

LOOK-AHEAD TECHNIQUES FOR FAST BEAM SEARCH

S. Ortmanns, H. Ney, N. Coenen, A. Eiden

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

SUMMARY

In this paper, we present two efficient look-ahead pruning techniques in beam search for large vocabulary continuous speech recognition. Both techniques, the language model look-ahead and the phoneme look-ahead, are incorporated into the word conditioned search algorithm using a bigram language model and a lexical prefix tree [5]. The paper presents the following novel contributions:

- We describe a method for language model (LM) look-ahead pruning which is similar to [1, 9]. We show special techniques to reduce the memory and computational requirements. These techniques are based on a compressed LM look-ahead tree. To compute the LM look-ahead tree probabilities in an efficient way, we present a backward dynamic programming scheme.
- We present a phoneme look-ahead pruning technique to increase the efficiency of the acoustic pruning. In particular, we refine the acoustic pruning strategy by a 1- and 2-phoneme look-ahead, respectively.
- We report results for both look-ahead pruning methods on the the 20,000-word North American Business (NAB'94) task.
- As a result, the combination of bigram look-ahead and 1-phoneme look-ahead reduces the search space by a factor of 10 without loss in recognition accuracy in comparison with the baseline search using a unigram language model look-ahead as described in [2]. The computational costs can be reduced by a factor of 5 on a SGI workstation (Indy R4400).

1. BASELINE SEARCH METHOD

The so-called word conditioned tree search algorithm is based on a strictly time-synchronous left-to-right dynamic programming search method combined with a tree-organized pronunciation lexicon [5, 6, 2]. When using a bigram language model, we introduce a separate copy of the lexical tree for each predecessor word v . To formulate the dynamic programming approach, we introduce the following quantity [6]: $Q_v(t, s) :=$ 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:

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

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:

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

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:

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

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 and are novel in the context of word conditioned tree search.

2. LANGUAGE MODEL LOOK-AHEAD

The basic idea of this pruning method is to incorporate the language model (LM) knowledge as early as possible into the search process [9, 1, 7, 8]. This is achieved by factoring the language model probabilities over the nodes of the lexical tree. For a 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. Strictly speaking, we use the tree nodes rather than the states of the Hidden Markov models that are associated with each node. However, each initial state of a phoneme arc can be identified with its associated tree node.

After the LM look-ahead tree factorization, i.e. computing $\pi_v(s)$, each node (or phoneme arc) of a lexical tree copy corresponds to the maximum bigram probability over all words that are reachable from this specific node with predecessor word v . Thus, we incorporate the factored LM probabilities $\pi_v(s)$ into the dynamic programming recursion across phoneme boundaries:

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

where \bar{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 1. To compute the start-up score $H(w, t)$, we have the dynamic programming equation:

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

Strictly speaking, this equation has to be modified when the word end represented by the state S_w is associated with a *word interior* node of the tree. This happens if a word is at the same time a prefix of another word. As a result of this LM look-ahead, we can use a tighter pruning threshold in the acoustic pruning as the recognition experiments will show.

When computing all entries of the table $\pi_v(s)$ beforehand, we have to keep a huge table in main memory. In our recognition experiments, the lexical tree consists of 63 000 phoneme arcs which are made up from an inventory of 4688 context dependent phoneme models for the 20 000-word NAB task. Therefore, about $20\,000 \cdot 63\,000$ LM factored probabilities would have to be stored. Since the size of this table is prohibitive, we use a different approach. The main idea is to calculate the LM factored probabilities on demand, i.e. only for those tree copies for which active state hypotheses exist. To reduce the memory and computational cost, this approach of on-demand calculation is further refined by additional steps which we describe in detail:

- *Compression of the lexical tree:* To reduce the memory requirements, we first generate a compressed LM look-ahead tree by eliminating those arcs of the lexical tree that only have one successor arc. The construction of this compressed tree can be performed in a pre-processing step. A further reduction of the memory cost can be achieved without significant loss in the recognition accuracy if we only consider the first 3-4 arc generations of the lexical tree.
- *Factorization of the LM look-ahead tree:* The LM look-ahead tree probabilities are computed only for those tree copies that are hypothesized during the search process. To compute these look-ahead probabilities efficiently on demand, we use a backward dynamic programming scheme. We initialize the leaves of the LM look-ahead tree with the bigram language model probabilities, e.g. $p(w|v)$. Then the LM factored probabilities are propagated backwards from the tree leaves to the tree root. For each node, the successor node with maximum look-ahead probability is selected.

3. PHONEME LOOK-AHEAD

The main idea of the k -phoneme look-ahead is to estimate the probability of the k phonemes which are based on the observations of

the next Δt time frames [5]. This probability estimate is then incorporated into the acoustic pruning strategy. To formulate the phoneme look-ahead pruning criterion for $k = 1$ phoneme, we introduce the quantity $q_{LA}(S_\psi; t, t + \Delta t)$ describing the probability that the phoneme ψ ending in state S_ψ produces the acoustic vectors $x_{t+1}, \dots, x_{t+\Delta t}$. The calculation of the so-called look-ahead score $q_{LA}(S_\psi; t, t + \Delta t)$ can be expressed by the dynamic programming recursion:

$$q_{LA}(S_\psi; t, t + \Delta t) = \max_{\tau} \left\{ \max_{\sigma} \{ (\hat{q}_{LA}(S_\psi; t, \tau))^{\frac{\Delta t}{\tau-t+1}} \}, \max_{\sigma} \{ \hat{q}_{LA}(\sigma; t, t + \Delta t) \} \right\},$$

where the auxiliary quantity $\hat{q}_{LA}(s; t, \tau)$ is defined as:

$$\hat{q}_{LA}(s; t, \tau) = \max_{\sigma} \{ \hat{q}_{LA}(\sigma; t, \tau - 1) \cdot q(x_\tau, s|\sigma) \}.$$

