Fast Search for Large Vocabulary Speech Recognition

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

Lehrstuhl für Informatik VI, Computer Science Department,
RWTH Aachen - University of Technology, Germany

Abstract. In this article we describe methods for improving the RWTH German speech recognizer used within the VERBMOBIL project. In particular, we present acceleration methods for the search based on both within-word and across-word phoneme models. We also study incremental methods to reduce the response time of the online speech recognizer. Finally, we present experimental off-line results for the three VERBMOBIL scenarios. We report on word error rates and real-time factors for both speaker independent and speaker dependent recognition.

1 Introduction

The goal of the VERBMOBIL project is to develop a speech-to-speech translation system that performs close to real-time. In this system, speech recognition is followed by subsequent VERBMOBIL modules (like syntactic analysis and translation) which depend on the recognition result. Therefore, in this application it is particularly important to keep the recognition time as short as possible. There are VERBMOBIL modules which are capable to work with partial results. For these modules, it is also desirable to have an incremental recognition output.

The RWTH LVCSR system is a continuous Gaussian mixture density speech recognition system which has been described in detail by Ney et al. (1998). In this paper we report in detail on:

- acceleration methods for within-word recognition (Section 2),
- search and acceleration methods for across-word recognition (Section 3),
- acceleration methods for vocal tract normalization (Section 4),
- incremental processing methods to reduce the response time (Section 5),

For these methods we report experimental results in terms of improvements in word error rate and real-time (see Section 6). Recognition tests were carried out on the VERBMOBIL German spontaneous speech development corpus dev99 (scenarios A and B: scheduling of appointments and hotel/travel reservations, speaker independent recognition). In addition, we present experimental results on the speaker dependent VERBMOBIL PC remote maintenance task (scenario C) in Section 7.

* We would like to thank Andreas Eiden, Stefan Ortmanns, and Lutz Welling for their initial work on our speech recognizer in the VERBMOBIL project.

2 Baseline System

In this section, we review the techniques used to increase the speed of the integrated one-pass tree-organized time-synchronous beam search algorithm (Ney et al., 1998):

- hypothesis pruning,
- language model look-ahead,
- phoneme look-ahead,
- fast likelihood computation of continuous mixture densities.

2.1 Hypothesis Pruning

During search we use conventional beam pruning and histogram pruning (e.g. Ortmanns et al., 1997a) at state and word level. We found that especially histogram pruning at the word level was capable to reduce the computational effort considerably. The upper bound of the number of word end hypotheses per time frame was decreased to roughly 20 without any increase in word error rate. The gain in real-time factor was 15–25%.

2.2 Enhanced Language Model Look-Ahead

For more efficient pruning during beam search, we distribute the language model probabilities over the tree using language model look-ahead (Ortmanns et al., 1996a). We enhanced this look-ahead by incorporating the language model score of the most probable successor word into the accumulated score *before* starting a new tree. This makes language model pruning more efficient. Language model look-ahead can be further extended when using across-word models (see Section 3).

2.3 Phoneme Look-Ahead

The idea of phoneme look-ahead is to anticipate the probability of future acoustic vectors for a phoneme arc before starting it in detailed search. To estimate the acoustic probabilities, we use simplified context independent models and perform an additional pruning step before starting the phoneme arcs. This method is described in detail in Ortmanns et al. (1996b) and results in an overall speedup by a factor of about 2.

2.4 Fast Likelihood Calculation

Without acceleration, the calculations of likelihoods of the continuous mixture densities make up more than 80% of the time required for the whole recognition process. We obtained considerable improvements in speed by quantizing the components of the acoustic means and input vectors in order to calculate the vector distances using parallelizing SIMD instructions (e.g. MMX on Intel Pentium, VIS on Sun Sparc).

As reported in Kanthak et al. (2000), this leads to an overall speedup of the recognizer by more than a factor of three without any loss in recognition performance. Additionally, we accelerated the likelihood calculations by using projection search and the preselection VQ method described in Ortmanns et al. (1997b).

Using all these methods, the baseline within-word recognizer as well as the across-word recognizer can be sped up by more than a factor of 10. As will be seen in Section 6, the loss in word error rate remains small (e.g. 2 % relative).

3 Across-Word Models and Search

It is well known (Alleva et al., 1992; Woodland et al., 1994; Beyerlein et al., 1997) that the word accuracy can be improved significantly by the use of across-word phoneme models.

In addition to within-word contexts, the across-word models capture also triphone dependencies across word boundaries. The triphone contexts across word boundaries cannot be determined from the pronunciation lexicon alone, they also depend on the surrounding words hypothesized during search. This drastically increases the complexity of the search.

In this section, we briefly describe the principle of the RWTH one-pass across-word recognizer and discuss methods for coping with the increase in complexity. The training of the across-word models is presented in Beulen et al. (1999).

