Large-Vocabulary Continuous Speech Recognition Using Linear Lexicon Search and 1-Best Approximation Tree-Structured Lexicon Search

Norihide Kitaoka, Nobutoshi Takahashi, and Seiichi Nakagawa

Toyohashi University of Technology, Toyohashi, 441-8580 Japan

SUMMARY

The computational cost of a large-vocabulary continuous speech recognition system based on HMM is proportional to its vocabulary size. A tree-structured lexicon is generally used to reduce the number of HMM states. An approximation of the dependence of word boundaries and likelihoods on word histories is also used to limit the increase in the number of hypotheses in the forward decoding procedure. We first compared search algorithms with a tree-structured lexicon using certain approximation methods and algorithms with a linear lexicon. The algorithm based on 1-best approximation with a tree-structured lexicon is efficient but frequently misses the optimal sentence hypothesis. Linear lexicon search can find the optimal hypothesis but has a high computational cost. Thus, we propose a search method using 1-best approximation tree-structured lexicon search and linear lexicon search. The words expanded in the linear lexicon are dynamically selected according to likelihood. We evaluated this new search algorithm using read speech, broadcast news speech, and lecture speech, and obtained significant improvement of recognition performance, expanding only 250 to 500 words out of 20,000 words in the linear lexicon.

Key words: large-vocabulary continuous speech recognition; search; tree-structured lexicon; 1-best approximation; linear lexicon.

1. Introduction

Large-vocabulary continuous speech recognition (LVCSR) based on hidden Markov model (HMM) acoustic modeling and N-gram language modeling has been developed to a practical level and applied to practical systems such as dictation and broadcast news captioning [1, 2], and further improvement of the technology will enable it to be used more widely.

HMM-based LVCSR generally has a computational cost proportional to its vocabulary size if all HMM states of the words in the vocabulary are expanded in memory. A tree-structured lexicon in which prefixes expressed in subword units such as phonemes or syllables are shared among words is often used to reduce the number of HMM states [3].

The LVCSR system expands partial sentence hypotheses by concatenating possible succeeding words, but this expansion makes the number of hypotheses grow exponentially. To avoid this, some approximations based on word histories are often used, such as 1-best approximation and word-pair approximation [4]. In 1-best approximation, only the 1-best hypothesis is expanded to new hypotheses in every frame. Under this assumption, the system has only
a tree-structured HMM network statically. On the other hand, under the latter assumption, word history-dependent tree-structured HMM networks must be expanded dynamically in memory. Here, we discuss recognition algorithms using HMMs and bigram language models. The 1-best approximation can optimize the word boundary of the new hypothesis with the maximum acoustic score, but those of the other hypotheses cannot be guaranteed to be optimized. Furthermore, the tree-structured lexicon prevents accurate language look-ahead and thus the optimal hypothesis may be lost when integrating acoustic and language scores. In contrast, the word-pair approximation guarantees retention of the optimal hypothesis in the search. These characteristics show that word-pair approximation is a more exact algorithm than 1-best approximation in searching for better hypotheses, but the word-pair approximation has a much higher computational cost than the 1-best approximation. To overcome these drawbacks, some systems expand tree-structured HMM networks according to the \(N\)-best histories [5, 8]. Ogata and Ariki [6] have proposed to construct tree-structured HMM networks dynamically for words with bigram probabilities given a certain history, and to use a static HMM network of the words whose probabilities were generated by unigram back-off. This method depended on the fact that the average number of words with bigram probabilities was about 100. However, these methods have the problem that the system cannot perform accurate look-ahead and thus cannot reduce the search space.

