DYNAMIC PROGRAMMING SEARCH TECHNIQUES FOR ACROSS-WORD MODELLING IN SPEECH RECOGNITION

Klaus Beulen [1], *Stefan Ortmanns*[2] *and Christian Elting*[1]

[1] Lehrstuhl für Informatik VI, RWTH Aachen – University of Technology, 52056 Aachen, Germany

[2] Lucent Technologies – Bell Labs., Murray Hill, NJ 07974, USA

ABSTRACT

We describe the integration of across-word models in the RWTH large vocabulary continuous speech recognition system, where our main focus is on the realization of the acoustic recognition process. This paper presents a study of two search methods based on the priniciple of dynamic programming. For both methods we discuss the implementation details and give experimental results on the *Verbmobil* and on the *Wall Street Journal* data. In addition, we introduce a score interpolation of within-word and across-word models for both search methods. In combination with across-word models this interpolation technique gives an improvement of the recognition accuracy by 14% relative to our standard system.

1. INTRODUCTION

This paper describes the integration of across-word modelling into the RWTH large vocabulary continuous speech recognition system [5]. In particular, we consider two search methods, namely the *n-best* and *one-pass* approach, for handling across-word models. Both methods are based on the word-conditioned search method using a tree-organized pronunciation lexicon. For the n-best method [6] several computation and optimization steps are required to obtain high recognition performance. In principle, a word graph is produced during the acoustic recognition process using within word models. We then extract the n best sentence hypotheses from the word graph and rescore only these hypotheses in a subsequent step with the across-word models. When using the one-pass algorithm combined with word-conditioned copies of the lexical prefix tree and across-word models, an efficient implementation is needed, since the recombination across word boundaries and the generation of a memory-optimized pronunciation tree is more complicated than in the case of handling within-word models [1, 7, 10]. To do this in an efficient way, we present some implementation details of this integrated search technique.

In addition, we introduce a simple method to interpolate the scores of the within-word and across-word models in a linear way. This linear interpolation scheme is especially useful for the n-best search where both models are already in use for the two acoustic recognition passes. For the one pass search, the interpolation also has proven its usefulness in achieving higher recognition accuracy.

The rest of the paper is organized as follows: In Section 2, we discuss the impact of across-word modelling on our training algorithm. Section 3 describes the n-best and the one-pass search method using across-word models. Furthermore, we present the interpolation scheme mentioned above. In the Section 4, we give experimental results on the Verbmobil and on the Wall Street Journal corpus, and finally in Section 5 we draw some conclusions.

2. TRAINING ASPECTS

For the integration of across-word models into the training procedure, the acoustic training is modified so that for every word transition, the between-word pause is estimated. According to this estimation, the correct across-word models are used at the word boundaries for the HMM of the sentence.

The pause length is estimated in every iteration of the training process. This information is saved in a file during one iteration and is then used in the next iteration to determine the models at the word boundaries according to the so-called silence threshold τ. Due to this approach, no information about the between-word pause length is available at the start of the training process on a new corpus. An examination of the duration distribution of the between word silences in the training corpora showed that about 80% of the word pairs have no pause at all between them. This means that 4 out of 5 transitions between words are modelled by across-word triphones, independent of the threshold for the length of the pause. Therefore, for a new corpus, we simply set all pause lengths to zero, taking into account the observation that the most part of the word transitions are without any pause between them. As a consequence, the training for a new corpus has to be done in two steps:

1. A phonetic decision tree for across-word models is calculated where all between-word silences are set to zero. Using this decision tree, a training is performed.
2. Using the information of the length of the between-word pause, a second phonetic decision tree is calculated and a training using this new decision tree is performed. The acoustic model estimated by this training is used for recognition.

The improvement of the word error rate is about 5% relative when compared to the results of recognition using the acoustic models of the first and the second iteration.

Apart from the iterative estimation of the between-word pause, no substantial modifications were made to the training procedure described in [5]. We use only one silence model with a single state, and position independent triphones [2].