3.1 Across-Word Search

The search is organized in a similar way as described in Aubert (1999) and Beulen et al. (1999). The across-word models are integrated into a word-conditioned tree search. Since the successor word is not known at a word end hypothesis, all possible right contexts have to be hypothesized in separate fan-out arcs. These fan-out arcs are treated in a dynamic way, i.e. for each word end we compile a proxy arc into the static representation of the lexical prefix tree. Whenever such a word end arc is to be activated during the search the set of fan-out arcs for all possible right contexts is integrated dynamically into the active search space.

Figure 1 shows a word transition between the fan-out of word end w and the first generation of the successor tree. Depending on the hypothesized fan-out arc, we activate those arcs in the successor tree whose central phoneme corresponds to the right across-word context of the fan-out arc. The left across-word context of these arcs corresponds to the central phoneme of the fan-out arcs. We call this kind of transition "word transition *with* coarticulation".

If the silence portion between two adjacent words is longer than a preselected threshold, we assume no coarticulation and therefore do not use the corresponding across-word model. We allow for this case by providing a fan-out arc with a special right context, which is denoted by the symbol \$ in Figure 1. Since there is no dependence assumed between the words, phoneme arcs with all possible central

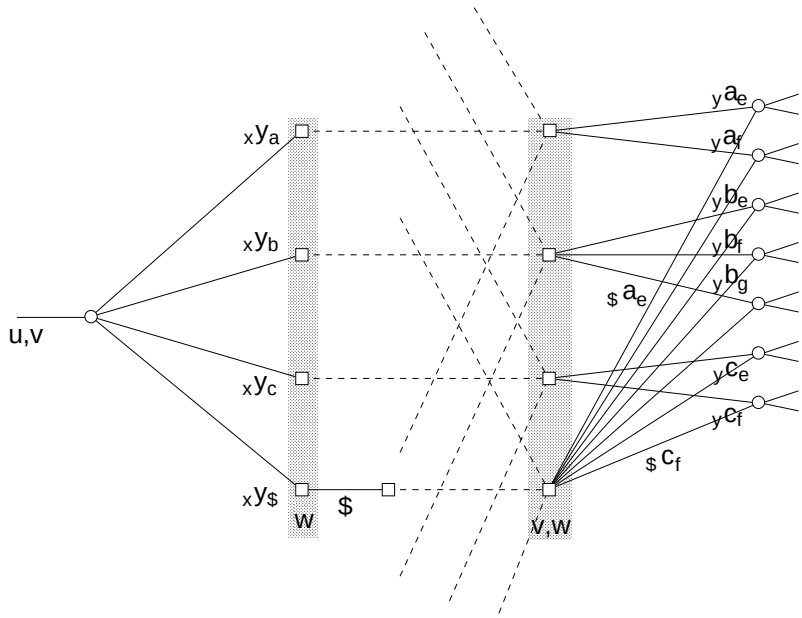

Figure 1. Transition between word end w and the successor tree using across-word modeling: The arcs of the word transition *with* coarticulation are recombined with the arcs of the word transition *without* coarticulation after the first phoneme generation.

phonemes have to be activated in search. In Figure 1 the left contexts of these arcs are denoted by the $ symbol as well.

The word transition affects only the final triphone of the predecessor word and the first triphone of the successor word. Thus, the arcs that were activated via the transition *with* coarticulation can be recombined with the arcs of the transition *without* coarticulation (see Figure 1). More details about this recombination step are given in Sixtus et al. (2000).

3.2 Enhanced Language Model Look-Ahead

As mentioned in Section 2, we use the language model look-ahead to make the pruning during beam search more efficient. As described in Aubert (1999) and Ortmanns et al. (1999), the language model look-ahead can be extended when using across-word models.

The set of successor words that may follow a word end hypothesis w with a particular right across-word context χ is constrained to those words whose first phoneme is χ. When activating the fan-out arc of w with the right context χ, we anticipate the language model score of the most probable successor word and incorporate it into the accumulated score. Then, in addition to the conventional pruning

techniques, *fan-out pruning* can be applied to the fan-out arcs. Since the potential search space is mainly increased by the fan-out arcs (40 to 50 additional arcs per word end) when using across-word models, this additional pruning step is essential for efficient search. Informal experiments have shown that the computational effort can be reduced by a factor of two by applying fan-out pruning.

3.3 Compressing the Lexical Prefix Tree

The acoustic models of the recognizer consist of 3-state HMM triphone models. The states of the triphones are tied using a decision tree. Therefore, there are many triphones which share the same state sequence. When constructing the lexical prefix tree, arcs of acoustically equivalent triphones are merged if they have the same predecessor arc. In particular, the number of fan-out arcs whose triphones differ in the right context only can be reduced substantially by this merging step.

