RECENT IMPROVEMENTS OF THE RWTH LARGE VOCABULARY SPEECH RECOGNITION SYSTEM ON SPONTANEOUS SPEECH

Achim Sixtus, Sirko Molau, Stephan Kanthak, Ralf Schlüter, and Hermann Ney

Lehrstuhl für Informatik VI, RWTH Aachen
University of Technology, 52056 Aachen, Germany
{sixtus,molau,kanthak,schlueter,ney}@informatik.rwth-aachen.de

ABSTRACT

This paper presents recent improvements of the RWTH large vocabulary continuous speech recognition system (LVCSR). In particular, we will report on the integration of across-word models into the first recognition pass, and describe better algorithms for fast vocal tract normalization (VTN). We will focus both on the improvements in word error rate and how to speed up the recognizer with only minimal loss in recognition accuracy. Implementation details and experimental results are given for the VerbMobil task, a German spontaneous speech corpus. The 25.0% word error rate (WER) of our within-word baseline system was reduced to 21.4% with VTN and across-word models. Decreasing the real-time factor (RTF) by up to 85% resulted in only a small degradation in recognition performance of 2% relative on average.

1. INTRODUCTION

The RWTH LVCSR system is a continuous Gaussian mixture density speech recognition system, which has been described in detail in [6]. The baseline system is a trigram Viterbi decoder using 3-state left-to-right HMM within-word models, a globally pooled variance vector, phonetically tied triphone models, and a linear discriminant analysis. In this paper we report on:

- the integration of across-word models into the word conditioned tree search algorithm of the decoder,
- the development of improved vocal tract normalization (VTN) procedures, and
- efficient real-time decoding.

We investigate the contributions of these methods to the improvements of WER and RTF in detail. Recognition tests were carried out on the German spontaneous speech corpus VerbMobil. The goal of the VerbMobil project is to develop a real-time speech recognition and translation system for a scheduling task [3]. We report recognition tests on the VerbMobil II development corpus dev99b.

2. ACROSS-WORD MODELLING

In this Section we discuss implementation details of our single-pass across-word trigram decoder. The training of the across-word models was presented in [2].

In principle the organization of the search is similar to the approaches described in [2] and [1]. The across-word models are integrated into a word conditioned tree search. In order to consider all possible right across-word contexts at the word ends, the final arcs of the words in the lexical prefix tree have to fan out. In the system described in [2] these fan-out arcs were compiled statically into the tree lexicon. This is prohibitive when working with large vocabularies containing 20,000 words and more. So, similar to the method described in [1], we now treat these fan out arcs in a dynamic way, i.e. we compile a proxy arc for the last arc of each word into the static lexicon. Whenever a proxy arc is to be activated during decoding, the set of fan-out arcs for all possible right contexts is integrated into the search space instead of the proxy arc.

In contrast to the system described in [1], we perform an additional recombination step during the search process. Beside the acoustic recombination within the tree copies and the language model recombination at word boundaries (which has to take the hypothesized right across-word context into account), we *recombine after the first phoneme generation*.

In principle there are two ways of modelling the transition between two words v and w, depending on the length of the pause between them:

1. word transition *with* coarticulation:
   When no or only a short pause occurs between the words, we have the following dependencies between v and w:
   - χ, denoting the right across-word context v ends with, corresponds to the initial phoneme of w.
   - γ, denoting the left across-word context w starts with, corresponds to the final phoneme of v.

   Hence, the set of words following v is constrained to those, whose initial phoneme equals χ.
2. word transition *without* coarticulation:
   When a long pause occurs between v and w it is not reasonable to assume coarticulation, i.e. v does not influence w. There are no constraints for the words that may follow v. This is modelled with a special across-word context \$.

When using triphones, the influence of the predecessor word v on the successor word w is constrained to the initial phoneme of w. Hence, it is possible to recombine the paths of the two possible transitions after the initial phoneme of the successor word, as illustrated in Figure 1.

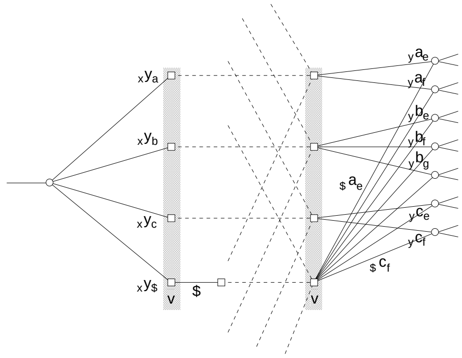

Figure 1: Recombination after the first phoneme generation.