Some pruning algorithms are also used to reduce the number of hypotheses. We consider time-synchronous beam search. In this method, the hypothesis scores are obtained by integration of the acoustic and language scores, and hypotheses with low scores are pruned at each time frame. When using a tree-structured lexicon, the root node of the tree is connected to each hypothesis and new hypotheses are generated, but the root node and the nodes of the prefixes are shared among words, so that the system cannot determine the corresponding word or apply bigram probability to the new hypotheses in advance. This means that the system cannot use language look-ahead. The factorization method [10] is often used to alleviate the lack of language look-ahead, but the constraints of language information are still weak.

We developed the SPOJUS large-vocabulary continuous speech recognizer using the method of expanding tree-structured HMM networks according to the \(N\)-best histories [8], but the current version of SPOJUS is based on the 1-best approximation.

In this paper, we propose a new efficient and accurate search algorithm using 1-best approximation tree-structured lexicon search and linear lexicon search. Linear lexicon search can apply language look-ahead exactly to each hypothesis, and thus compensates the drawbacks of 1-best approximation and tree-structured lexicon search.

### 2. Conventional Methods

#### 2.1. Linear lexicon and tree-structured lexicon

HMM-based LVCSR increases in computational cost as the vocabulary size becomes larger. When using a linear lexicon as illustrated in Fig. 1(a), the number of HMM states increases in proportion to the vocabulary size, and as a consequence the computational cost is also proportional to the vocabulary size. The tree-structured lexicon illustrated in Fig. 1(b) is conventionally used to reduce the number of HMM states.

When using \(N\)-gram language models, the likelihood of a hypothesis at time frame \(i\) which is generated by connecting a word \(w\) to the word sequence \(W\) is

\[
P_{W,w}(i) = \max_j \{ P_{W}(j) + Q_w(j, i) + \text{lang}(w|W) \} \tag{1}
\]

where \(j\) satisfies \(j < i\), \(P_{W}(j)\) is the sum of the likelihoods of the hypothesis \(W\) at time \(j\), \(Q_w(j, i)\) is the acoustic likelihood of word \(w\) for period \([j, i]\), and \(\text{lang}(w|W)\) is the language probability of word \(w\) given the word history \(W\). If we use a linear lexicon, \(\text{lang}(w|W)\) can be considered at the time of connection of \(w\) and \(W\), but if the lexicon is structured as a tree, the root node and many other nodes corresponding to word prefixes are shared among words, so that \(\text{lang}(w|W)\) cannot be added to \(P_{W,w}\) at that time point.

#### 2.2. Approximations of word history dependency for tree-structured lexicon search

The likelihood and the boundary of a word depend on the word history. Thus, we must perform forward decod-
ing independently for all new hypotheses generated by concatenating a word to all possible hypotheses. So many hypotheses are generated in this process that we must make an assumption about the word history dependency and bundle the hypotheses using an approximation based on the assumption.

1-best approximation assumes that word boundaries do not depend on the histories, and word-pair approximation assumes that each word boundary depends on only one preceding word. N-best approximation [5, 8] is an implementation of word-pair approximation which expands the N-best hypotheses to new hypotheses at each time frame.

The word-pair approximation assumes that the word boundaries depend on just one previous word. Under this assumption, Viterbi forwarding is executed on previous word-dependent HMM networks. This means that HMM networks must be copied according to the word histories and that Viterbi search must be executed on the networks in parallel as illustrated in Fig. 2(a).

We consider the combination of bigram language models and HMM acoustic models. Using tree-structured HMM networks, the language probability \( P(w_k|w_{k-1}) \) is multiplied by the acoustic score with an appropriate weight at the leaf node of the network, because the word for the node is uniquely identified only at the leaf. Each network has a unique word history under word-pair approximation, so that \( P(w_k|w_{k-1}) \) can be exactly applied.