The term $q(x_\tau, s|\sigma)$ describes the product of emission and transition probabilities of the underlying HMM. Thus, any arbitrarily successor phoneme ψ of predecessor phoneme ϕ in the lexical tree with predecessor word v will be activated if:

$$Q_v(t, S_\phi) \cdot q_{LA}(S_\psi; t, t + \Delta t) > f_{LA} \cdot Q_{LA}(t),$$

where f_{LA} is the so-called phoneme look-ahead pruning threshold and

$$Q_{LA}(t) = \max_{(s,v)} \{ Q_v(t, s) \} \cdot \max_{\psi} \{ q_{LA}(S_\psi; t, t + \Delta t) \}.$$

In the case of the 2-phoneme look-ahead the equation of the pruning criterion is more complicated, since we have to take into account the factored LM probability across the phoneme boundary within the phoneme look-ahead.

However, we use context dependent (CD) phoneme models than context independent (CI) phoneme models in the detailed search process. Therefore, the calculation of the phoneme look-ahead probability estimate must be done in an efficient way. We reduce the computational effort of the phoneme look-ahead by the following approximations:

- Instead of CD-phoneme models, we use CI-phoneme models, say 40 – 50, in the phoneme look-ahead;
- We simplify the structure of the underlying Hidden Markov Model by collapsing the 6-state standard HMM into a 1-state HMM;
- We reduce the number of laplacian mixture densities used in the phoneme look-ahead.
- The calculation of the phoneme look-ahead is performed every second time frame.

4. EXPERIMENTAL RESULTS

The experimental analysis of both look-ahead techniques were performed on the North American Business (NAB, Nov.'94) H1 development corpus. The test set contains 310 sentences with 7387 spoken words. In the experiments, we used a 20 000-word vocabulary and a bigram language model with a test set perplexity of

Table 1: Effect of the LM look-ahead and 1-phoneme look-ahead on the search effort and recognition results for the NAB'94 H1 development set (20 speakers, 310 sentences, 7387 spoken words) using a bigram language model ($PP_{bi} = 198.4$).

type of LM look-ahead	type of phoneme look-ahead	LM look-ahead tree			search space			WER[%]	real-time factor
		generations	arcs	trees	states	arcs	trees		
unigram (baseline) ($PP_{uni} = 996.6$)		17	63155	1	16960	4641	32	16.4	95.7
					9443	2599	22	16.8	76.6
bigram ($PP_{bi} = 198.4$)		3	12002	300	3277	924	13	16.5	45.7
		4	18625	300	3263	922	13	16.5	45.9
		17	29270	300	3312	935	13	16.5	46.5
	1-state HMM				2013	483	12	16.6	25.8
	6-state HMM				1579	378	12	16.6	19.7

$PP_{bi} = 198.4$. 199 of the spoken words were not part of the vocabulary. The corresponding lexical tree of the 20 000-word vocabulary consists of 63 155 phoneme arcs which are distributed over 17 arc generations. The training of the emission probability distributions was carried out on the WSJ0 and WSJ1 training data as described in [4].

Table 1 shows the results for various language model and phoneme look-ahead types. The pruning parameter were adjusted as described in [9]. In an initial experiment, we performed tests with a unigram LM look-ahead described in [8]. Comparing these results with the results of the bigram LM look-ahead for different numbers of arc generations of the LM look-ahead tree, namely 3,4 and 17 (which is the full LM look-ahead tree), we see that the search space can be reduced by a factor of 5 without significant loss in the recognition accuracy. Finally, we tested the combination of bigram LM look-ahead and 1-phoneme look-ahead as shown in Table 1. The combination of both look-ahead pruning methods further reduces the search effort by a factor of 2. Using this result, the search required 19.7 times real time on a SGI workstation (Indy R4400) instead of a real-time factor of 95.7.

In the full paper, we will present recognition results for a 2-phoneme look-ahead.

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. X. Aubert, C. Dugast, H. Ney, V. Steinbiss: Large Vocabulary, Continuous Speech Recognition of Wall Street Journal Corpus. Proc. IEEE Int. Conf. on Acoustics, Speech and Signal Processing, Adelaide, Australia, Vol. II, pp. 129-132, April 1994.
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. C. Dugast, R. Kneser, X. Aubert, S. Ortmanns, K. Beulen, H. Ney: Continuous Speech Recognition Tests and Results for the NAB'94 Corpus. Proc. ARPA Spoken Language Technology Workshop, Austin, TX, pp. 156-161, January 1995.
5. 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.
6. H. Ney: Search Strategies for Large-Vocabulary Continuous-Speech Recognition. NATO Advanced Studies Institute, Bunion, 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.
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.
9. S. Ortmanns, H. Ney, A. Eiden: Language-Model Look-Ahead for Large Vocabulary Speech Recognition. To appear in Proc. Int. Conf. on Spoken Language Processing, Philadelphia, PA, October 1996.