely broadcast news transcription.

## 7. ACKNOWLEDGMENT

This work was partly supported by the Swiss authorities in the framework of COST 278 and by the Swiss National Science Foundation in NCCR IM2.

## 8. REFERENCES

- [1] E. Brill, R. Florian, J. Henderson, and L. Mangu, “Beyond N-grams: can linguistic sophistication improve language modeling?,” in *COLING/ACL 1998*.
- [2] Charniak E., “Immediate-head parsing for language models,” in *ACL 2001*, pp. 116–123.
- [3] P. Xu, C. Chelba, and F. Jelinek, “A study on richer syntactic dependencies for structured language modeling,” in *ACL 2002*, pp. 191–198.
- [4] B. Roark, *Robust probabilistic predictive syntactic processing*, Ph.D. thesis, Brown University, 2001.
- [5] M. Collins, *Head-driven statistical models for natural language parsing*, Ph.D. thesis, University of Pennsylvania, 1999.
- [6] G. van Noord, et al., “Robust grammatical analysis for spoken dialogue systems,” *Natural Language Engineering*, vol. 5, no. 1, pp. 45–93, 1999.
- [7] B. Kiefer, et al., “Efficient and robust parsing of word hypotheses graphs,” in *VerbMobil. Foundations of Speech-to-Speech Translation*, W. Wahlster, Ed., pp. 280–295. Springer, Berlin, Germany, artificial intelligence edition, 2000.
- [8] C. J. Pollard and I. A. Sag, *Information-based syntax and semantics, Vol. 1*, Number 13 in CSLI Lecture Notes. CSLI Publications, Stanford University, 1987, Distributed by University of Chicago Press.
- [9] S. Müller, “The Babel-system – an HPSG prolog implementation,” in *Proceedings of the Fourth International Conference on the Practical Application of Prolog*, London, 1996, pp. 263–277.
- [10] I. Müller, *OKiDOKi die Lernhilfe*, Schroedel, 2001.
- [11] J. A. Nelder and R. Mead, “A simplex method for function minimization,” *Computer-Journal*, vol. 7, pp. 308–313, 1965.
- [12] Cambridge University (S. Young et al.), *HTK V3.2.1: Hidden Markov Toolkit*, <http://htk.eng.cam.ac.uk>.

- [13] A. Stolcke, *SRILM – The SRI language modeling toolkit*, SRI Speech Technology and Research Laboratory, <http://www.speech.sri.com/projects/srilm>.
- [14] S. Müller, *Deutsche Syntax deklarativ: Head-Driven Phrase Structure Grammar für das Deutsche*, Niemeyer, 1999.
- [15] L. Gillick and S. Cox, “Some statistical issues in the comparison of speech recognition algorithms,” in *ICASSP*, 1989, pp. 532–535.