3. SEARCH METHODS

In this section we present two methods for the integration of across-word models into the recognition process. In particular, we consider in the following a multiple-pass and a one-pass search strategy to handle across-word models in an efficient way. These two techniques are the n-best method and the integrated search method.

3.1. N-best Search

The *n*-best search method based on the word graph method as described in [8]. From a word graph it is relatively easy to derive n-best lists [9]. Therefore, our approach works basically in three steps:

- Generation of a word graph using the standard search algorithm and word internal triphones
- Generation of the n best sentences according to this word graph
- Rescoring of these sentences using both word internal and across-word triphones

The generation of the word graph uses the so-called word-pair approximation as described in [3]. For this it is assumed that the start time of a word w depends only on its direct predecessor word v. For short words v this approximation leads to a deterioration of the word error rate as demonstrated in [8]. However, for an n-best list this effect is not relevant because the approximation causes only small changes in the word scores. Therefore, the n-best list should be affected only for very small n ($n < 10$).

The generation of the n-best list on this word graph is based on the algorithm which normally extracts the best sentence out of the word graph. This algorithm works in the following way. For a given node in the word graph, it optimizes over the scores of all nodes which have an arc leading into the actual node, plus the local score at this node and the language model score. This optimization is done for all nodes of the word graph in a time-synchronous fashion. At the last node of the word graph, i. e. the ending time of the spoken sentence, the best sentence hypothesis can be extracted by using the traceback fields which also were generated during the search through the graph [8].

This method can simply be extended to extract the n best sentences from the word graph. Instead of storing only the best hypothesis in every node, the n best hypotheses are stored. The optimization is then performed over all mn hypotheses of the m predecessor words. By using this method it can be made sure that at the last node of the word graph the n best hypotheses can be found. A substantial speed-up of this method can be achieved by caching the scores of the language model because the word sequences of the sentences hypotheses considered in this process are very similar. Recently, we have implemented a new algorithm for the generation of a time-conditioned word graph without any approximations. This algorithm gives us an n-best list as a side effect, so we do not need any distinct method for the n-best list generation any more [4].

This n-best list is then rescored using both word-internal and across-word triphones. Therefore, the length of the pause between the word has to be measured. This is done either in the first or the second pass of the search process (see Section 4). The rescoring process can also be sped up by using a cache that stores the acoustic scores for each mixture distribution s and each time frame t. Because the sentence hypotheses are very similar, most pairs (s,t) did not change when rescoring a new sentence and can therefore be re-used.

3.2. Integrated Search

The integrated search method is based on the tree-organized pronunciation lexicon using word-conditioned copies of the lexical tree, where the recognition process is only performed in one search pass. Therefore, we have to incorporate the across-word models into the lexical prefix tree. In principle, the lexical tree has to fan out at phoneme arcs corresponding to word ends. For every phoneme following the actual word, a new arc is added to this fan-out so that every arc of the fan-out represents a hypothesized coarticulation with the starting phoneme of the possible following words. For these words, only those arcs of the first phoneme generation of the lexical tree are activated representing the corresponding phoneme arc of the fan-out. In addition, a silence arc is added to the tree root to allow no coarticulation at word boundaries. To do this in a memory-efficient way, we build in a pre-processing step a generic tree including the fan-out arcs for each word end. In addition, a separate copy of the first phoneme generation will be stored for each possible word start-up phoneme. During the recognition process, this part of lexical tree will be linked to the generic tree, of course, with respect to the across-word context.

