LOOK-AHEAD TECHNIQUES FOR FAST BEAM SEARCH

Stefan Ortmanns, Andreas Eiden, Hermann Ney and Norbert Coenen

Lehrstuhl für Informatik VI, RWTH Aachen – University of Technology,
D-52056 Aachen, Germany

ABSTRACT

This paper presents two look-ahead techniques for speeding up large vocabulary continuous speech recognition. These two techniques, which are referred to as language model look-ahead and phoneme look-ahead, are incorporated into the pruning process of the time-synchronous one-pass beam search algorithm. The search algorithm is based on a tree-organized pronunciation lexicon in connection with a bigram language model. Both look-ahead techniques have been tested on the 20 000-word NAB'94 task (ARPA North American Business Corpus). The recognition experiments show that the combination of bigram language model look-ahead and phoneme look-ahead reduces the size of search space by a factor of about 30 without affecting the word recognition accuracy in comparison with no look-ahead pruning technique.

1. INTRODUCTION

In this paper, we present the combination of language model look-ahead and phoneme look-ahead pruning for large vocabulary continuous speech recognition. The basic idea of the language model look-ahead is to fully incorporate the language model (LM), e.g. a bigram or trigram language model, as early as possible into the pruning process of the time-synchronous search algorithm using word dependent copies of the lexical tree (to be more exact: lexical prefix tree). To use the look-ahead for a bigram language model, we factor the bigram probabilities over the nodes of the lexical tree for each copy of the lexical tree [1, 2, 6, 7, 8]. To do this in an efficient way, we introduce special implementation details [6]. In addition to the LM look-ahead, we present a phoneme look-ahead which is similar to the method described in [4]. The idea of this look-ahead technique is to estimate the likelihood of each phoneme ahead of the current time frame. This probability estimate is then used in an additional pruning step. To reduce the computational cost of the phoneme look-ahead, we describe suitable simplifications of the phoneme models.

The organization of this paper is as follows. In Section 2, we review the one-pass beam search using a tree-organized pronunciation lexicon in combination with a bigram language model. In Section 3, we present the language model look-ahead. In Section 4, we present the phoneme look-ahead. In Section 5, we give experimental results on the NAB'94 20000-word development data.

2. BASELINE SEARCH METHOD

In this section, we briefly review the widely used time-synchronous one-pass dynamic programming search method in connection with a tree-organized pronunciation lexicon and a bigram language model [4, 5]. To formulate the dynamic programming approach, we introduce the following quantity [5]:

> $Q_v(t,s) :=$ overall score of the best path up to time t that ends in state s of the lexical tree for predecessor word v.

The dynamic programming recursion for $Q_v(t,s)$ in the word interior is:

$$Q_v(t,s) \quad = \quad \max_{\sigma} \{ \, q(x_t, s|\sigma) \cdot Q_v(t-1,\sigma) \, \} \; ,$$

where $q(x_t, s|\sigma)$ is the product of transition and emission probabilities of the underlying 6-state Hidden Markov Model. At the word level, we have to find the best predecessor word for each word hypothesis. For this purpose, we define:

$$H(w;t) \quad := \quad \max_{v} \{ \, p(w|v) \cdot Q_v(t, S_w) \, \} \; ,$$

where S_w denotes a terminal state of the lexical tree for word w. To start up new words, we have to initialize $Q_v(t,s)$ as:

$$Q_v(t-1,0) \quad = \quad H(v;t-1) \; ,$$

where the fictitious state $s = 0$ is used to initialize a tree.

The standard pruning approach consists of three steps, e.g. standard beam pruning or so-called acoustic pruning, language model pruning and histogram pruning, that are performed every 10-ms time frame as described in [8]. The efficiency of this standard pruning approach can be improved by using the so-called look-ahead techniques, which are presented in the following.