Merging the fan-out arcs results in single fan-out arcs with several corresponding right contexts. Thus, in the following successor tree *all* those subtrees have to be activated that correspond to *each* of the right contexts attached to the fan-out arc. Similarly, the enhanced language model look-ahead has to take care of the merged fan-out arcs since the number of possible successor words of a fan-out arc is increased if there are several contexts attached to this arc.

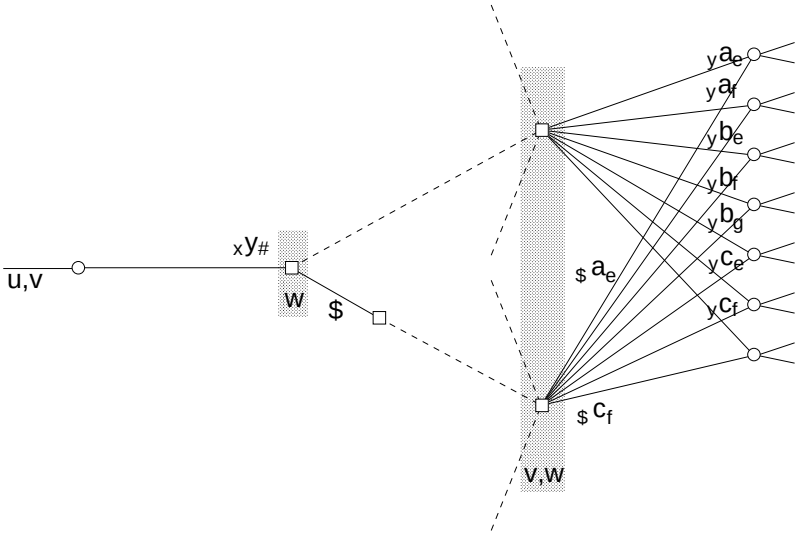

Figure 2. Transition between word end w and the successor tree without regarding the right context.

In the experiments on the VERBMOBIL corpus, we obtained a reduction of the number of fan-out arcs by 45 % which reduced the computational effort by roughly 25 %. We tested a similar compression for the within-word recognizer, but there was no gain in real-time performance.

3.4 Considering Left Across-Word-Contexts Only

The main part of the increase in complexity is caused by the right across-word contexts represented by the fan-out arcs. On the other hand, the left contexts result in a small overhead only, since in a word conditioned tree search the predecessor word and thus its last phoneme is known. We studied how much we loose in recognition performance if we disregard the right across-word contexts and consider the left across-word contexts only. A similar approach is described in Alleva et al. (1992).

Figure 2 shows a word transition, where only the left context of the successor tree is considered. The right context of the predecessor word w is represented by a dummy symbol $\#$. As can be seen in Section 6, nearly 40 % of the improvements achieved by across-word models can be retained if only the left across-word contexts are considered.

4 Vocal Tract Normalization (VTN)

The idea of VTN is to remove speaker dependencies caused by variations in the length of the vocal tract by scaling the frequency axis. In this section, we shortly describe the principles of VTN. Then, we discuss a text-independent Gaussian mixture based approach for fast warping factor estimation (*fast VTN*). Finally, we describe the method for incremental estimation of warping factors.

4.1 VTN Principles

The baseline VTN algorithm has been described in detail in Welling et al. (1999). In speaker-adaptive training, the warping factor $\hat{\alpha}_i$ for each training speaker i is obtained by exhaustive search in the range $0.88, \ldots, 1.12$ with step size 0.02:

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

In this equation, the HMM emission probability $Pr(X_i^{\alpha}|\mu, W_i)$ is computed by a Viterbi alignment of the warped acoustic vectors X_i^{α} for the HMM parameter set μ and the spoken word sequence W_i.

So-called normalized references are obtained by training the parameters of the emission distributions on warped acoustic vectors, i.e. the vectors are normalized using the optimal speaker dependent warping factors (Welling et al., 1999).

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 no vocal tract normalization. Then the factor $\hat{\alpha}_i$ is determined which maximizes

the likelihood using $\hat{W}_i$. Finally, the acoustic vectors are warped using $\hat{\alpha_i}$, and a second recognition pass is performed using the normalized references. Recognition tests have shown that a significant performance improvement is obtained by this two-pass method. However, the computation time is more than doubled. 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% (Welling et al., 1999).