To make the application of the dynamic programming (DP) approach possible, we structure the search as follows. Considering a bigram language model, we introduce for each hypothesized word end w and the final fan-out arc ${}_{\alpha}\beta_{\gamma}$ of w a separate copy of the lexical tree denoted by (w,β_γ). For incorporating the across-word triphones within a new tree start-up hypotheses, only the center phoneme β and the right phoneme context γ is required. Fig. 1 illustrates this concept for word end hypotheses of B with different predecessor words (A,B,C) in a simplified schematic form. Instead of showing the whole lexicon tree, Fig. 1 depicts only the last phoneme generation or strictly speaking the fan-out arcs of the word end hypotheses B and the first phoneme generation of the new tree hypotheses of B depending on the across-word context. The bigram probability $p(w|v)$ is incorporated into the partial overall score when the final state of word w with predecessor word v has been reached. The symbol $\circ$ denotes tree internal nodes and the symbol $\bullet$ illustrates word end nodes of a tree copy. We then collect for each word end hypothesis the best predecessor word v with respect to the fan-out arc ${}_{\alpha}\beta_{\gamma}$ so that each tree copy depends on the pair (v,β_γ). The corresponding overall score is then propagated into the root of the associated lexical tree, being represented by the symbol $\square$ in Fig. 1. The shadowed area in Fig. 1 marks the potential tree start-up hypotheses of B. For each new tree copy only the arcs of the first phoneme generation of the lexical tree are activated which are associated to the corresponding fan-out arc of B. When using this concept, we can formulate the recursions for the DP approach by introducing the following quantities:

> $Q_{(v,\beta_\gamma)}(t,s)$:= overall score of the best partial path that at time t ends in state s of the lexical tree for predecessor v with fan-out arc ${}_{\alpha}\beta_{\gamma}$.
>
> $B_{(v,\beta_\gamma)}(t,s)$:= starting time of the best partial path that at time t ends in state s of the lexical tree for predecessor v with fan-out arc ${}_{\alpha}\beta_{\gamma}$.

Note that the left phoneme context α of the fan-out arc ${}_{\alpha}\beta_{\gamma}$ stands for the immediate predecessor phoneme independent of the across-word context. Within a tree we can apply the usual DP recursion for the time alignment:

$$
\begin{aligned}
Q_{(v,\beta_\gamma)}(t,s) &= \max_{\sigma} \left\{ q(x_t,s|\sigma) \cdot Q_{(v,\beta_\gamma)}(t-1,\sigma) \right\} \\
B_{(v,\beta_\gamma)}(t,s) &= B_{(v,\beta_\gamma)}(t-1,\sigma_v^{max}(t,s)) \ ,
\end{aligned}
$$

where $\sigma_v^{max}(t,s)$ denotes the optimum predecessor state for the hypothesis (t,s) and predecessor pair (v,β_γ). The term $q(x_t,s|\sigma)$ is the product of transition and emission probabilities when going

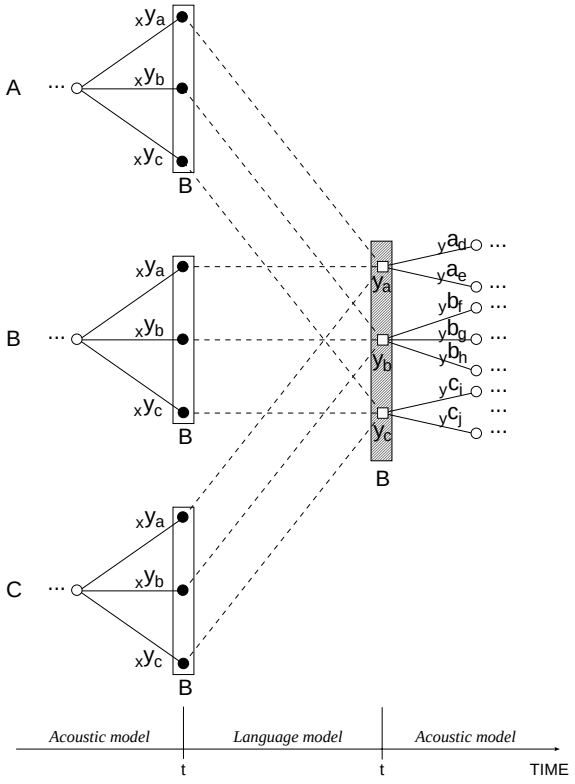

Figure 1: Word boundary recombination using a bigram language model combined with a lexical tree and across-word triphones.

