6th European Conference on
Speech Communication and Technology
(EUROSPEECH'99)
Budapest, Hungary, September 5-9, 1999

AN EFFICIENT DECODING METHOD FOR REAL TIME SPEECH RECOGNITION

Stefan Ortmanns, Wolfgang Reichl, Wu Chou

Bell-Labs – Lucent Technologies
600 Mountain Ave., Murray Hill, NJ 07974, USA
ortmanns@research.bell-labs.com

10.21437/Eurospeech.1999-128

ABSTRACT

In this paper, we describe approaches for improving the search efficiency of a dynamic programming based one-pass decoder for dialogue applications. In order to allow the use of long-term language models (LM) and cross-word acoustic models, efficient pruning techniques and fast methods for the calculation of emission probability density functions (pdfs) are required. This is particularly important for real-time and memory constrained applications such as dialogue systems involving automatic speech recognition (ASR) and natural-language understanding. We propose an effective pruning technique exploiting the LM and cross-word context. We also present a fast distance calculation method to reduce the cost of state likelihood calculations in HMM-based systems. Experimental results on a natural language call routing task indicate that the proposed techniques speeded up the search process by a factor of 4 without loss in the recognition accuracy. In addition, we present a technique for generating word graphs incorporating cross-word context.

1. INTRODUCTION

For many ASR-based applications the performance of the decoder is essential for user acceptance of a system. We first give a brief description of a time-synchronous one-pass beam search decoder based on a layered self-adjusting decoding graph [8]. We then present some techniques to speed up the decoding process for fast system response. The contributions of this paper are:

- We propose a new effective pruning technique to reduce the number of tree hypotheses conditioned upon the full cross-word model contexts and the word history associated with long term constraints.
- In order to reduce the computational effort of the state likelihood calculation, we developed several new extensions to the vector quantization (VQ) based fast distance calculation method [2].
- Finally, we describe an accurate method for producing very small word hypothesis graphs. This method works in a time-synchronous manner and fits directly into the architecture of a the one-pass beam search decoder.

Experimental results are given on a natural language call routing task with a 1200 -word vocabulary [6]. In all experiments, a trigram LM with a perplexity of 25 and phonetic decision tree based full (within-word and cross-word) context dependent acoustic models were used.

2. BASELINE SEARCH ALGORITHM

The search algorithm is based on a self-adjusting decoding graph as described in [8]. This approach utilizes a scaffolding-layer technique to steer the expansion and the release of the active search space with respect to long-term language models and cross-word acoustic models. To evaluate the search space in a time-synchronous way by exploiting the principle of dynamic programming (DP) and a tree-organized pronunciation lexicon, we condition the search hypotheses to the full cross-word contexts and, of course, the word history associated with long term language model constraint [3]. Strictly speaking, for each word ending corresponding to a specific cross-word context, the part (or subtree) of the lexical tree will be activated which represents the particular cross-word context and corresponds to the required word-history due to the LM constraints.

We will now describe the DP approach in more detail. We use the notation similar to [1]. For a general m-gram language model, we have the quantity:

$Q_{(v_1^{m-1}, {}_\alpha\beta\_)}(t,s)$ denoting the overall score of the partial path that ends at time t in state s of the lexical subtree for predecessor word sequence $v_1^{m-1} = v_1, ..., v_{m-1}$ with the cross-word context triphone ${}_\alpha\beta\_$, where α denotes the left context of the actual phone β.