On the other hand, 1-best approximation does not assume any word-history dependence. This approximation reduces the computational cost greatly because the system does not need copies of the HMM network and uses only one reentrant network, as illustrated in Fig. 2(b). We can determine the preceding word (that is, the word history) by back-tracing from the leaves of the network and thus we can apply the bigram probability \( P(w_k|w_{k-1}) \). Even if only one hypothesis is expanded to new hypotheses on the HMM network in every time frame, we can approximately obtain other hypotheses by keeping other histories \( w_{k-1} \)s at each frame and compensating the language scores by the differences between \( P(w_k|w_{k-1}) \) and \( P(w_{k+1}|w'_{k-1}) \). But this approximation is not correct. The HMM network is shared by the hypotheses expanded at all time frames and thus many paths are overridden and eliminated by the other path. Therefore, a path which is optimal when considering the language score that can be applied at the leaf node may be rejected by this mechanism. In this way, the optimal word sequence is not guaranteed to be obtained by tree-structured lexicon search under the 1-best approximation.

We consider the combination of bigram language models and HMM acoustic models. Using tree-structured HMM networks, the language probability \( P(w_k|w_{k-1}) \) is multiplied by the acoustic score with an appropriate weight at the leaf node of the network, because the word for the node is uniquely identified only at the leaf. Each network has a unique word history under word-pair approximation, so that \( P(w_k|w_{k-1}) \) can be exactly applied.

In essence, SPOJUS uses the 1-best approximation. Word boundaries and scores depend on the word histories, and the hypotheses must be expanded separately according to the histories, generating too many hypotheses. With a tree-structured lexicon with 1-best approximation, approximate language scores obtained by a factoring method can be used, but this approximation affects the best-hypothesis search, so that we may not be able to obtain the best hypothesis.

Shibata and Kobayashi [11] investigated the effects of treatment of word history in the context of a one-pass decoding algorithm with a trigram language model. The approximate search algorithms which “bundle” several word histories into one are called “bundle search” algorithms [12]. Two previous words must be considered to prevent bundling of histories, but this method needs copies of the tree-structured HMM network according to the two-word histories. No copies of the network are needed if all the word histories are bundled, but a factoring method cannot be used. Shibata and Kobayashi [11] proposed to bundle hypotheses according to the first words of the two-word histories and to use bigram factoring. Ogata and Ariki [6] proposed a method in which trees were dynamically expanded only for the words which had bigram probabilities. Static trees were also used for unigram back-off. This method guaranteed the optimal search for the best hypothesis. Hori and colleagues [7] assumed that the word boundary depended only on the preceding short phoneme sequence instead of the preceding word. This method has a smaller computational cost than word-pair approximation and also facilitates word boundary approximation. But both methods still have much higher computational costs than the 1-best approximation.
2.3. Recognition using linear lexicon

We next compare linear lexicon search and tree-structured lexicon search. Linear lexicons for vocabulary words are connected with each other from tails to heads in an HMM network. Using this network, subword HMMs are not shared among words, unlike the tree lexicon, and they determine the word associated with each node. This enables the system to apply the bigram language scores at the first node of each word, whereas the tree-structured lexicon does not. This means that the system can know not only the word history but also the succeeding word at the beginning of the HMM of the succeeding word, and can use language look-ahead, resulting in efficient constraints on the search space.

Figure 3(a) shows new word attachment to hypotheses using a linear lexicon. The word “a-sa-ga-o” is attached to hypothesis D which maximizes the sum of the total score of hypothesis D and the language score \(P(a-sa-ga-o|D)\). In contrast, the HMM network is attached to hypothesis A in the case of a tree-structured lexicon, as illustrated in Fig. 3(b), because of its highest likelihood among the hypotheses not considering word probabilities when given the histories.

The HMM trellis is shared among words and histories in the 1-best approximation method, and thus conflicts often occur. The HMM nodes in linear lexicons are also shared among histories, and thus the optimal \(N\)-best hypotheses are not guaranteed to be obtained, but there are no conflicts among words and the first best hypothesis is guaranteed to be obtained.

But a linear lexicon does not allow node sharing among words and thus the computational cost is much higher than that for a tree-structured lexicon.

