AN EFFICIENT LEXICAL TREE SEARCH
FOR LARGE VOCABULARY CONTINUOUS SPEECH RECOGNITION

J.Ogata and Y.Ariki
Department of Electronics and Informatics
Ryukoku University, Seto, Otsu-shi, Shiga, 520-2194, Japan
ogata@arikilab.elec.ryukoku.ac.jp

ABSTRACT
This paper describes an efficient search algorithm for a high speed and high accuracy LVCSR system. A conventionally used lexical tree search is an efficient method, but has a problem in incorporating the language probability. To solve this problem, we propose in this paper a new efficient search algorithm incorporating the language model structure. In our developed LVCSR, 2-pass search algorithm is adopted to produce a word graph as an intermediate expression. The experimental results on the 20,000-word Japanese dictation task showed that the proposed method can reduce approximately half of the processing time without increasing any errors.

1. INTRODUCTION
In order to reduce the computational cost, lexical tree search is effective in time-synchronous beam search decoding[1]. However, there is a problem that the language probability can not be uniquely determined until the search arrives at the word end (leaf) nodes in the lexical tree. To solve this problem, N-gram factorization (N-gram lookahead) is normally used (Here, we describe about bigram case)[1][2][3]. In the lexical tree search, bigram factorization is a crucial problem in computational cost or memory requirements. Therefore, in [3], an efficient search algorithm using a static network exploiting the bigram structure is proposed.

In this paper, we propose more efficient search methods by expanding those mentioned above, and evaluate them. The conventional and proposed methods are described as follows:

- All-word back-off connection (conventional)
  In search process, when the search arrives at the word end nodes, its successor tree as well as a back-off tree are connected.

- Without back-off connection
  In search process, when the search arrives at the word end nodes, the back-off tree is not connected and only the successor tree is connected.

- Best-word back-off connection
  The back-off tree is only connected to the predecessor word with the best partial score at each frame.

This paper is organized as follows. In section 2., we give a brief review of our 2-pass LVCSR system and explain about some problems mainly treated in this paper. In section 3., we propose efficient search methods incorporating back-off connection. In section 4., the recognition experiments have been carried out on the 20K-word Japanese dictation task.

2. BASELINE LVCSR SYSTEM
In this section, we briefly describe the 2-pass decoder as a baseline system (Figure 1). At the 1st-pass, we adopted the lexical tree search using a bigram language model for constructing the word graph. Then, at the 2nd-pass, the best sentence is computed in the word graph using a trigram language model.

Figure 1: Overview of 2-pass decoder
2.1. Lexical Tree Search (1st-pass)

We focus on a search algorithm using a bigram language model at the 1st-pass in the 2-pass search strategy. The conventional lexical tree search can reduce an acoustical search space but language probability cannot be uniquely determined until the search arrives at the word end (leaf) nodes in the lexical tree. To solve this problem, bigram factorization (bigram lookahead) is normally used. For bigram language model, the factored LM probability \( \pi_v (s) \) for state \( s \) and predecessor word \( v \) is defined as:

\[
\pi_v (s) := \max_{w \in W(s)} p(w|v)
\]

where \( W(s) \) is the set of words that can be reached from tree state \( s \)[2]. For implementation of the bigram factorization, there are two methods. One is a dynamic computation of the maximum bigram value during the search process. In this method, the computational cost of bigram factorization is too high. The other is a static computation of the maximum bigram table before the search process. In this method, since the size of this table is too huge, the implementation of the bigram factorization is not feasible in terms of the memory requirements.

In this study, we adopted latter method. However, in order to reduce the memory requirements, a static network is used[3] which exploited the special structure of the bigram language model as described in subsection 3.1.

2.2. Word Graph Rescoring (2nd-pass)

At the 2nd-pass, word graph rescoring is carried out at the word level using dynamic programming algorithm. The computational cost of the rescoring is typically small compared with the 1st-pass. Therefore, in this paper, we focus on the efficient search method at the 1st-pass.

3. IMPROVED LEXICAL TREE SEARCH

3.1. All-word Back-off Connection

In \( N \)-gram language modeling, two basic computation schemes are generally employed: back-off and interpolation. In this study, we use a back-off bigram model for beam search decoding. The structure of back-off bigram is divided into two parts: bigram section (a group of words with bigram probability) and back-off section (a group of words with back-off smoothed probability based on unigram probability).

In the search process, when the search arrives at the word end nodes, its successor tree as well as a back-off tree are connected[3]. Here, “successor tree” indicates a tree organized for a bigram section, and “back-off tree” indicates a whole lexical tree for a back-off section. This type of search method is called “all-word back-off connection (Figure 2)”.

\[ \text{Figure 2: All-word back-off connection} \]

3.2. Without Back-off Connection

In a back-off smoothing method, some probabilistic value may be assigned to grammatically incorrect word sequence. This causes the decoding errors by the back-off connection.