The right context of this specific cross-word context of the subtree start-up phoneme does not affect the cross-word boundary and can be consequently neglected (denoted by the symbol "_"). For the time alignment within a word-conditioned tree hypothesis, we use the well-known DP recursion:

$$Q_{(v_1^{m-1},{}_\alpha\beta\_)}(t,s) \;=\; \max_{\sigma}\left\{\, q(x_t,s|\sigma)\cdot Q_{(v_1^{m-1},{}_\alpha\beta\_)}(t-1,\sigma)\,\right\}.$$

The term $q(x_t,s|\sigma)$ denotes the product of emission and transition probabilities when going from state σ to state s and observing the acoustic vector x_t at time frame t. To define the DP recombination at word boundaries, we use the auxiliary quantity $H(v_2^m,\_\gamma_\delta;t)$ as the joint probability of observing the acoustic vectors $x_1,...,x_t$ and a word sequence ending at time t with v_2^m in cross-word context $\_\gamma_\delta$. For a potential word hypothesis v_m, we then have:

$$H(v_2^m,\_\gamma_\delta;t) \;=\; \max_{v_1}\left\{\, p(v_m|v_1^{m-1})\cdot Q_{(v_1^{m-1},{}_\alpha\beta\_)}(t,S(v_m,\_\gamma_\delta))\right\},$$

where $S(v_m,\_\gamma_\delta)$ is a final state of word v_m ending at time frame t in phoneme arc $\_\gamma_\delta$ (γ describes the actual phone and δ the right context in the sense of a triphone model). Finally, we have to initialize the new subtree start-up hypothesis by passing the score $H(v_2^m,\_\gamma_\delta;t)$ before processing the new hypothesis at time frame $t+1$:

$$Q_{(v_2^m,\gamma_\delta\_)}(t,s=0) \;=\; H(v_2^m,\_\gamma_\delta;t)\,,$$

where $s=0$ denotes an artificial start-up state. Note that due to the scaffolding layer technique only phoneme arcs of the new lexical tree hypothesis corresponding to the cross-word context $\_\alpha_\delta$ will be dynamically activated. To realize a fast access during the search process according to the so-called word-conditioned tree hypotheses a two-level hashing structure is used [8].

2.1. Improved Pruning

In order to control the search space a standard pruning approach is applied during the search process. This pruning approach consists of: *acoustic pruning*, *word-end pruning* and *histogram pruning*, which are performed at every time frame. However, for handling full cross-word models and long term LMs this approach is not sufficient to achieve real-time decoding with high recognition accuracy. Strictly speaking, for decoding using cross-word context dependent models, multiple new tree hypotheses corresponding to different cross-word contexts are started for each active word-end sequence and at each time frame, causing a major increase in search complexity.

The traditional beam search techniques are ineffective to control such expansions. Therefore, we apply in addition to the word-end pruning, a so-called *cross-word pruning*. This pruning technique exploits the knowledge of the LM and the full cross-word model context to reduce the number of active tree hypotheses during the search process. For each active word-end sequence, we determine the best score among all possible starting phoneme arcs and all copies of the lexical tree associated with that word history. The number of possible active phoneme arcs depends on the number of different cross-word contexts at that word end. Then, a new copy of a subtree will be initialized from that word history, only if the associated phoneme arc score is close to the locally best score at that time frame.

The proposed approach can be integrated into *language model look-ahead based pruning*, and the efficiency of this pruning technique can be further improved. For this purpose, the distributed LM probabilities assigned to the first phoneme associated with all potential start-up lexical subtree hypotheses are incorporated into the pruning criterion. That means, for states representing an active word-end sequence with respect to the corresponding cross-word context, we consider the maximum LM score of all words that can be reached via this specific cross-word context. Due to the memory constraints, we use a *simplified* LM look-ahead based on a unigram LM or on the *best* m-gram LM $\max_{v_m\in W(\_\gamma_\delta)} p(v_m|v_1^{m-1})$ where $W(\_\gamma_\delta)$ is the set of words reachable via this particular cross-word context $\_\gamma_\delta$.

2.2. Fast likelihood calculation

One of the most computationally expensive operations in HMM-based speech recognition system is the calculation of the state likelihoods. To reduce the computational effort of the state likelihood calculation, we developed several new extensions to the vector quantization (VQ) based fast distance calculation method [2]. In particular, we have studied:

- *Maximum approximation:*
  Instead of summing up over all Gaussian mixture densities of a state, the best density of each state is only used.