On the left side of Figure 1 the fan-out of the word v with the last phoneme y is shown. a, b, and c are possible right across-word contexts for word transitions with coarticulation, and \$ denotes the context used for transitions without coarticulation. Each fan-out arc of v is recombined with the corresponding fan-out arcs from other tree copies (language model recombination). In the successor tree those arcs are activated, which match the right across-word context χ. The correct left across-word context γ of these arcs depends on v and is determined on the fly. The recombination after the first phoneme generation is performed by joining the arcs of the transitions with and without coarticulation as shown in Figure 1.

To handle this recombination step efficiently, the subtrees with a common predecessor word and different central phonemes are combined in one tree structure. This tree is degenerated in the first generation since it has several roots (one for each possible right across-word context of the predecessor word), and the arcs, that can be recombined, are joined after the first generation. This tree structure inherently takes care of correct handling of χ. So, unlike the systems described in [2] and [10], the tree copies used during the search can be regarded as independent of χ. Like in a conventional within-word system they only depend on the predecessor word.

As we show in Section 5 a relative improvement in performance of 12 % is gained when using across-word models at the cost of a much higher RTF. Nevertheless, by applying the methods described in the next section, across-word decoding can be accelerated drastically.

3. EFFICIENT REAL-TIME DECODING

In order to achieve real-time, we use the following three standard techniques in our decoder: language model look-ahead [7], phoneme look-ahead [8], and fast likelihood calculation of continuous mixture densities [9][4].

3.1. Enhanced Language Model Look-Ahead

For more efficient pruning during beam search, we smear the language model probabilities over the tree using language model look-ahead. As described in [1] and [10] the language model look-ahead can be extended when using across-word models. By hypothesizing a word end with a particular right across-word context, the set of possible successor words is constrained as described above. By incorporating the language model score of the least probable successor word into the accumulated score when activating a fan-out arc with a given right across-word context, a further pruning step can be performed. This so-called *fan-out pruning* reduces the computational effort by a factor of two.

We also applied a similar pruning to the within-word decoder by incorporating the language model score of the least probable successor word into the accumulated score *before* starting a new tree. This makes language model pruning more efficient.

3.2. Phoneme Look-Ahead

To incorporate knowledge of future arc hypotheses into the pruning as early as possible we estimate the acoustic probabilities of each arc beforehand using simple monophone models and perform a pruning step before starting the arc. This method has been described in detail in [8].

3.3. Fast Likelihood Calculation

Without acceleration, the calculations of the likelihood of continuous mixture densities made up more than 80% of the time required for the whole decoding process. We obtained considerable improvements in speed by quantizing the components of the acoustic means and input feature vectors in order to calculate the vector distances using SIMD instructions (e.g. MMX on Intel Pentium, VIS on Sun Sparc). As reported in [4], this leads to a speedup by more than a factor of two without any loss in recognition performance. Additionally, we accelerated the likelihood calculations by using projection search and the preselection VQ method as described in [9].

In the case of the within-word decoder we achieved an overall speed-up by a factor of 4.5. The across-word decoder was accelerated by a factor of 7.3. As can be seen in Section 5 the loss in performance was small.

4. VOCAL TRACT NORMALIZATION

The idea of vocal tract normalization is to remove speaker dependencies caused by variations in the length of the vocal tract by a piecewise linear warping of the frequency axis.

We will first describe the baseline *two-pass VTN* and then discuss improvements of the single mixture based approach for fast warping factor estimation (*fast VTN*).

4.1. Two-Pass VTN

The baseline algorithm has been described in detail in [11]: In speaker-adaptive training, the warping factor for each training speaker is required. Suppose λ is a set of HMM models, X_i^α is a set of feature vectors obtained by applying the warping factor α to the utterance i, and W_i is the corresponding transcription. Then, the optimal warping factor $\hat{\alpha}_i$ is obtained by maximizing the likelihood of each utterances with respect to λ:

$$\hat{\alpha}_i = \arg\max_{\alpha} Pr(X_i^\alpha | \lambda, W_i).$$

Normalized references $\bar{\Lambda}$ are obtained by the standard training procedure on feature vectors which are normalized with the corresponding speaker dependent warping factor [11].

In recognition a similar procedure is applied. Since the spoken utterance W_i is unknown, a preliminary transcription $\hat{W}_i$ is obtained in a first recognition pass with unnormalized feature vectors and parameters Λ. Then the factor $\hat{\alpha_i}$ is determined which maximizes the likelihood with respect to the fully trained normalized acoustic model $\bar{\Lambda}$. Using $\hat{W}_i$ instead of the actually spoken, but unknown transcription W_i does not degrade the recognition performance even if the preliminary transcription has a large word error rate in the order of 20 to 30% [11]. Finally, the feature vectors are warped using $\hat{\alpha_i}$ and a second recognition pass is performed using normalized references $\bar{\Lambda}$. As can be seen in Section 5, we gain a significant performance improvement at the cost of more than doubling the real-time factor due to the two recognition passes.

