USING RULE-BASED KNOWLEDGE TO IMPROVE LVCSR

René Beutler, Tobias Kaufmann and Beat Pfister

ETH Zurich, Switzerland
Computer Engineering and Networks Laboratory
Speech Processing Group
{beutler,kaufmann,pfister}@tik.ee.ethz.ch

ABSTRACT

We show that an elaborate linguistic model of a natural language can be a valuable knowledge source to improve large vocabulary continuous speech recognition (LVCSR). Our approach is to complement a statistical language model with rule-based linguistic knowledge. A hidden Markov model based recognizer and an N-gram language model are used to compute a word lattice which is subsequently processed by a parser. We succeeded in enhancing recognition performance by favouring word sequences which the parser identified as being grammatically correct.

1. INTRODUCTION

To incorporate knowledge about the structure of language above word level, most speech recognizers use simple word 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.

Due to this inadequacy, N-grams tend to perform worse on German than on English. German has a relatively free word order, a rich morphology and agreement of case, number and gender. Whether a word or word form is likely to occur at a given position in a sentence will often depend on words which can be located at almost arbitrary positions. Such dependencies are more adequately modeled by grammar rules than by N-grams. We therefore argue that, at least for German, rule-based knowledge should be incorporated into speech recognition. Last but not least such an approach is supported by the fact that sophisticated and computationally manageable grammars have recently evolved in the natural language processing community.

Attempts to incorporate rule-based information have been made mostly in the areas of dialog systems and natural language understanding. Approaches similar to the one described here have been followed in [1, 2], namely the robust parsing of word graphs. However, these projects aim at increasing semantic accuracy while our goal is to increase word accuracy.

In this paper we will describe a method of incorporating linguistic knowledge into a speech recognizer. Further we will give evidence that the use of a state-of-the-art grammar formalism can reduce the word error rate (WER). Our basic assumption is that the utterances to be recognized are grammatical to a sufficient degree, which enables us to decrease word error rate by favouring grammatical phrases. This assumption holds particularly well for dictation systems, which is why we chose a German dictation system as the scenario for our experiments. However, we do not require that the utterances to be recognized are covered by the grammar. Rather, the recognizer is supposed to be able to cope with ungrammatical and unparsable utterances.

Sections 2 and 3 describe how we integrate syntactic knowledge into our recognition system. The grammar is discussed in Section 4. We report and discuss our results in Section 5 followed by the conclusions in Section 6.

2. ARCHITECTURE

We use an architecture which is frequently used in natural language understanding systems: a word lattice serves as an interface between an acoustic recognizer and a natural language processing module. In our approach, a score derived from the syntactic structures found by the parser is used to rescore the word lattice such that grammatical phrases are slightly favoured.

Initially, the word lattice is produced by a hidden Markov model (HMM) recognizer with a statistical language model. As such lattices can be very large and parsing is an expensive operation, several measures are taken to keep the computational complexity within reasonable bounds. During decoding the size of the word lattice is controlled by appropriately setting the beam search parameters. Once the decoding is complete the lattice size is reduced by forward-backward posterior pruning. In the resulting lat-

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.

tice, the same word sequence can be represented by multiple paths with different acoustic scores due to the uncertainty of word boundaries in continuous speech. 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 more efficiently processed by the parser.

To guide the parser to process the most promising hypotheses first, a stack decoder extracts the N best paths from the recognizer lattice and maps them to the word graph. The mapped paths are processed step by step. Initially, each node in the word graph is marked as inactive. To process a path, all its nodes are activated. A bottom-up chart parser produces all phrases which can be derived from the active nodes in the word graph. The phrases which have already been derived in previous parsing steps are reused for efficiency reasons. This incremental parsing procedure ends if all N paths are processed or if a timeout occurs.

3. SCORING SYNTACTIC STRUCTURES

The recognizer's aim is to find the word sequence $\hat{W}$ which was most likely uttered given the acoustic observation O:

$$\hat{W} \;=\; \arg\max_{W} P(O|W) \cdot P(W) \tag{1}$$

This is called the *maximum a posteriori* (MAP) criterion. However, in practical applications the acoustic likelihood $P(O|W)$ and the language model probability $P(W)$ have to be balanced to optimize the performance:

$$\hat{W} \;=\; \arg\max_{W} P(O|W) \cdot P(W)^{\lambda} \cdot |W|^{ip} \tag{2}$$