- *State flooring:*
  States that are not represented by the nearest VQ cell for a given acoustic observation vector are assigned to a predefined standard floor value.

- *Density Preselection:*
  When going from time frame t to time frame $t+1$ the distance between the adjacent observation vectors x_t and x_{t+1} is first calculated. If the distance is below a certain threshold, the same log-likelihood scores are used for state hypotheses which were active at time frame t.

Furthermore, the VQ method is enhanced by using a simplified *projection search algorithm* (PSA) [5]. The idea of this hybrid fast distance calculation is to calculate the likelihood only for densities (prototype vectors) in selected VQ cells and also inside a hypercube surrounding the input observation vector. Note that the hypercube is generated on demand. In contrast to the method described in [5] which was established for Laplacian densities, we apply in our approach Gaussian densities. In addition, we scale each side of the hypercube to accommodate different scaling in each parameter dimension.

3. WORD GRAPH METHOD

In applications such as dialogue systems, it is useful to generate a small word graph for postprocessing and natural language understanding. The word graph method presented in this section tries to avoid any approximation in the framework of beam search and is similar to the work described in [4, 7]. However, in our approach we incorporate the cross-word context information to build a *time-conditioned word lattice*. The basic idea of this word graph method is to store all hypothesized word ends during the acoustic search process within a predefined beam width, which is similar to the pruning threshold used in the word-end pruning technique. In particular, we store the word identity v_m, the corresponding acoustic word score $h(v_{m,\alpha}\,\beta\_,\_\gamma_\delta;\tau,t)$, starting time τ, ending time t and the starting phoneme ${}_\alpha\beta\_$ and the ending phoneme $\_\beta_\gamma$ of v_m. $h(v_{m,\alpha}\,\beta\_,\_\gamma_\delta;\tau,t)$ can be computed as:

$$h(v_{m,\alpha}\,\beta\_,\_\gamma_\delta;\tau,t) \;:=\; \frac{Q_{(v_1^{m-1},\alpha\beta\_)}(t,S(v_m,\_\gamma_\delta))}{H(v_1^{m-1},_\alpha\,\beta\_;\tau)}\;.$$

However, keeping all word-end hypothesis within the beam width leads in general to unmanageable storage requirements. To reduce the size of the word graph, we apply the so-called *word boundary optimization* [4] during the word graph construction. That means, when applying an m-gram LM it is sufficient to distinguish word sequences only by their final $(m-1)$ words so that we can recombine word sequences having the same final $(m-1)$ words at a given time frame:

$$H(v_2^m,\_\gamma_\delta;t) \;=\; \max_{v_1}\left\{\, p(v_m|v_1^{m-1})\cdot \max_\tau\left\{H(v_1^{m-1},\_\alpha_\beta;\tau)\cdot h(v_{m,\alpha}\,\beta\_,\_\gamma_\delta;\tau,t)\right\}\right\}.$$

This calculation can be efficiently performed in a subsequent operation by generating a word-hypothesis tree (sentence hypothesis tree) [4]. The use of the m-gram LM probabilities serves only for the purpose of a more efficient word graph pruning. In a final transformation, we store all word hypotheses surviving the word boundary optimization and that are also within the search beam width. Word hypotheses with the same word identity and the same cross-word contexts will be stored only once if they have the same starting and ending time nodes.

To further reduce the memory requirements needed during the construction process all word hypotheses leading not to the end of the spoken sentence will be removed from the word graph. This so-called *purging* process can be also applied in regular intervals, say every 100 time frames. Word hypotheses which cannot be reached from any active state hypotheses will be removed from the word graph and from the sentence hypothesis tree. In addition, we limit the number of path hypotheses at a given time frame to a maximum number. When generating the sentence hypothesis tree, the list of the n best sentences can easily be produced.