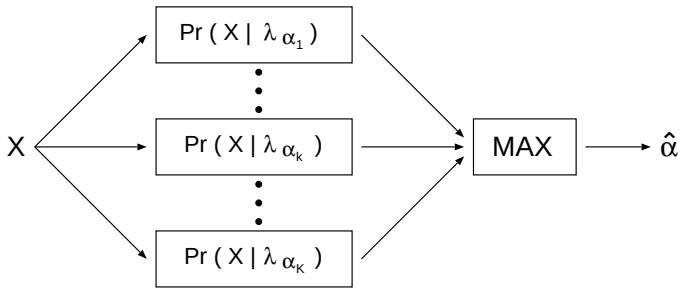

Figure 3. Mixture based warping factor approach using unwarped acoustic vectors X and one single mixture model λ_α for *each* warping factor.

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, Lee et al. (1996) and Welling et al. (1999) suggested a text-independent method using Gaussian mixture models. The approach used here relies on a separate emission distribution $Pr(X|\lambda_\alpha)$ for *each* warping factor α, where X denotes the sequence of acoustic vectors and λ_α denotes the α-dependent distribution parameters. For each α, the parameters λ_α are trained only on those *un*normalized acoustic vectors which are assigned to this warping factor. 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 order to determine the optimal $\hat{\alpha}$ for an observed sequence X of acoustic vectors during recognition, we use the following equation (see Figure 3):

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

Using this approach, the vocal tract normalization is significantly accelerated. The signal analysis has to be carried out twice (unwarped and warped with $\hat{\alpha}$), and there is no need for two recognition passes. Tests have shown that the fast VTN performs almost as well as the baseline two-pass VTN, but is twice as fast (Sixtus et al., 2000).

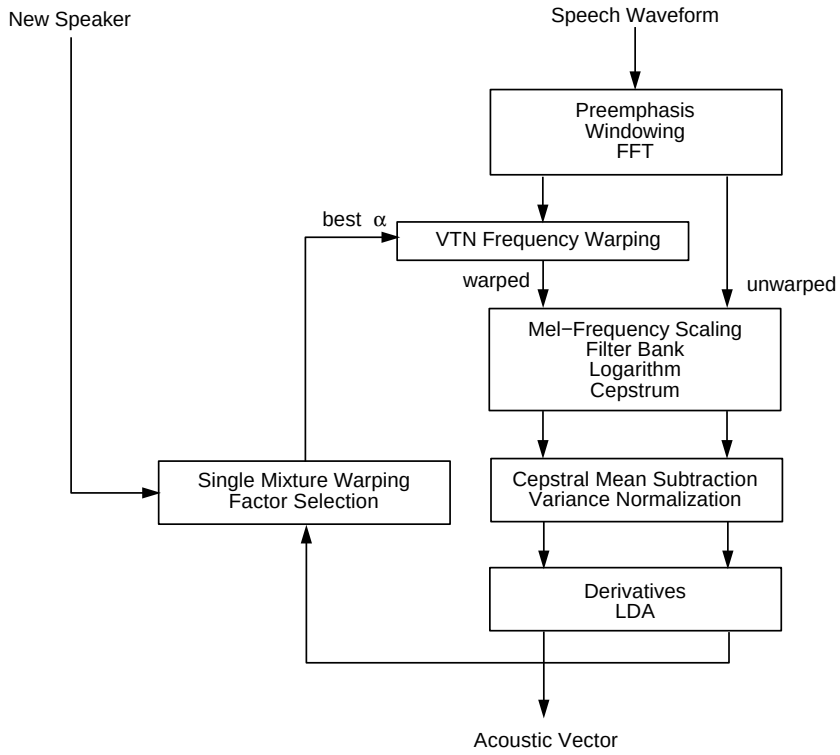

Figure 4. Signal analysis with fast VTN and incremental warping factor estimation without time delay.

4.3 Incremental Estimation of Warping Factors

The time required by VTN can be further improved by reducing the time delay caused by the estimation of the warping factor. For this purpose, we investigated a frame incremental warping factor estimation for fast VTN. As shown in Figure 4, the signal analysis following FFT is always carried out twice. The acoustic vector warped with the currently best warping factor is used for recognition. The unwarped acoustic vector is immediately evaluated with the text-independent mixture models and the scores $Pr(X|\lambda_\alpha), \alpha = 0.88, \ldots, 1.12$ are computed by accumulation over time. As long as not enough time frames have been collected (200 time frames, i.e. 2 seconds of speech), the warping factor is set to 1.0. After a few seconds, the best warping factor does not change anymore.

Tests have shown that incremental warping factor estimation can be carried out without any delay at the cost of only little performance degradation in the first few seconds of a new speakers' utterance.

5 Incremental Processing for Online Recognition

In this section, we focus on integrating the recognizer into the online environment with subsequent processing steps of the VERBMOBIL system. The time delay introduced by the recognizer should be as small as possible since all postprocessing steps depend on the recognition output. To this purpose, we incorporated the following concepts:

- reducing the time delay of the recognizer by performing cepstral mean subtraction not on the whole sentence but on a sliding window with constant length,
- reducing the response time of the system by incrementally preparing recognition results,
- reducing the density of word graphs in order to reduce the processing time of following steps.