3. LANGUAGE MODEL LOOK-AHEAD

The basic idea of the language model look-ahead is to incorporate the language model probabilities as early as possible into the search process and thus into the associated pruning process. This is achieved by factoring the language model probabilities over the nodes of the lexical tree. For a

0-8186-7919-0/97 $10.00

bigram language model, the factored LM probability $\pi_v(s)$ for state s and predecessor word v is defined as:

$$\pi_v(s) := \max_{w \in \mathcal{W}(s)} p(w|v) ,$$

where $\mathcal{W}(s)$ is the set of words that can be reached from tree state s. The term $p(w|v)$ denotes the conditional bigram probabilities. After the LM look-ahead tree factorization, i.e. computing $\pi_v(s)$, each node (or phoneme arc) of a lexical tree copy is assigned to the maximum bigram probability over all words that are reachable via this specific node from predecessor word v.

We incorporate the factored LM probabilities $\pi_v(s)$ into the dynamic programming recursion across phoneme boundaries:

$$Q_v(t,s) = \frac{\pi_v(s)}{\pi_v(\tilde{s})} \cdot \max_{\sigma} \{ q(x_t, s|\sigma) \cdot Q_v(t-1, \sigma) \} ,$$

where $\tilde{s}$ is the parent node of s. For state transitions not involving phoneme boundaries, we have to use the same equation as described in Section 2. To compute the start-up score $H(w,t)$, we have to take into account that, at the end nodes of the lexical trees, the language model probabilities have already been included. Hence we have simply:

$$H(w;t) := \max_{v} \{ Q_v(t, S_w) \} .$$

To reduce the memory and computational cost, this approach of on-demand calculation is further refined by additional steps [6]: The memory cost for storing the LM look-ahead probabilities depends on the number of nodes of the original pronunciation tree. This tree can be compressed because there are many tree nodes that have only one successor node.

Instead of calculating the LM factored probabilities for all possible tree copies beforehand, we calculate the LM factored probabilities on demand for each new tree copy depending on predecessor word v and store these factored probabilities in a look-up table. A dynamic programming procedure allows us to compute the LM factored probabilities in an efficient way. We initialize the leaves of the LM look-ahead tree with the bigram language model probabilities $p(w|v)$. Then the LM factored probabilities are propagated backwards from the tree leaves to the tree root by using a dynamic programming recursion, which, for each tree node, determines the successor node with maximum look-ahead probability.

4. PHONEME LOOK-AHEAD

The phoneme look-ahead is based on the following concept [4]. Each time a hypothesis is formed about a new phoneme arc to be started in the search process, it is first checked whether this new phoneme arc hypothesis is likely to survive the pruning steps that will be performed for the next future time frames. To this purpose, we compute an approximate probability estimate for each possible phoneme arc that can be activated at a given time frame in the beam search. This approximate probability estimate, which is referred to as look-ahead score, is then combined with the detailed score of its predecessor phoneme and used in an additional pruning step, in which all hypotheses of phoneme arcs to be started up are considered time-synchronously in the usual spirit of beam search.

To formulate the phoneme look-ahead and the associated pruning operation in detail, we use the following notation:

> α: one of the phoneme arcs to be started in the lexical tree. Note that the same phoneme arc α may occur in different copies of the lexical tree.
>
> $\tilde{\alpha}$: the unique parent arc of α in the lexical tree, for which one of the final states has been reached in the search process. Note that this mapping $\alpha \rightarrow \tilde{\alpha}$ captures the lexical constraints as given by the pronunciation lexicon.
>
> $\hat{q}(\alpha; t, \Delta t)$: probability that the phoneme α produces the acoustic vectors $x_{t+1}, ..., x_{t+\Delta t}$. Δt is in the order of an average phoneme duration, e.g. 6 or 7 10-ms time frames.

For the phoneme look-ahead pruning, we combine this look-ahead score $\hat{q}(\alpha; t, \Delta t)$ with the detailed score $Q_v(t,s)$. Thus for a given time frame t, we compute the following score for each possible pair (α, v) of phoneme arc α and lexical tree for predecessor word v:

$$\hat{Q}_v(t,\alpha) := \hat{q}(\alpha; t, \Delta t) \cdot Q_v(t, S_{\tilde{\alpha}}) ,$$

where $S_{\tilde{\alpha}}$ denotes the final state of phoneme arc $\tilde{\alpha}$. For notational simplicity, we have assumed that there is exactly one final state. If there are several final states, we select the best one. As in all time-synchronous pruning methods, the pruning is based on computing the best score $Q_{LA}(t)$ of all hypotheses under consideration for time t:

$$Q_{LA}(t) := \max_{(v,\alpha)} \{ \hat{Q}_v(t,\alpha) \} .$$

A phoneme arc hypothesis (α, v) at time t is removed (or, depending on the viewpoint, not started at all in the detailed search) if

$$\hat{Q}_v(t,\alpha) < f_{LA} \cdot Q_{LA}(t) ,$$

where f_{LA} denotes the phoneme look-ahead pruning threshold. In the experimental tests, we have found that there is no loss in performance when we use the following (or a similar) approximation for $Q_{LA}(t)$:

$$Q_{LA}(t) \cong \max_{\alpha} \{ \hat{q}(\alpha; t, \Delta t) \} \cdot \max_{(v,\beta)} \{ Q_v(t, S_\beta) \} ,$$

where the symbol β stands for an arbitrary phoneme arc *independent* of phoneme arc α. This means that we do *not* use the lexical constraints when computing the reference score for the pruning step. However for each individual arc hypothesis, it is very important to take the exact lexical constraints into account.

In order to compute the phoneme look-ahead score $\hat{q}(\alpha; t, \Delta t)$, we perform a time alignment for each hypothesized phoneme α. To this purpose, we define:

> $\phi_\alpha(\tau, s; t)$: score of time aligning the acoustic vectors $x_{t+1}, ..., x_{t+\tau}$ with the states $1, ..., s$ of phoneme arc α.

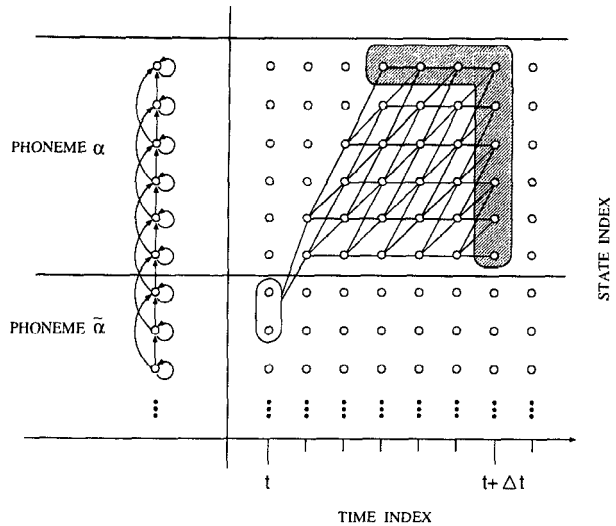

Figure 1. Phoneme look-ahead using a 6-state HMM.

The time alignment scores $\phi_\alpha(\tau, s; t)$ are computed by dynamic programming. The details and the computational effort depend on the type of phoneme models and of the underlying HMM. In general, we use a 6-state HMM representing a phoneme model. For such a 6-state HMM, the concept of the look-ahead time alignment is illustrated in Fig. 1. The shadowed area in Fig. 1 marks the potential states in which the look-ahead time alignment path may end.

To compute the phoneme look-ahead score $\hat{q}(\alpha; t, \Delta t)$, we have to consider the scores of the potential ending states of the time alignment path. By normalizing the scores with respect to different durations τ, we obtain the following equation for the phoneme look-ahead score $\hat{q}(\alpha; t, \Delta t)$:

$$\hat{q}(\alpha; t, \Delta t) := \max\left\{\max_s\{\phi_\alpha(\Delta t, s; t)\}\ ,\ \max_\tau\left\{\phi_\alpha(\tau, S; t)^{\Delta t/\tau}\right\}\right\},$$

where the symbol S stands for the final state of the HMM.

So far, we have not considered the computational cost of computing the time alignment look-ahead scores. Evidently, the phoneme look-ahead can only result in a speed-up of the search process if this additional computational effort is sufficiently small. Using the same phoneme models in both the detailed search and the look-ahead time alignment is prohibitive for the following reason. Like most other speech recognition systems, we use context dependent (CD) phoneme models rather than context independent (CI) phoneme models in the detailed search process. The number of these CD models is typically in the range of several thousands. In addition, for the emission distributions of the HMMs, we use mixture distributions with a huge number of component densities.