The problems of the method described above are summarized in Table 1. We propose a parallel use of linear lexicon search and 1-best approximation tree-structured lexicon search.

3. Dynamic Expansion of Linear Lexicon

When using bigram language models, the probabilities of word pairs which are not found or rarely found in the training data are smoothed by unigram back-off. The combination of bigram and 1-best approximation search on a tree-structured lexicon misses the optimal search path, but unigram search does not. The method proposed in Ref. 6 uses these characteristics well, but this method cannot yet use language look-ahead. We therefore propose a new search method using search on a dynamically expanded linear lexicon along with 1-best approximation search on a tree-structured lexicon. That is, we basically use 1-best approximation tree-structured lexicon search, and also use linear lexicon search in parallel. Only a small number of words which are dynamically selected are expanded in a linear lexicon network. For a linear lexicon, language probability can be applied at the beginning of the HMM of each word and thus the search space can be reduced.

If all words which have bigram probability are expanded to the linear HMM network, the number of the words expanded in the linear network becomes large. We therefore constrain the number of words. We evaluate all
the vocabulary words in every time frame by using the following equations:

\[ Q_{la}(w, t) = \max_v (P_{hyp}(v, t) + \text{lang}(w|v)) \]  \hspace{1cm} (2)
\[ Q_{lin}(w, t) = \max_{s_w} (P_{state}(s_w, t)) \]  \hspace{1cm} (3)

where \( P_{hyp}(v, t) \) is the likelihood of the final HMM state of the hypothesis which ends with the word \( v \), \( P_{state}(s_w, t) \) is the likelihood of the HMM state \( s_w \) in word \( w \) which is expanded on a linear HMM network at the time frame \( t \), and \( \text{lang}(w|v) \) is a look-ahead probability which is the appearance probability of word \( w \) given the word history \( v \). \( Q_{la}(w, t) \) in Eq. (2) denotes the maximum value among the sums of the likelihoods of the preceding word sequence ended by word \( v \) in a tree-structured network and the look-ahead probability. When \( \text{lang}(w|v) \) is obtained by unigram back-off, not by bigram, the word \( w \) need not to be expanded in the linear network, and thus \( Q_{la}(w, t) \) is calculated for the words which have bigram probabilities for at least one of the existing histories. \( Q_{lin}(w, t) \) in Eq. (2), which denotes the maximum value among the HMM states of the word \( w \), is calculated for the word \( w \) which has already been expanded in the linear HMM network. If a word \( w \) has both \( Q_{tree}(w, t) \) and \( Q_{lin}(w, t) \), then the maximum value of \( Q_{tree}(w, t) \) and \( Q_{lin}(w, t) \) is selected as the evaluation value for the word.

\( N_{lin} \)-best words are selected based on the values. The words selected and not yet expanded on the linear network will be expanded. The words selected and already expanded on the network will be kept on the network. If a word on the linear network is not selected, it will be removed from the network. Using a linear lexicon, the optimal path lost by 1-best approximation search on the tree-structured lexicon will be likely to be kept. On the other hand, paths with low linguistic probability will be pruned out from the linear network without acoustic evaluation because of language look-ahead, but such paths also have a chance to be kept on the tree-structured network.

### 4. Combinational Use of Dynamically Expanded Linear Lexicon and Static Tree-Structured Lexicon

We evaluated the combination of efficient 1-best approximation tree-structured lexicon search and exact linear lexicon search.

#### 4.1. Experimental conditions

We used 100 sentences from read Japanese newspaper speech in the JNAS corpus, 175 sentences from NHK broadcast news speech, and lecture speech of four males (A01M0074, A01M0035, A01M0007, A05M0031) from the CSJ corpus (Corpus of Spontaneous Japanese). The speech data were sampled with a sampling frequency of 16 kHz, and the signal was preemphasized by a factor of 0.98. A Hamming window 25 ms long was applied and shifted with a step of 10 ms. Thirty-eight-dimensional feature vectors were used, including 12-dimensional MFCCs, their first and second deviation coefficients, and the first and second deviations of the log power.