from state σ to state s and observing vector x_t. To perform the recombination at word level, we have to select the predecessor pair $(v, \tilde{\beta_\gamma})$ for each word end hypothesis (w, β_γ). For this, we define the DP equation:

$$H(w, \beta_\gamma; t) \quad := \quad \max_{(v, \tilde{\beta_\gamma})} \left\{ p(w|v) \cdot Q_{(v, \tilde{\beta_\gamma})}(t, S_{(w, \beta_\gamma)}) \right\} \quad ,$$

where state $S_{(w,\beta_\gamma)}$ denotes the final state of word w in the fan-out arc ${}_\alpha\beta_\gamma$. Finally, we have to initialize the new tree-start hypothesis by passing on the score and the time index before processing the new hypotheses for time t:

$$\begin{aligned} Q_{(w,\beta_\gamma)}(t, s=0) &= H(w, \beta_\gamma; t) \\ B_{(w,\beta_\gamma)}(t, s=0) &= t \ . \end{aligned}$$

3.3. Interpolation

An observation we made during our experiments for the n-best method was the increase of the error rate for $n > 20$. This effect was found in all experiments we made for the *Wall Street Journal* corpus, and was called the "parasite effect" (see Table 2). By analysing the results more deeply, we observed an increase of insertions and substitutions when increasing the length of the n-best list. This phenomenon can be explained by the fact that by using across-word models, the modelling of the word boundaries is enhanced while the modelling of the word interior stays the same or, if a triphone occurs both in the word interior and the boundary, is deteriorated. For small n-best lists, this effect is partly masked because the decision for the best sentence is made both using within-word and across-word models. For longer n-best lists, the decision relies only on the across-word models. This means that by using relatively short n-best lists, a kind of implicit interpolation takes place. Therefore, we tried to explicitly interpolate the sentence end scores achieved by both models using the formula

$$C_{int} = \lambda \cdot C_{CW} + (1 - \lambda) \cdot C_{WW} \quad ,$$

where λ is the interpolation factor between 0 and 1, and C_{CW} and C_{WW} are the sentence end scores for across-word and within-word models. For the corpora we have used for our recognition experiments, we found that a factor of $\lambda = 0.7$ was optimal.

For the one-pass algorithm, the interpolation of the sentence end scores is not possible. Here, we interpolated the score at the state level. Therefore, we labelled every HMM of the lexial tree with two mixture indices, one for the within-word and one for the across-word models. During the search, two scores were calculated for each HMM state and then interpolated using the formula given above.

Table 1: Word error rate for different silence thresholds τ on WSJ Nov. 92 using the n-best method, $n = 20$.

silence threshold τ	WER[%]
baseline	7.1
1	6.4
2	6.6
5	6.7
∞	7.0

Table 2: Word error rate for different n and different estimations of the between-word pause length on WSJ Nov. 92 using the n-best method.

n best sentences	WER[%]	
	first pass	second pass
5	6.4	6.3
10	6.4	6.3
20	6.4	6.3
50	6.5	6.4
100	6.5	6.4

4. RECOGNITION EXPERIMENTS

In a first series of experiments, we analyzed the effect of the silence threshold τ on the Wall Street Journal corpus (WSJ Nov.'92 development and evaluation test data) using the n-best method. For this, we performed several training and recognition passes with silence thresholds from 1 up to ∞, and compared them to the baseline result without across-word modelling which corresponds to a silence threshold of 0. Table 1 shows that the optimal value for this parameter on the WSJ corpus is 1. For higher values a clear deterioration of the error rate can be observed, even for a value of 2 the error rate goes up from 6.4% to 6.6%.