The constraint which all these concepts should satisfy is not to increase the word error rate.

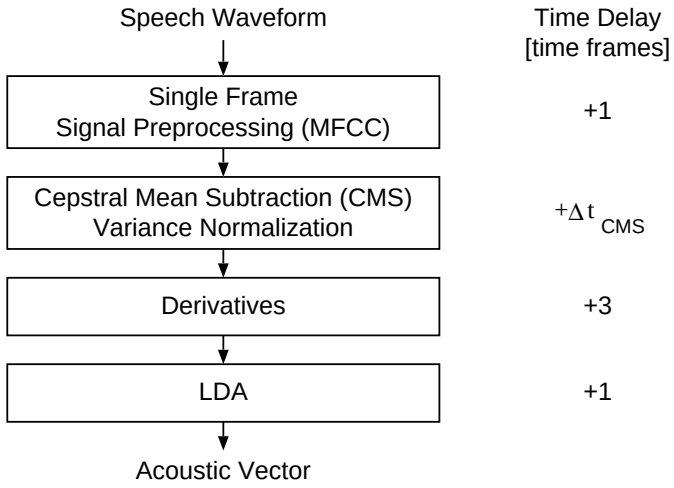

Figure 5. Signal processing steps and associated time delays.

5.1 Recognition Preprocessing

The time delay introduced by the signal analysis and feature extraction of the recognizer is shown in detail in Figure 5. We consider two types of cepstral mean subtraction: the cepstral mean can be either computed on the whole sentence or on a running window. As we will see, the minimum window length Δt_{CMS} of the cepstral mean subtraction is in the order of several hundred 10-ms time frames. Therefore, the cepstral mean subtraction contributes most to the overall time delay.

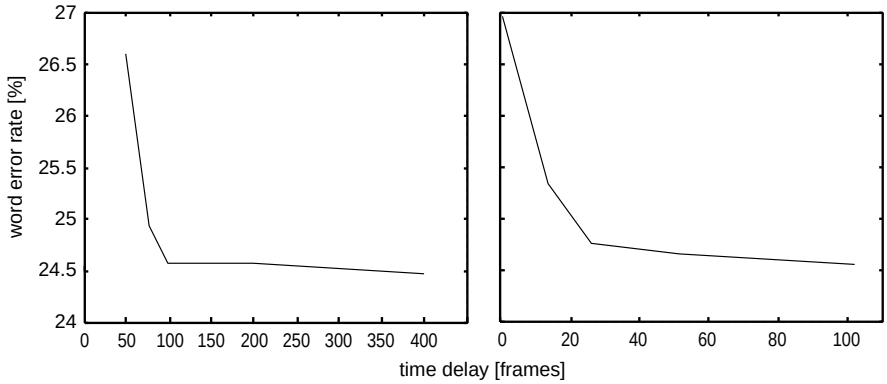

Figure 6. Word error rate as a function of time delay Δt_{CMS} for cepstral mean subtraction using symmetric (left) and asymmetric (right) windows. For the asymmetric case, the window length was kept fixed at 200 time frames.

In several experiments, we found that for the German VERBMOBIL recognition task a symmetric running window of at least 2-seconds length results in exactly the same word error rate as sentence-wise normalization while reducing the time delay to 1 second (Figure 6 left). By using an asymmetric window of the same length, the delay can be further reduced to 250 ms without any loss in recognition performance (Figure 6 right).

5.2 Incremental Recognition Output

The output of the two main recognition results, the single best word sequence and the word graph, is usually delayed until the end of the utterance. Nevertheless, the subsequent modules could already start working if there were partial results available from the recognizer. The approach to output partial first best word sequences and word graphs is based on search space decisions and was inspired by Spohrer et al. (1980).

The idea is to trace back the currently active state hypotheses in order to determine a single word end node shared by all paths. Once we have found such a word end node we can already report the first best word sequence up to this node because in the future it will not change anymore. This partial traceback is performed only after a sufficiently long silence segment has been recognized.

Since the paths of the currently active hypotheses share a single node in the resulting word graph, too, partial word graphs can be generated in a similar fashion. Using this method it is guaranteed that the single best word sequence is still in the word graph.