4.2. Fast VTN

For real-time performance, a two pass recognition approach is virtually prohibitive. To estimate the warping factor of the test speaker without a preliminary recognition pass, a mixture based approach has been suggested in [5]. The approach used in the RWTH LVCSR so far was based on a global single mixture model λ of Gaussian densities which represents the feature vector distribution of normalized speech [11]. This model was obtained on the normalized training data.

During recognition each utterance i is warped with 13 different warping factors to obtain a set of feature vectors X_i^α. In the actual decoding pass the warping factor that maximizes the likelihood of the voiced time frames with respect to the global single mixture model λ is applied:

$$\hat{\alpha_i} = \arg\max_\alpha Pr(X_i^\alpha|\lambda).$$

For the decision between silence and non-silence frames a simple heuristic rule is applied: The Gaussian density that is most frequently aligned with the feature vectors of the utterance is defined to be the silence density and does not contribute to the warping factor estimation. This single mixture approach gives less performance gain than the two-pass VTN described earlier, but requires less computations.

We have improved the approach by introducing one single mixture model λ_α for *each* warping factor similar to the procedure described in [5]. The models are trained only on those *un*normalized feature vectors which belong to the corresponding warping factor. Hence, each model λ_α represents the distribution of the feature vectors of a given warping factor class. Since the training data is distributed over different models, only a fraction of the corpus is available for each model. To increase the robustness we tie the variance over all single mixture models. In contrast to the old approach described in [11] the silence density of each model is not determined on each utterance but in advance on the whole training corpus.

In order to determine $\hat{\alpha_i}$ during recognition, the *unnormalized* feature vectors are aligned with the 13 single mixture models:

$$\hat{\alpha_i} = \arg\max_\alpha Pr(X_i|\lambda_\alpha).$$

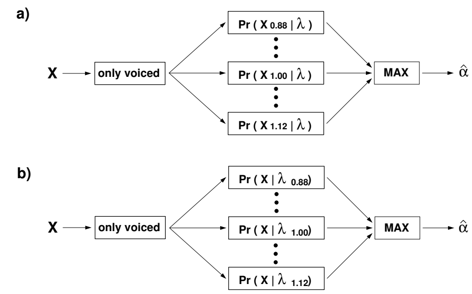

Figure 2: Mixture based warping factor approaches: a) old approach with warped feature vectors X_α and one model λ for *all* warping factors, b) new approach with unwarped feature vectors X and one model λ_α for *each* warping factor.

The vocal tract normalization is significantly accelerated, since the signal analysis has to be carried out only twice (for unwarped feature vectors and for $\hat{\alpha_i}$) instead of 13 times (for each considered warping factor) in case of one global single mixture model.

Figure 2 shows the difference between the old and the new approach. The new multi-mixture VTN performed as good as the baseline two-pass VTN, but was twice as fast. In contrast we did not obtain any improvement with the old fast VTN approach on the VerbMobil corpus compared to the baseline within-word system.

5. EXPERIMENTAL RESULTS

5.1. Test Condition

Experiments were performed for speaker independent recognition of German spontaneous speech with a vocabulary of 10,157 words. The test set, which was recorded for the German VerbMobil II project (development test set of the third integration 1999), consists of 30 minutes of speech, 336 sentences with 4346 spoken words.

The experiments were conducted on a Pentium III 450-MHz PC using the following different recognition setups which are described in this paper:

- baseline within-word (WW)
- within-word + two-pass VTN (WW+2VTN)
- within-word + fast VTN (WW+fVTN)
- across-word (XW)
- across-word + two-pass VTN (XW+2VTN)
- across-word + fast VTN (XW+fVTN)

The baseline recognition system is characterized by:

- recognition lexicon with 10157 phonetic base forms and 137 pronounciation variants;
- 2501 decision tree based triphone states including noise plus one state for silence;
- between 237k (WW+VTN) and 252k (XW+VTN) gender independent Gaussian densities with global pooled diagonal covariance;
- 16 cepstral features with first derivatives and the second derivative of the energy, 10 ms frame shift;
- class-trigram language model with a perplexity of 74.6 on the test set.

The training corpus consisted of 51 hours of speech data from VerbMobil I and II (CDs 1-32).

5.2. Recognition Experiments

In a first set of experiments we focussed on the best word error rate without taking care of the real time factor. As can be seen in Table 1 the relative gain in recognition performance ranges between 8% for the within-word VTN system and 12% for the across-word decoder. Combining VTN and across-word models yields the best result (14% relative improvement in WER) at the cost of increasing the computing time by a factor of 4. This last result is not yet satisfactory and subject of further studies. In all tests the components of the mean and feature vectors were already quantized and the enhanced language model look-ahead was used as described in Section 3.