In this experiment, the silence length between the words is estimated in the first recognition pass which uses only word-internal triphones. Because this length decides whether a word transition

Table 3: Word error rate for different n using the n-best method combined with and without linear score interpolation on WSJ Nov. 92 and Verbmobil 96.

n best sentences	WER[%]			
	WSJ		Verbmobil	
	no int	int	no int	int
5	6.3	6.2	21.1	21.1
10	6.3	6.2	20.8	20.7
20	6.3	6.1	20.6	20.5
50	6.4	6.1	20.6	20.2
100	6.4	6.1	20.5	20.0

Table 4: Word error rate for the integrated search method on Verbmobil 96.

method	WER[%]
baseline	21.9
no interpolation	21.4
interpolation	20.2

with or without coarticulation is used in the rescoring, one can argue that better estimations can be achieved if the decision about the handling of the word transition is made in the second pass where across word models can be used. Therefore, we modified the recognition algorithm to estimale the pause length during the second pass of the search. The results for this experiment are shown in Table 2. The influence of this modification on the error rate is marginal, only 0.1% gain can be achieved by the modified word transition handling. Nevertheless, because this method should be more accurate than the baseline method, we kept it for the rest of our tests to prevent any masking of certain effects by the baseline word transition handling. The results for the interpolation of the scores are shown in Table 3. For both corpora, the error rate was improved by about 3% relative, and, as a second improvement, the parasite effect was not observed any more. Even for higher n, where experiments were run only for the WSJ corpus, the error rate stayed around 6.1%. All in all, the proposed method gives a reduction of 14% relatively to the baseline result (see Table 1).

Finally, we tested the integrated search method on the Verbmobil 96 corpus. The recognition results are shown in Table 4. In an initial experiment, we obtained a word error rate of 21.9% by using only word-internal triphones. When using across-word models, a word error rate of 21.4 and 20.2 can be achieved without and with score interpolation respectively. These results suggest that the introduced linear interpolation scheme is important in order to achieve the maximal performance improvement by applying position-independent triphone models. The results of the integrated method will now be compared with the results of the n-best method. It seems that the n-best method leads to a slightly better recognition accuracy. However, we found that in most of the sentences the integrated search methods results in better sentence score than the n-best method.

5. SUMMARY

In this paper we have presented two efficient search methods for handling across-word models. Furthermore, we have introduced a linear interpolation scheme for combining the scores of the within-word and the across-word models. When using this linear interpolation in the context of position-independent triphone models the recognition accuracy can be improved by $10 - 14\,\%$.

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.

6. REFERENCES

[1] F. Alleva: Search Organization in the Whisper Continuous Speech Recognition, Santa Barbara, CA, December 1997, pp. 295–302, in S Furui, B.-H. Juang, W. Chou (eds.): '1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings', 1997.

[2] K. Beulen, E. Bransch, H. Ney: State Tying for Context Dependent Phoneme Models, Proc. Europ. Conf. on Speech Communication und Technology, Rhodes, Greece, pp. 1179-1182, September 1997.

[3] H. Ney, X. Aubert: A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition, Proc. Int. Conf. on Spoken Language Processing, Yokohama, Japan, Vol. 3, pp. 1355-1358, September 1994.

[4] H. Ney, S. Ortmanns, I. Lindam: Extensions to the Word Graph Method for Large Vocabulary Continuous Speech Recognition, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Munich, Germany, pp. 1787-1790, April 1997.

[5] 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, Seattle, WA, pp. 853-856, May 1998.

[6] L. Nguyen, R. Schwartz: Efficient 2-Pass N-Best Decoder, Fifth European Conference on Speech Communication and Technology, Rhodes, Greece, pp. 167-170, September 1997.

[7] J. J. Odell, V. Valtchev, P. C. Woodland, S. J. Young: A One-Pass Decoder Design for Large Vocabulary Recognition, Proc. ARPA Spoken Language Technology Workshop, Plainsboro, NJ, pp. 405-410, March 1994.

[8] S. Ortmanns, H. Ney, X. Aubert: A Word Graph Algorithm for Large Vocabulary Continuous Speech Recognition, Computer, Speech and Language, Vol. 11, No. 1, pp. 43-72, January 1997.

[9] R. Schwartz, S. Austin: A Comparison of Several Approximate Algorithms for Finding Multiple (N-Best) Sentence Hypotheses, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Toronto, Canada, pp. 701-704, September 1991.

[10] Q. Zhou, W. Chou: An Approach to Continuous Speech Recognition Based on Layered Self-Adjusting Decoding Graph, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Munich, Germany, pp. 1770-1782 April 1997.