Therefore to keep the effort for computing the phoneme look-ahead scores small, we consider the following methods:

- Instead of CD phoneme models, we use CI phoneme models, say 40 – 50, for the phoneme look-ahead.
- We use only a small number of component densities, e.g a total of a few hundreds, to model the emission distributions.

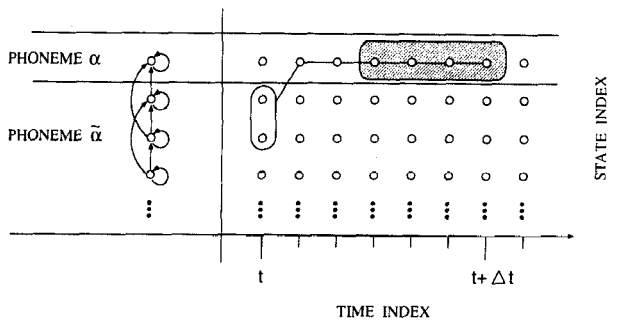

Figure 2. Phoneme look-ahead using a 1-state HMM.

- The calculation of the phoneme look-ahead can be performed every second time frame [4].
- To further reduce the amount of computation, we simplify the structure of each phoneme by collapsing all states into only one state as shown in Fig. 2 [3]. As a result, each model has only one emission probability distribution.

5. EXPERIMENTAL RESULTS

The experimental tests were carried out on the ARPA North American Business (NAB'94) H1 development corpus comprising 310 sentences with a total of 7387 words. 199 of the spoken words were out-of-vocabulary words. The training of the emission probability distributions was performed on WSJ0 and WSJ1 training data. In all experiments, we used about 290 000 Laplacian mixture densities (with a single pooled vector of absolute deviations) for each gender and a bigram language model with a perplexity (PP) of 198.4. The experiments were performed on a SGI workstation with a R4400 processor (91.7 SpecInt92).

First, we investigated the effect of the LM look-ahead on the size of search space and the word error rate. Table 1 shows the results and the set-up of the LM look-ahead tree in terms of the number of arcs and arc generations (gen.) and of the maximum number of LM look-ahead trees. In addition, the search space, the recognition word error rate (DEL–INS and WER[%]) and the real time factor (RT) are given. In an initial experiment, we performed two tests without any language model look-ahead. Then we tested the unigram LM look-ahead as described in [8]. The unigram LM look-ahead reduces the search space by a factor of about 4 without loss in recognition accuracy. Finally, we tested the bigram LM look-ahead for various numbers of arc generations used for the LM look-ahead trees, namely 17 (full look-ahead tree), 3, 2, 1 as shown in Table 1. We see that the best results are obtained for 3 and more arc generations. The search space is reduced by a factor of about 5 over that of the unigram LM look-ahead.

In a second recognition experiment, we added the phoneme look-ahead to the bigram LM look-ahead and studied the effect on the search effort. Unlike the detailed search, in which 4688 context dependent phoneme models are used, the inventory of the look-ahead phoneme models consists of only 43 context independent phoneme models. In addition, there is a silence model that always comprised a single state. We tested the following variants of the look-ahead models: 1-state models with a total of either 175 or

Table 1. Effect of the LM look-ahead on the search effort and recognition results (NAB'94 H1 development set; bigram LM with $PP = 198.4$).

LM look-ahead	LM look-ahead tree			search space			recognition errors [%]		RT
	gen.	arcs	trees	states	arcs	trees	DEL / INS	WER	
no	–	–	–	65568	16932	26	2.4 / 2.5	16.3	139.3
	–	–	–	50020	13034	20	2.5 / 2.5	16.6	115.7
unigram	17	63155	1	16960	4641	32	2.5 / 2.5	16.4	86.2
($PP = 972.6$)	17	63155	1	9443	2599	22	2.6 / 2.5	16.8	68.9
bigram	17	29270	300	3312	935	13	2.5 / 2.6	16.5	41.6
($PP = 198.4$)	3	12002	300	3277	924	13	2.4 / 2.6	16.5	39.8
	2	4097	300	3611	1012	12	2.4 / 2.6	16.9	40.8
	1	544	300	5786	1643	11	2.6 / 2.8	17.0	45.8