λ is the language model weight and ip is the word insertion penalty. We extend the MAP criterion further with an additional parsing score which allows us to slightly favour grammatical utterances:

$$\hat{W} \;=\; \arg\max_{W} P(O|W) \cdot P(W)^{\lambda} \cdot |W|^{ip} \cdot f(W) \tag{3}$$

In the remainder of this section we explain how such a score can be computed. Let W be a word sequence in the lattice spanning the whole utterance. W can be decomposed into a sequence $U = \langle u_1, u_2, \ldots, u_n \rangle$ of so-called parsing units u_i. A parsing unit u_i represents the word sequence $w(u_i)$ which the parser identified as being grammatically correct. The decomposition is such that the concatenation $w(u_1) \circ w(u_2) \circ \cdots \circ w(u_n) = W$. Note that for most W there 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 the 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:

$$s(u, W) \;=\; \begin{cases} c_\alpha & \text{if } w(u) = W, \\ c_\beta & \text{if } 1 < |w(u)| < |W|, \\ c_\gamma & \text{else} \end{cases} \tag{4}$$

where c_α, c_β, and c_γ denote the scores for utterance units, fragment units and single word units, respectively. The score of a decomposition of W is

$$g(\langle u_1, \ldots, u_n \rangle, W) \;=\; \sum_{i=1}^{n} s(u_i, W) \tag{5}$$

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

$$f(W) \;=\; \max_{U} g(U, W) \tag{6}$$

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 unparsable sentences is superfluous.

The parameters λ, ip, c_α, c_β and c_γ are optimized to minimize the empirical word error rate (cf. Section 5.1).

4. GRAMMAR

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 unparsable word sequences, there is no need to artificially weaken the grammar rules.

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 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.

We decided to use the *head-driven phrase structure grammar* (HPSG) formalism. HPSG is unification-based and was developed by [3] to describe natural languages.

It uses linguistically motivated abstractions which substantially simplify the task of writing precise large-scale grammars. For instance, constituent dependence (immediate dominance) and constituent order (linear precedence) are described by two separate sets of rules. This is particularly convenient for modeling languages with relatively free word order such as German. Further, HPSG allows to precisely define the valency structures of verbs.

A practical reason for our choice was that existing systems like [4] demonstrate that substantial fragments of the German language can be modeled in HPSG, and that efficient HPSG parsers can be implemented.

We have developed a German grammar which is largely based on the one proposed by [5]. We omitted some rather special phenomena and added syntactic constructions observed in the experimental data (e.g. the genitive attribute and expressions of quantity). The semantical component of HPSG was discarded as we only focus on grammaticality.

5. EXPERIMENTS

5.1. Training

Continuous density HMMs have been trained by means of HTK [6] with 7 hours of continuous German speech of a single male speaker in an office environment with low background noise and a headset microphone sampled at 16 kHz. The 39-dimensional feature vector consists of 13 mel-frequency cepstral coefficients (MFCCs) including the 0th coefficient, the delta and the delta-delta coefficients. The HMMs are three state left-to-right models with 8 or 32 Gaussian mixtures per state. For each of the 40 phonemes a context-independent monophone model was trained (called mono_8 and mono_32). Context-dependent cross-word triphone models were trained as well (tri_8 and tri_32). The states have been tied using a decision-tree based clustering according to yes/no questions regarding phonetic context, resulting in 3355 triphone models.

N-grams serve as statistical language models (LM). The N-gram probabilities were estimated with the SRI language modeling toolkit [7] on a 50 million words text corpus (German newspaper text and literature) using Good-Turing discounting. The N-grams were estimated for a recognizer vocabulary of 7k words. There are no out-of-vocabulary words.

For the recognition experiments we recorded the first 300 sentences of an exercise book containing dictation texts for pupils in their third year of education [8]. These sentences were partitioned into a development set (200 sentences, 1255 words) and a test set (100 sentences, 637 words). Although these sentences are rather simple, they comprise a wide variety of grammatical constructions, including verbal complexes with up to three verbs, prefix verbs, coordination of nominal and verbal projections, extraposition and genitive attributes. On the test set, the task perplexity is 339.5 for the bigram LM and 274.9 for the trigram LM. The sentences in the test set do neither occur in the acoustic training corpus nor in the text corpus used for the estimation of the language models.

The lexicon contains a basic set of closed-class words and those open-class words occurring in the development and test set, which amounts to about 7'000 full word forms in total. The open-class words include about 200 verbs, 340 nouns and 90 adjectives. For each verb, the possible valency structures (800 in total) have been determined using several sources independent of test set and development set [9, 10].