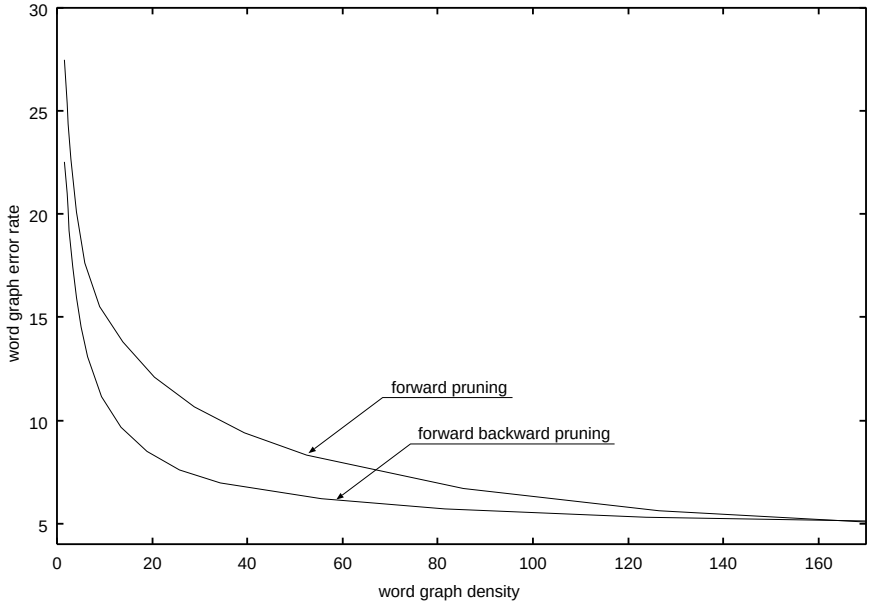

Figure 7. Comparison of forward- and forward-backward word graph pruning. The WER of the single best word sequence was 23.0 %.

5.3 Forward-Backward Word Graph Pruning

To keep the computational effort for the subsequent VERBMOBIL modules as low as possible, it is desirable to generate compact word graphs with low *word graph density* (the total number of word graph arcs divided by the number of spoken words) and low *graph error rate* (the minimum WER of all sentences represented by the word graph) (Ortmanns et al., 1997a).

To generate compact word graphs we apply the *forward-backward pruning* described in detail in Sixtus et al. (1999). This pruning technique is based on the score of the global best path through the entire word graph. For every arc of the graph representing a word hypothesis $(w; \tau, t)$, with word identity w, starting time $\tau + 1$ and ending time t, we compute the overall score $Q(w; \tau, t)$ of the best path passing through the arc. Only arcs with path scores close to the global best path are kept.

This word graph pruning technique is based on the paradigm of the forward-backward algorithm since $Q(w; \tau, t)$ is computed by combining the following two quantities:

1. The *forward score*, which is defined as the overall score of the best partial word sequence that starts at the first time frame of the utterance and ends at time t in word arc $(w; \tau, t)$;

2. The *backward score*, which is defined as the overall score of best partial word sequence that starts at the time frame $\tau + 1$ in word arc $(w; \tau, t)$ and ends at the last time frame of the utterance.

Both quantities are computed by traversing the word graph from left to right (*forward pass*) and from right to left (*backward pass*).

As can be seen in Figure 7, at a given GER, forward-backward pruning leads to much smaller word graphs than the conventional forward pruning algorithm, where only the forward score is used as the pruning criterion (Ortmanns et al., 1997a). Furthermore, using forward pruning the single best word sequence might be pruned from the word graph while it is preserved when using forward-backward pruning.

6 Off-line Recognition Results (Scenario A and B)

6.1 Testing Conditions

All experiments presented in this section were performed on the speaker independent VERBMOBIL German spontaneous speech scenarios A and B. The statistics of the training and testing set are summarized in Table 1. The baseline recognition system is characterized by:

- 16 cepstral coefficients with first derivatives and the second derivative of the energy, 10 ms frame shift;
- linear discriminant analysis on three adjacent vectors resulting in a 33-dimensional acoustic vector;
- 3-state HMM triphone models with skip, forward and loop transitions;
- decision tree with 2,501 tied states including noise and one silence state;
- gender independent Gaussian mixtures with a total of 170k to 240k densities and globally pooled diagonal covariance matrix;
- recognition vocabulary of 10,810 words with additional 136 pronunciation variants in the pronunciation lexicon;
- trigram language model with a testing set perplexity of 62.0.

Table 1. Training (CD 1-41) and testing set (development set of the third integration 1999).

	training	testing
acoustic data	61.5h	1.6h
silence portion	13%	11%
# speakers	857	16
# sentences	36,015	1,081
# running words	701,512	14,662
# running phonemes	2,331,927	60,716
perplexity (trigram LM)	-	62.0

In the experiments to be reported (see Tables 2 and 3), we compare the search space (active states, arcs, trees after pruning), word error rate (WER) and real-time factor (RTF) of the within-word recognizer (WW), the across-word recognizer (XW) and the modified across-word recognizer, considering only the left contexts (XW-light). In addition, we combine the WW system and the XW-light system with the implementation of the fast vocal tract normalization (WW + VTN, XW-light + VTN). All experiments were conducted on a Pentium III 600 MHz PC.