Table 1: Improvements with across-word models and VTN.

system	search space			word errors [%]		RTF
	states	arcs	trees	del - ins	WER	
WW	3486	1705	25	4.3 - 5.6	25.0	6.8
WW+fVTN	3300	1584	49	5.0 - 4.2	23.4	8.0
WW+2VTN	3264	1566	48	4.9 - 4.3	23.1	17.7
XW	8676	4173	53	4.7 - 3.9	22.1	24.0
XW+fVTN	9190	4428	59	4.0 - 4.1	21.4	25.4
XW+2VTN	9748	4692	62	4.0 - 3.9	21.7	55.1

In a second set of recognition tests we aimed at speeding up the system using the methods described above, while keeping the loss in recognition performance as small as possible. The results are given in Table 2. The within-word system could be accelerated by a factor of 4.5 while the word error rate increased by 0.5%. For the across-word decoder the reduction of RTF was even larger while the increase in word error rate was comparable. Finally, the real-time factor of the combined across-word VTN system decreased by a factor of 6.2 at the cost of a degradation in recognition performance of 0.2% only.

Table 2: Accelerated systems.

system	search space			word errors [%]		RTF
	states	arcs	trees	del - ins	WER	
WW	1669	778	20	4.2 - 5.8	25.5	1.5
WW+fVTN	1733	751	24	4.9 - 4.5	23.7	1.8
XW	4846	1585	28	4.8 - 4.3	22.6	3.3
XW+fVTN	4455	1978	30	4.2 - 4.3	21.6	4.1

6. SUMMARY

We have described implementation details of our single-pass across-word decoder and different VTN approaches. Experiments carried out on the German VerbMobil corpus show significant reductions in word error rate at the cost of a much higher computational effort, however. We have presented acceleration methods that result in a drastic speedup with only a small increase in word error rate.

Acknowledgement. This research was partly funded by grant 01 IV 701 T4 from the German Ministry of Science and Technology (BMBF) as a part of the VerbMobil project. The views and conclusions contained in this document are those of the authors.

7. REFERENCES

[1] X.L. Aubert: One Pass Cross Word Decoding For Large Vocabularies Based on a Lexical Tree Search Organization, Proc. European Conf. on Speech Communication and Technology, pp. 1559-1562, Budapest, Hungary, September 1999.

[2] K. Beulen, S. Ortmanns, C. Elting: Dynamic Programming Search Techniques for Across-Word Modelling in Speech Recognition, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 609-612, Phoenix, AZ, March 1999.

[3] T. Bub, J. Schwinn: VERBMOBIL: The Evolution of a Complex Large Speech-to-Speech Translation System, Proc. Int. Conf. on Spoken Language Processing, pp. 2371-2374, Philadelphia, PA, October 1996.

[4] S. Kanthak, K. Schütz, H. Ney: Using SIMD Instructions for Fast Likelihood Calculation in LVCSR, submitted to IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Istanbul, Turkey, May 2000.

[5] L. Lee, R. Rose: Speaker Normalization using Efficient Frequency Warping Procedures, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 353-356, Atlanta, GA, May 1996.

[6] H. Ney, L. Welling, S. Ortmanns, K. Beulen, F. Wessel: The RWTH Large Vocabulary Continuous Speech Recognition System, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 853-856, Seattle, WA, May 1998.

[7] S. Ortmanns, H. Ney, A. Eiden: Language-Model Look-Ahead for Large Vocabulary Speech Recognition, Proc. Int. Conf. of Spoken Language Processing, pp. 2091-2094, Philadelphia, PA, October 1996.

[8] S. Ortmanns, H. Ney, A. Eiden, N. Coenen: Look-Ahead Techniques for Improved Beam Search, Proc. CRIM-FORWISS Workshop, pp. 10-22, Montreal, October 1996.

[9] S. Ortmanns, H. Ney, T. Firzlaff: Fast Likelihood Computation Methods for Continuous Mixture Densities in Large Vocabulary Speech Recognition, Proc. European Conf. on Speech Communication and Technology, pp. 139-142, Rhodos, Greece, September 1997.

[10] S. Ortmanns, W. Reichl, W. Chou: An Efficient Decoding Method for Real Time Speech Recognition, Proc. European Conf. on Speech Communication and Technology, pp.499-502, Budapest, Hungary, September 1999.

[11] L. Welling, S. Kanthak, H. Ney: Improved Methods for Vocal Tract Normalization, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, pp. 761-764, Phoenix, AZ, March 1999.