Table 2. Effect of the phoneme look-ahead ($\Delta t = 6$) in combination with the bigram LM look-ahead (17 phoneme generations) on the search effort and recognition results (NAB'94 H1 development set; bigram LM with $PP = 198.4$).

phoneme look-ahead		search space			recognition errors [%]		
model	densities	states	arcs	trees	DEL / INS	WER	RT
1-state HMM	175	1901	452	10	2.5 / 2.7	16.6	22.7
	175	1551	359	9	2.5 / 2.7	16.7	19.5
	675	1808	434	11	2.5 / 2.6	16.6	20.7
	675	1508	354	10	2.5 / 2.6	16.8	18.0
6-state HMM	497	2472	605	12	2.5 / 2.6	16.5	24.5
	497	1671	379	10	2.4 / 2.7	16.8	20.7
	1926	2160	516	11	2.5 / 2.6	16.5	22.8
	1926	1784	413	11	2.5 / 2.7	16.6	21.0

675 densities and 6-state models with a total of either 497 or 1926 densities. Table 2 summarizes the results. It can be seen that there are no significant differences between the two look-ahead variants. Increasing the number of mixture densities in the phoneme look-ahead leads to a small reduction of the search effort. For the best choice of conditions, the size of the search space *and* the total recognition time are halved while the word error rate goes up only from 16.5% to 16.6%. The computational cost of the phoneme look-ahead is negligible (5%) in comparison with the effort for the total search.

6. SUMMARY

This paper has presented and studied the combination of language model look-ahead and phoneme look-ahead for improved beam search. The experiments performed on the NAB'94 20 000-word task have shown that this combination leads to a reduction of the size of search space by a factor of about 30 with virtually no loss in the recognition accuracy in comparison with no look-ahead techniques. Due to the cost of the likelihood calculations for the detailed search, this results in an overall speed-up of the recognition process by a factor of about 5.

REFERENCES

[1] F. Alleva, X. Huang, M.-Y Hwang: Improvements on the Pronunciation Prefix Tree Search Organization. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Atlanta, GA, pp. 133 - 136, May 1996.

[2] G. Antoniol, F. Brugnara, M. Cettolo, M. Federico: Language Model Representations for Beam-Search Decoding. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Detroit, MI, Vol. 1, pp. 588 - 591, May 1995.

[3] L.R. Bahl, S.V. De Gennaro, P.S. Gopalakrishnan, R.L. Mercer: A Fast Approximate Acoustic Match for Large Vocabulary Speech Recognition. IEEE Trans. on Speech and Audio Processing, Vol. 1, No. 1, pp. 59-67, January 1993.

[4] H. Ney, R. Haeb-Umbach, B.-H. Tran, M. Oerder: Improvements in Beam Search for 10000-Word Continuous Speech Recognition. 1992 IEEE Int. Conf. on Acoustics, Speech and Signal Processing, San Francisco, CA, pp. 13-16, March 1992.

[5] H. Ney: Search Strategies for Large-Vocabulary Continuous-Speech Recognition. NATO Advanced Studies Institute, Bubion, Spain, June-July 1993, pp. 210–225, in A.J. Rubio Ayuso, J.M. Lopez Soler (eds.): 'Speech Recognition and Coding – New Advances and Trends', Springer, Berlin, 1995.

[6] S. Ortmanns, H. Ney, A. Eiden: Language-Model Look-Ahead for Large Vocabulary Speech Recognition. Proc. Int. Conf. on Spoken Language Processing, Philadelphia, PA, pp. 2095-2098, October 1996.

[7] S. Renals, M. Hochberg: Efficient Search Using Posterior Phone Probability Estimates, Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Detroit, MI, Vol. 1, pp. 596 - 599, May 1995.

[8] V. Steinbiss, B.-H. Tran, H. Ney: Improvements in Beam Search. Proc. Int. Conf. on Spoken Language Processing, Yokohama, Japan, pp. 2143-2146, September 1994.