Joint Conference of 1996 International Computer Symposium
December 19~21, Kaohsiung, Taiwan, R.O.C.

A Local-Segment-Based Approach for Spotting Large Vocabulary Chinese Keywords from Mandarin Speech

Bo-Ren Bai, and Lin-Shan Lee

Dept. of Electrical Engineering, National Taiwan University
Taipei, Taiwan, R.O.C.
E-mail: white@speech.ee.ntu.edu.tw

Abstract

This paper presents an efficient local-segment-based approach to spotting large vocabulary Chinese keywords from a spontaneous Mandarin speech utterance. Instead of searching through the whole utterance with the aid of filler models, this approach simply searches through the local segments within the utterance to find out the most possible keywords. The monosyllable-based technique for very large vocabulary Mandarin speech recognition[1][2] was modified and applied to a three-phase framework with a special scoring method for keyword spotting. There are three main advantages in this method. First, the difficulties in training the filler models can be avoided. Second, it is unnecessary to retrain the keyword models when the vocabulary is changed. And third, this approach not only works for small vocabulary problems, but it works for large keyword vocabulary problems as well. To improve the performance of the approach, several additional techniques are also developed to further enhance its speed and accuracy.

Two tasks with completely different characteristics were tested here. The first one has 9 keywords, and each keyword includes 2 syllables. A spotting rate as high as 93.73% is obtained for the test based on this task. The second task has 2611 keywords, and each keyword includes 2 to 20 syllables. A 84.52% spotting rate for the top 10 candidates is attained with a speed requiring only 1.6 times of utterance length when the test was performed on a Sparc 20 workstation.

1. Introduction

Many different algorithms have been adopted for detecting a predefined set of keywords form continuous speech. Most of them are based on Viterbi decoding and require word/sub-word and filler models to decode an input speech into a sequence of keywords and non-keywords. Many different ways have also been developed to adjust the scores of the keywords appearing in the Viterbi alignment. However, it is always difficult to train filler models for the non-keywords speech, and it is even more difficult to model the lower level events such as non-speech noise[3]. In addition, most reported keyword spotting techniques can only deal with small vocabulary problems. Huang, *et. al.*[4] proposed an algorithm to deal with the large vocabulary problems, it needs to carefully design the filler models. In this research, a new strategy is proposed for Chinese keyword spotting: by applying the modified very-large-vocabulary continuous Mandarin speech recognition techniques[2] to detect the keywords directly from the local segments of the speech utterance. The purpose is to deal with the large vocabulary problems without training the filler models.

In Chinese language all the characters are monosyllabic, a word is composed of several characters, and the total number of syllables is relatively small. Taking advantage of this monosyllabic structure, we have achieved a great success in continuous speech Mandarin dictation technique. But when a keyword spotting problem is considered, the ability to deal with spontaneous speech is required. In spontaneous speech, it is always full of lower level events such as pauses, filled pauses (e.g. "uh"), hesitation, laughter and other non-speech noises (inhalation, cough), so it is difficult to recognize such an utterance. After carefully considering the monosyllabic nature of Mandarin speech and the different phenomena between spontaneous speech and

read speech, a three-phase framework with a special scoring method is developed. The first phase is for estimating the possible syllable boundaries, and some obvious phrase boundaries will also be picked out. The second phase is for syllable recognition, so that most possible syllable candidates and tone scores for each local segment can be found. The third phase is for keyword matching, in which the syllable candidates are concatenated to form larger segments which may include keywords.

In the following, in Section 2, we first introduce the three-phase approach in detail. In Section 3, several speedup techniques are presented. In Section 4, the likelihood scoring techniques are adopted to improve the accuracy. Moreover, some experiments are designed and the results are shown in Section 5. Finally, in Section 6, some conclusions are drawn.

2. Three-phase framework

In this section, we will introduce the main algorithm used in this paper. It is a three-phase framework. The first phase is designed for possible syllable boundaries estimation, the second phase for acoustic recognition, including obtaining base syllable candidates and tone scores, and the third one for keyword-matching.