6.2 Recognition Experiments

In a first set of experiments we compare the best performing systems without using phoneme look-ahead, projection search and the preselection VQ method. Note that we already use the SIMD instructions to accelerate the distance calculation as described in Section 2. Results are shown in Table 2. The word error rate can be reduced from 24.6 % to 22.3 % (9.3 % relative) using across-word models instead of within-word models; however, the real-time factor is increased by a factor of 2.8. When disregarding the right contexts of the across-word models, the word error rate is decreased from 24.6 % to 23.7 % (3.7 % relative) only, but the increase in real-time is much smaller.

Table 2. Comparison of the systems optimized for word error rate.

	search space			*word errors [%]*		
system	states	arcs	trees	del - ins	WER	RTF
WW	4173	2022	55	5.3 - 3.8	24.6	6.7
XW-light	4330	2050	36	6.3 - 2.6	23.7	7.8
XW	11118	5289	62	5.5 - 2.8	22.3	18.5
WW + VTN	3710	1806	49	4.8 - 3.6	23.0	5.7
XW-light + VTN	5180	2480	34	5.6 - 2.7	22.3	7.9

Furthermore, combining VTN with within-word models results in an improvement from 24.6 % to 23.0 % (6.5 % relative). Combining VTN with the XW-light system improves the error rate from 24.6 % to 22.3 % (9.3 % relative) as compared to the baseline WW system.

Table 3. Comparison of the accelerated systems.

	search space			*word errors [%]*		
system	states	arcs	trees	del - ins	WER	RTF
WW	1839	827	19	5.8 - 3.9	25.1	1.3
XW-light	2363	1198	21	5.7 - 3.3	24.2	1.6
XW	2915	1445	27	5.4 - 3.0	22.8	2.5
WW + VTN	1771	767	17	5.5 - 3.5	23.5	1.2
XW-light + VTN	2117	1018	23	5.5 - 2.9	22.8	1.5

In a second series of experiments, we accelerate all systems using the methods described in Section 2. Results are shown in Table 3. For each best performing system shown in Table 2, there is an accelerated version in Table 3. The acceleration parameters are chosen in such a way that the word error rate goes up by 0.5 % absolute. Comparing the WW system and the XW system, we see that the WW system is faster by a factor of 2. The XW-light system, however, is only 20 % slower than the WW system.

As can be seen in Table 3, VTN allows for more efficient pruning. Thus, the combined WW + VTN system is fastest and was chosen for integration into the VERBMOBIL prototype system.

7 Off-line Recognition Results (Scenario C)

All experiments presented in this section are performed on the speaker dependent VERBMOBIL PC remote maintenance task scenario C. The idea of this scenario was to investigate rapid adaptation to a new domain, larger vocabulary recognition and speaker dependent recognition. The system uses the within-word recognizer as described in Section 2. The set-up can be summarized as follows:

- 16 cepstral coefficients with first derivatives and the second derivative of the energy, 10ms frame shift;
- speaker independent linear discriminant analysis on three adjacent vectors resulting in a 33-dimensional acoustic vector;
- 6-state HMM triphone models with skip, forward and loop transitions;
- speaker independent decision tree with 1,001 tied states including one silence state;
- speaker dependent Gaussian mixtures with about 7,000 densities and globally pooled diagonal covariance matrix;
- recognition vocabulary of 14,342 words with additional 831 pronunciation variants in the pronunciation lexicon;
- trigram language model with an average testing set perplexity of 108.

For each of the five test speakers the acoustic models are trained on 30 minutes of speech. All recognition experiments are conducted on a Pentium III 600 MHz PC using 15 minutes of speech for each speaker. Language model look-ahead and SIMD instructions are used for acceleration.

Table 4. Recognition performance for all speakers on the scenario C.

	search space			*word errors [%]*		
speaker	states	arcs	trees	del - ins	WER	RTF
CB	2166	584	3	0.3 - 0.3	4.3	0.7
RS	1676	468	3	0.4 - 0.9	4.2	0.6
RK	1960	530	3	0.4 - 0.9	5.5	0.7
UU	2638	703	4	0.6 - 0.5	6.3	0.8
TR	1830	501	3	0.3 - 0.5	3.6	0.6

Results are shown in Table 4. It can be seen that both the word error rate and the real-time performance are comparable to commercial text dictation systems.

During the project the vocabulary has been extended to almost 35,000 words and the domain has been shifted in order to ensure proper language model training. However, the perplexity of the language model raised from about 100 to 300. Informal tests have shown that due to the modified domain the word error rate is between 18% and 20% and not comparable to the results presented above although the speed of the recognizer can mainly be retained.