One hundred sixteen Japanese context-independent syllable HMMs (strictly speaking, mora HMMs [13]) including short pauses and silence were used. Each continuous density HMM had five states, and four of them had pdfs of output probability. Each pdf consisted of four Gaussians with full-covariance matrices. HMMs with discrete duration probability density (for 1 to 14 frames) were used in all of the experiments, and HMMs with a self-loop were used in some experiments. The HMMs were trained using syllable segments obtained from the utterances of the 503 ATR sentences read by 8 male speakers and the 216 ATR phoneme-balanced words read by 10 males first, and then were trained by MAP concatenated training using 4518 utterances read by 30 males (ASJ corpus) and 23474 utterances read by 175 male speakers (JNAS corpus). We used a two-pass decoding method, which recognized each test utterance using bigram and context-independent HMMs in the first pass and then rescored the 200-best hypotheses generated by the first pass to generate the final result in the second pass. Two types of rescoring were used: language rescoring using trigrams, and language and acoustic rescoring using trigram and context-dependent HMMs [13] (914 models) dependent on the preceding vowels. The text of 45 months of the newspaper Mainichi comprising 90 million words of read speech, and scripts of NHK broadcast news comprising 2.4 million words of broadcast news speech, were used to train the bigram and trigram language models. For CSJ, we used the bigrams and trigrams made by Kyoto University and provided by the CSJ corpus. All of the language models had approximately 20k words. The beam width at the word end was set to 30 according to preliminary experiments.

We used correctness (Cor.) and accuracy (Acc.) as measures in evaluation:

\[
\text{Cor.} = \frac{\# \text{Cor.} - \# \text{Sub.} - \# \text{Del.}}{\# \text{Cor.}} \]  \hspace{1cm} (4)
\[
\text{Acc.} = \frac{\# \text{Cor.} - \# \text{Sub.} - \# \text{Del.} - \# \text{Ins.}}{\# \text{Cor.}} \]  \hspace{1cm} (5)
where #Cor., #Sub., #Ins., and #Del. are the numbers of words correctly recognized, of substitution errors, of insertion errors, and of deletion errors, respectively.

4.2. Recognition experiments using linear lexicon search

We compared linear lexicon search and 1-best approximation tree-structured lexicon search. The results for read speech are shown in Table 2. In this experiment, we used only language rescoring in the second pass. “Optimal rescoring” means the case in which the best hypothesis is selected from each 200-best list by the second pass, that is, “oracle.” Comparing the results of the first pass, 1-best approximation tree-structured lexicon search was inferior to linear lexicon search. We can find the same relation in the results of optimal rescoring, and thus the final results had also a performance difference between these two methods. On the other hand, the search on a linear lexicon, which had approximately 78k HMM nodes, was less efficient than that on a tree-structured lexicon with approximately 26k HMM nodes.

4.3. Recognition experiments with dynamic linear lexicon expansion

We evaluated the combination of a dynamically expanded linear lexicon and a static tree-structured lexicon described in Section 4. The results for various numbers of words expanded in the linear network in every time frame \(N_{\text{lin}}\) are summarized in Fig. 4 and the details are shown in Table 3. The real time factors (xRT) are also shown in the table for comparison of the computational costs.

All words were expanded in the linear network in the case of “\(\infty\).” In this case, no search errors occurred except for errors caused by beam search (that is, optimal search was performed), and thus the results were the upper bounds of our models (acoustic models and language model). We used language rescoring in the second pass. We employed a Linux PC with a Pentium IV 2.2-GHz CPU to measure the recognition speed.