4. EXPERIMENTAL RESULTS

4.1. Recognition Task and Database

To demonstrate the efficiency of the proposed techniques, experimental tests were applied to a natural language call routing task (2 100-word vocabulary) [6]. The test set consists of 1956 spoken sentences (a total of 11116 spoken words with a length of 7672 seconds). In all experiments, a trigram LM with a perplexity of 25 and a set of phonetic decision tree based full (within-word and cross-word) context dependent acoustic model were used.

The results of the speed-up techniques are summarized in Table 1. The baseline search algorithm without any kind of look-ahead pruning techniques leads to 8.3 times real-time (real-time factor (RTF): 8.3) on a PC 350 MHz under Linux. Incorporating a simplified LM look-ahead on the basis of the best trigram, the word error rate (WER) and the real-time behaviour of the search method can be significantly improved. Using the proposed cross-word pruning in combination with hybrid fast distance calculation method (VQ+PSA) the search speeded up by a factor of 4.4 with nearly no loss of recognition performance as compared to the baseline result. Finally, the density preselection gives additionally a further acceleration of the search process.

Table 1: Performance of a single-best decoder on a natural language call routing task (PC 350 MHz).

System	WER [%]	RTF
Baseline	24.1	8.3
+ Simplified LM-LA	22.9	4.4
+ VQ + PSA	24.1	2.4
+ Cross-word Pruning	24.0	1.9
+ Density Preselection	24.0	1.7

Table 2: Recognition results for the word graph method on a natural language call routing task using different word-graph pruning thresholds (single-best result: WER = 22.9%).

F_{WG}	WGD	GER [%]
180	457	11.18
160	433	11.20
100	251	11.26
80	162	11.38
60	89	11.57
40	40	12.18
20	11	13.73
10	4	14.70
5	2	14.85

In another series of recognition experiments, we have studied the word-graph method for different thresholds F_{WG} of the time-synchronous pruning. Table 2 shows the *word graph error rate* (GER) as a function of the *word graph density* (WGD). The word graph error rate is computed by determining that sentence through the word graph that best matches the spoken sentence in terms of word errors. The word graph density is defined as the number of word hypotheses per spoken word. It can be seen form this table, that even for a WGD of 2 the GER is clearly smaller than the WER of 22.9% (see Table 1) for the single-best sentence and can help achieve better understanding results in a natural language ASR based call routing dialogue system [6]. The cost for the word graph generation, including all optimization steps, is less than 2% of the whole search process.

5. SUMMARY

This paper has presented some techniques for speeding up the acoustic search process of a one-pass decoder. In particular, we have introduced a new word-end pruning technique with regard to full cross-word context and we have studied techniques to reduce the effort of the HMM state log-likelihood calculation. When using these techniques, we achieve nearly real-time performance on a PC 350 MHz for a natural language call routing task. In addition, we have presented a method for generating word graphs. This method builds efficiently time-conditioned word graphs including the cross-word context information.

6. REFERENCES

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

[2] S.M. Herman, R.A. Sukkar: Variable Threshold Vector Quantization for Reduced Continuous Density Likelihood Computation in Speech Recognition, Santa Barbara, CA, Dec. 1997, pp. 331-338, '1997 IEEE Workshop on Automatic Speech Recognition and Understanding Proceedings', 1997.

[3] 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.

[4] H. Ney, S. Ortmanns: 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] S. Ortmanns, H. Ney, T. Firzlaff: Fast Likelihood Computation Methods for Continuous Mixture Densities in Large Vocabulary Speech Recognition. Fifth European Conference on Speech Communication and Technology, Rhodes, Greece, pp. 143-146, September 1997.

[6] W. Reichl, B. Carpenter, J. Chu-Carroll, W. Chou: Language Modeling for Content Extraction in Human-Computer Dialogues, Proc. Int. Conf. on Spoken Language Processing, Syndey, Australia, Nov./Dec. 1998.

[7] A. Sixtus, S. Ortmanns: Hiqh Quality Word Graphs using Forward-Backward Prunig. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Phoenix, AZ, pp. 593-596, March 1999.

[8] 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.