The parameters λ, ip, c_α, c_β and c_γ introduced in Section 3 are optimized on the development set to minimize the empirical word error rate. Because the word error rate 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) is applied instead [11].

The development set was also used to manually choose the beam search and posterior pruning parameters. The parameters were set to values which substantially reduce the lattice sizes and at the same time yield a reasonably high lattice accuracy. All optimizations are done for each HMM set (mono_8, mono_32, tri_8, tri_32) individually.

5.2. Testing

The HTK decoder performs a time synchronous Viterbi beam search storing the 5 best tokens per HMM state. The recognition network is a back-off bigram word-loop. The resulting lattices are rescored with the trigram language model and posterior pruning is applied. The 100 best scored recognizer lattice paths are processed by the incremental parser as described in Section 2. The parsing timeout was set to one minute on a 1 GHz UltraSPARC IIIi processor. Finally, the optimal word sequence is extracted by combining acoustic, language model and parsing scores.

5.3. Results and discussion

The relative reduction of the word error rate using the parser in addition to the trigram language model was 48.6% in the best case and 28.9% in the worst case. The detailed results are given in Tables 1 and 2. The word error rate was consistently decreased for all acoustic models. Despite of that, not all improvements are statistically significant. We are currently expanding the size of our test and development sets to tackle this issue.

Our basic assumption was that the utterances to be recognized have to be grammatical to a sufficient degree. This assumption holds well for our experiment since most sentences in the test and development sets are covered by our

WER	mono_8	mono_32	tri_8	tri_32
no LM	19.94	13.97	10.68	8.79
+ bigram	7.22	5.65	3.61	2.35
+ trigram	5.97	5.81	2.35	1.88
+ parsing	4.24	2.98	1.57	1.26

Table 1. *Word error rates in percent measured for different acoustic models and language models.*

model	rel. ΔWER
mono_8	-28.9%
mono_32	-48.6%
tri_8	-33.3%
tri_32	-33.3%

Table 2. *Relative reduction of the word error rate on the test set due to extending the MAP criterion with a parsing score.*

grammar. Yet there is still room for improving recognition performance: the parser sometimes does not arrive at processing the correct utterance because parsing is stopped due to a timeout. Timeouts are quite frequent, as parsing efficiency is still a major problem. We expect to decrease the word error rate further by improving the performance of the linguistic subsystem.

However, sometimes the criterion of grammatical correctness is not discriminative enough. Word lattices often contain several grammatically sound utterances. If a grammatically correct utterance is ranked before the correct utterance, the latter will not be chosen. For example, the acoustic recognizer sometimes confuses different verb tenses, which usually does not affect grammaticality.

6. CONCLUSIONS

We have given evidence that rule-based knowledge capturing the structure of natural language can be a valuable information source complementary to N-grams. The additional score derived from the syntactic structures considerably decreased the word error for all acoustic models. To our knowledge, a comparable reduction of the word error rate due to applying a parser has not yet been reported for a similar task.

7. REFERENCES

[1] G. van Noord, G. Bouma, R. Koeling, and M-J. Nederhof, "Robust grammatical analysis for spoken dialogue systems," *Natural Language Engineering*, vol. 5, no. 1, pp. 45–93, 1999.

[2] B. Kiefer, H.-U. Krieger, and M.-J. Nederhof, "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.

[3] 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.

[4] 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.

[5] S. Müller, *Deutsche Syntax deklarativ: Head-Driven Phrase Structure Grammar für das Deutsche*, Niemeyer, 1999.

[6] Cambridge University (S. Young et al.), *HTK V3.2.1: Hidden Markov Toolkit*, `http://htk.eng.cam.ac.uk`.

[7] A. Stolcke, *SRILM – The SRI Language Modeling Toolkit*, SRI Speech Technology and Research Laboratory, `http://www.speech.sri.com/projects/srilm`.

[8] I. Müller, *OKiDOKi die Lernhilfe*, Schroedel, 2001.

[9] *Duden "Das grosse Wörterbuch der deutschen Sprache in 10 Bänden"*, 3. Auflage, Mannheim, Leipzig, Wien, Zürich, 1999.

[10] U. Engel and H. Schumacher, "Kleines Valenzlexikon deutscher Verben," Gunter Narr Verlag Tübingen, 1976, Forschungsberichte des Instituts für deutsche Sprache, Nummer 31.

[11] J. A. Nelder and R. Mead, "A simplex method for function minimization," *Computer-Journal*, vol. 7, pp. 308–313, 1965.