We find that recognition performance was improved by using a larger number of words expanded in the linear network. We achieved almost the same recognition performance with \(N_{\text{lin}} = 250\) as the upper bound. In the case of \(N_{\text{lin}} = 250\), the computational cost was only about 10% larger than \(N_{\text{lin}} = 0\) (which means conventional 1-best approximation tree-structured lexicon search). Thus, the setting \(N_{\text{lin}} = 250\) was appropriate.

Such performance may be achieved only with linear lexicon search. We conducted recognition experiments only with dynamic expansion of \(N_{\text{lin}}\) words on the linear network using Eqs. (2) and (3). Here, Eq. (2) corresponds to lookahead word concatenation in Fig. 3(a). The results are shown in Table 4. We find that we must expand a large number of words on the linear network to obtain perform-

![Fig. 4](image-url)
ance comparable to the optimal performance by expanding all of the words on the linear network. This result showed that the combined use of static tree-structured lexicon and dynamic linear lexicon was effective.

We evaluated the proposed method with broadcast news speech and lecture speech. The results are shown in Tables 5 and 6. The large variation of utterance speed in the lecture speech cannot match the duration distributions of HMMs trained from read speech. We also evaluated the proposed method with self-loop HMMs for lecture speech. The proposed method also improved the recognition performance in these tasks.

4.4. Using acoustic rescoring

We applied acoustic rescoring to the $N$-best results of the proposed method. We evaluated it with read speech and broadcast news speech. The results are shown in Tables 7 and 8. Dynamic linear lexicon expansion and acoustic rescoring additively improved recognition performance. Dynamic linear lexicon expansion generated accurate $N$-best hypotheses and thus the acoustic rescoring worked well.

5. Conclusions

In this paper, we have discussed some methods of approximation of word histories when using the tree-structured lexicon and bigram language model, and also compared these methods with recognition method using a linear lexicon. Based on the findings, we propose a combination of a linear lexicon and a tree-structured lexicon for efficient and accurate large-vocabulary continuous speech recognition. We evaluated the method using read speech, broadcast news speech, and lecture speech, and showed that the small number of words dynamically expanded in the linear lexicon improved recognition performance so that it nearly reached the upper bound.

Our method can almost always obtain the optimal 1-best hypothesis, but it cannot obtain the optimal $N$-best...
hypotheses. By adopting a lattice search method into back-tracing [4], we can obtain more accurate multiple hypotheses in the first pass, so that the language rescoring in the second pass will work better.

REFERENCES


AUTHORS (from left to right)

Norihide Kitaoka (member) graduated from the Department of Information Science, Kyoto University, in 1992, completed the M.E. program in 1994, and joined Denso Corporation, where he engaged in research on speech recognition at Denso Research Laboratories. From 1997 to 2000 he was enrolled in the second half of the doctoral program at Toyohashi University of Technology. In 2001, he joined the faculty of Toyohashi University of Technology as a research associate. He is now a lecturer in the Department of Information and Computer Sciences. He holds a D.Eng. degree, and is a member of the Acoustical Society of Japan, IEICE, and the Information Processing Society of Japan.

Nobutoshi Takahashi graduated from the Department of Information and Computer Sciences, Toyohashi University of Technology, in 2001 and completed the M.E. program in 2003. His research interest is speech recognition algorithm.
Seiichi Nakagawa (member) completed the doctoral program at Kyoto University in 1976 and joined the Department of Information Engineering. He moved to Toyohashi University of Technology in 1980, and has been a professor in the Department of Information and Computer Sciences since 1990. He was a visiting researcher at Carnegie-Mellon University from 1985 to 1986. His research interests are speech information processing, natural language processing, and artificial intelligence. He holds a D.Eng. degree. He received 1997/2001 Paper Awards from IEICE and an IETE Paper Award in 1988. He is the author of *Speech Recognition by Probabilistic Models* (IEICE), *Fundamentals and Applications of Information Theory* (Kindai Kagakusha), and *Pattern Information Processing* (Maruzen Co.), and a coauthor of *Speech, Auditory Perception, and Neural Network Models* (Ohm Press).