From this viewpoint, we propose a network structure that the back-off tree is not connected and only the successor tree is connected. This kind of search method is called “without back-off connection (Figure 3)”. In this “without back-off connection”, the correct word sequence which can be originally recognized by back-off connection cannot be recognized any more, but the grammatically incorrect word sequence caused by the back-off connection can be excluded. In addition, comparing with the all-word back-off connection, the search process of this method is more efficient because the number of active nodes is small at the word end.

3.3. Best-word Back-off Connection

We propose here a more efficient search method even by incorporating the back-off connection.

In the all-word back-off connection, the language probability in the back-off tree is not affected by the predecessor word because the language probability in back-off tree is estimated using unigram probability. From this viewpoint, a new fact is discovered that the partial score until the word end (leaf) node in the back-off tree depends only on the score of the predecessor word end node. Consequently, following new search algorithm, called “best-word back-
off connection (Figure 4)”, is derived: the back-off tree is connected to the predecessor word with the best partial score at each frame.

This method can solve a problem that the back-off connected word can not be recognized, which was the problem in the without back-off connection. In addition, comparing with the all-word back-off connection, the number of active nodes is small because the number of the connection to the back-off tree is only one per frame.

Table 1: Acoustic analysis and HMM

<table>
<thead>
<tr>
<th>Parameter</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Sampling frequency</td>
<td>16kHz</td>
</tr>
<tr>
<td>Feature parameter</td>
<td>MFCC (39 dimensions)</td>
</tr>
<tr>
<td>Analysis frame length</td>
<td>20ms</td>
</tr>
<tr>
<td>Analysis frame shift</td>
<td>10ms</td>
</tr>
<tr>
<td>Analysis window</td>
<td>Hamming window</td>
</tr>
<tr>
<td>Number of states</td>
<td>5</td>
</tr>
<tr>
<td>Number of loops</td>
<td>3</td>
</tr>
<tr>
<td>Number of mixtures</td>
<td>12/state</td>
</tr>
</tbody>
</table>

4. EXPERIMENTS

4.1. Experimental Condition

Experiments were performed on 20K-word Japanese dictation task. The training of the acoustic model (triphone HMM) was carried out on the JNAS (Japanese Newspaper Article Sentences) corpus[4]. Table 1 shows the experimental conditions for acoustic analysis and HMM. Language model trained by the Mainichi Newspaper articles is available from Japanese dictation toolkit[5]. Evaluation data was 100 sentences spoken by 23 male speakers taken from JNAS corpus.

In the evaluation, we used three measures: WAC (Word Accuracy), GAC (Graph Accuracy) and RTF (Real Time Factor). GAC is computed as WAC of sentence which best matches the spoken sentence in th word graph. These experiments were performed on a workstation, SGI Origin200 with an R12000 250MHz CPU.

4.2. Experimental Results

The evaluation was carried out with various beam width. The experimental results for each method are shown in Figure 5, 6 and Table 2. In Figure 5, 6, “best” indicates the best-word back-off connection, “all” indicates the all-word back-off connection, “without” indicates the without back-off connection.

In comparison with the all-word back-off connection and without back-off connection, the without back-off connection is lower than the all-word back-off connection in terms of WAC and GAC. It is caused depending on whether the back-off connected word sequence could be recognized or not. Actually, investigating the number of back-off connected word pair, 2.9% (57/1974) of all the word pairs are back-off connected word pairs. 72% (41/57) of them were recognized in the all-word back-off connection. In contrast, the back-off connected word pairs could not be recognized at all in the without back-off connection. However, from the viewpoint processing time, the without back-off connection is faster than the all-word back-off connection by a factor of 2.

In the best-word back-off connection, there is no loss in word accuracy comparing with the all-word back-off connection. Actually, the back-off connected word pairs were recognized to the same degree as the all-word back-off connection. From this viewpoint, it can be said that back-off connected words can be recognized enough by connecting
the word with the best partial score to the back-off tree at each frame. As for processing time, the best-word back-off connection is faster than the all-word back-off connection by a factor of 2. Therefore, it can be said that the proposed method (best-word back-off connection) can reduce approximately half of the processing time without increasing any errors.

Table 2: Comparative recognition results (all: all-word back-off connection, without: without back-off connection, best: best-word back-off connection)

<table>
<thead>
<tr>
<th></th>
<th>all</th>
<th>without</th>
<th>best</th>
</tr>
</thead>
<tbody>
<tr>
<td>WAC</td>
<td>94.0%</td>
<td>91.3%</td>
<td>94.0%</td>
</tr>
<tr>
<td>GAC</td>
<td>96.6%</td>
<td>94.8%</td>
<td>97.1%</td>
</tr>
<tr>
<td>RTF</td>
<td>5.7</td>
<td>2.6</td>
<td>2.6</td>
</tr>
</tbody>
</table>

5. CONCLUSION

This paper has presented an efficient search algorithm for the high speed and high accuracy LVCSR system. We have focused on the efficient search method exploiting the bigram structure. In various investigations, we have proposed search method called “best-word back-off connection” which links the word with the best partial score at each frame to the back-off connection. The experimental results show that this method can reduce about half of the processing time without increasing any errors.

6. REFERENCES