8 Conclusion

In this paper, we have considered acceleration methods for the RWTH recognizer of the VERBMOBIL system. We have presented several methods to handle the increase of complexity caused by across-word models. When real-time performance is the primary objective, the best trade-off between recognition accuracy and computation time has been found for the within-word system. We have successfully combined the speaker adaptation based on vocal tract normalization with both the within-word recognizer and the XW-light version of the across-word recognizer. We have studied several incremental methods for online speech recognition to reduce the response time of the recognizer. Finally, we have presented experimental results for both speaker independent and speaker dependent VERBMOBIL scenarios.

References

Alleva, F., Hon, H., Huang, X., Hwang, M., Rosenfeld, R., Weide, R. (1992): "Applying SPHINX-II to the DARPA Wall Street Journal CSR Task", In *Proc. of the DARPA Speech and Natural Language Workshop*, pp. 393–398, Harriman, NY, February 1992.

Aubert, X.L. (1999): "One Pass Cross Word Decoding For Large Vocabularies Based on a Lexical Tree Search Organization", In *Proc. of the European Conf. on Speech Communication and Technology*, pp. 1559–1562, Budapest, Hungary, September 1999.

Beulen, K., Ortmanns, S., Elting, C. (1999): "Dynamic Programming Search Techniques for Across-Word Modeling in Speech Recognition", In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 609–612, Phoenix, AZ, March 1999.

Beyerlein, P., Ullrich, M., Wilcox, P. (1997): "Modeling and Decoding of Crossword Context Dependent Phones in the Philips Large Vocabulary Continuous Speech Recognition System", In *Proc. of the European Conf. on Speech Communication and Technology*, pp. 1163–1166, Rhodes, Greece, September 1997.

Kanthak, S., Schütz, K., Ney, N. (2000): "Using SIMD Instructions for Fast Likelihood Calculation in LVCSR", To be published in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, June 2000.

Lee, L., Rose, R. (1996): "Speaker Normalization using Efficient Frequency Warping Procedures", In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 353–356, Atlanta, GA, May 1996.

Ney, H., Welling, L., Ortmanns, S., Beulen, K., Wessel, F. (1998): "The RWTH Large Vocabulary Continuous Speech Recognition System", In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 853–856, Seattle, WA, May 1998.

Ortmanns, S., Ney, H., Eiden, A. (1996): "Language-Model Look-Ahead for Large Vocabulary Speech Recognition", In *Proc. of the Int. Conf. of Spoken Language Processing*, pp. 2091–2094, Philadelphia, PA, October 1996.

Ortmanns, S., Ney, H., Eiden, A., Coenen, N. (1996): "Look-Ahead Techniques for Improved Beam Search", In *Proc. of the CRIM-FORWISS Workshop*, pp. 10–22, Montreal, October 1996.

Ortmanns, S., Ney, N., Aubert, X. (1997): "A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition", In *Computer, Speech and Language*, Vol. 11, No. 1, pp. 43–72, January 1997.

Ortmanns, S., Ney, H., Firzlaff, T.: (1997): "Fast Likelihood Computation Methods for Continuous Mixture Densities in Large Vocabulary Speech Recognition", In *Proc. of the European Conf. on Speech Communication and Technology*, pp. 139–142, Rhodes, Greece, September 1997.

Ortmanns, S., Reichl, W., Chou, W.: "An Efficient Decoding Method for Real Time Speech Recognition", In *Proc. of the European Conf. on Speech Communication and Technology*, pp. 499–502, Budapest, Hungary, September 1999.

Spohrer, J.C., Brown, P.F., Hochschild, P.H., Baker, J.K. (1980): "Partial Traceback in Continuous Speech Recognition", In *Proc. of the Int. Conf. on Cybernetics and Society*, pp. 36–42, Cambridge, MA, October 1980.

Sixtus, A., Ortmanns, S. (1999): "High Quality Word Graphs Using Forward-Backward Pruning", In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, pp. 593–596, Phoenix, AZ, March 1999.

Sixtus, A., Molau, S., Kanthak, S., Schlüter, R., Ney, H. (2000): "Recent Improvements of the RWTH Large Vocabulary Speech Recognition System on Spontaneous Speech", To be published in *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Istanbul, Turkey, June 2000.

Welling, L., Kanthak, S., Ney, H. (1999): "Improved Methods for Vocal Tract Normalization", In *Proc. of the IEEE Int. Conf. on Acoustics Speech and Signal Processing*, pp. 761–764, Phoenix, AZ, March 1999.

Woodland, P.C., Odell, J.J., Valtchev, V., Young, S.J. (1994): "Large Vocabulary Continuous Speech Recognition using HTK", In *Proc. of the IEEE Int. Conf. on Acoustics, Speech and Signal Processing*, Vol. 2, pp. 125–128, Adelaide, Australia, April 1994.