2.1 Phase 1 – Syllable boundaries estimation

For a given input speech utterance, all possible syllable beginning frames can be first obtained by picking up all the dips in the energy contour, such as x, y, z in Figure 1. Corresponding to a beginning frame such as x in Figure 1, the possible ending frames for the syllable such as $y-1$, $z-1$ can then be found by using the estimated minimum and maximum duration of a syllable, e.g. $D_{\min}$ and $D_{\max}$. Besides dips in the energy contour, several other useful features are also used for better estimating the syllable boundary, for example, zero crossing rate, and pause duration. Given all these possible beginning frames and their corresponding ending frames, we can perform acoustic recognition for each segment which may include a syllable.

The syllable boundaries estimated in this phrase can be further distinguished to three kinds of boundaries. For those with higher probability to be syllable boundaries, we will assign them to be hard syllable boundaries. For those with lower probability, we just assign them to be soft syllable boundaries. And the third kind of boundaries are phrase boundaries, which are of course also hard syllable boundaries. Pause duration is very helpful to estimate this kind of boundaries here, and in the later phase, phrase-final lengthening[5] can help to estimate phrase boundaries better. Distinguishing these three kinds of boundaries will be very helpful in improving the speed. This will be sown latter.

2.2 Phase 2 – Acoustic recognition

With all possible syllable beginning frames and their corresponding ending frames, in this phase, the Viterbi search is then performed for each utterance segment which may include a syllable to produce N most possible syllable candidates. Context dependent initial/final models are adopted here, where "initial" is the initial consonant of a syllable, and "final" is everything in the syllable after the "initial", including the vowel or diphthong part plus optional medial or nasal ending[2]. During the syllable recognition process, a syllable is not allowed to cross any hard syllable boundaries mentioned in the previous phase. This constraint will not only help to save much search time, but also improve the recognition accuracy. In addition, tone recognition is handled similarly but separately. By this way, less storage memory is needed to store the acoustic recognition result, also in the later process, it is capable of getting more tonal syllables by combining these two kinds of acoustic scores. With the N most possible syllable candidates and acoustic recognition scores reserved for every segment which may include a syllable, an asynchronous syllable lattice is then constructed which caries all information needed for further process in the later phase. Moreover, since we can first get the mean and standard deviation of every syllable length form the speech database, now more phrase boundaries can be estimated by comparing the recognized syllable length to the statistical syllable length.

2.3 Phase 3 – Keyword matching

The third phase is a keyword-matching process. Given the asynchronous syllable lattice together with their acoustic scores, here we will find out the most possible keywords directly from the local segments of the speech utterance by searching through the syllable

lattice. To tolerate the information loss caused by recognition error, here we adopt a fuzzy matching, instead of exact matching, to pick out the keyword from adequate location.

Since a possible ending frame of a syllable is also a possible ending frame of a keyword, starting from each ending frame, a backward search can be performed to construct a possible keyword by using its component syllables. As shown in Figure 2, the root node x_0 is a possible ending frame, and x_1^1, x_1^2 are the possible ending frames for the preceding syllable. Each of these two branches $(x_0 \rightarrow x_1^1)$ $(x_0 \rightarrow x_1^2)$ may include a syllable, and we have stored the N most possible syllable candidates with their scores and 5 tone scores for each branch at x_0. In the same way, we can find all possible ending frames for the next preceding syllable starting from x_1^1, x_1^2 and so on, until the length of this tree equals the syllable number of the desired keyword. Now every path, from the root node to a leaf node can be matched to the keyword, i.e. the $i-th$ level branch is matched to the last $i-th$ syllable of the keyword. For every path within the tree, if the last $i-th$ syllable of the keyword is found among the N candidates of the $i-th$ level branch, the acoustic scores, including base syllable score and tone score, of the syllable will be accumulated to the keyword. If the syllable is not one of the N syllables candidates, a relatively lower score, which is dynamically decided according to the N candidates' scores, will also be accumulated to the keyword to keep the path for further observation. While reaching a leaf node, the accumulated score is to be adjusted, normalized by the duration and compared with those already stored in a stack. If this normalized score is high enough, the keyword with this score will be stored in the stack too. This keyword-matching process will be repeated for every keyword starting at every possible ending frame. After all ending frames and all keywords have been searched through, the keywords with the highest scores will be picked out.

2.4 Score measurement

After separately describing each phase of the three-phase framework, here we will further formulate the problem in this sub-section. Taking the logarithm of the conditional probability as the measurement for the likelihood, we can formulate our problem as below. The log-likelihood score $L_k(b_1,b_2)$ of the observation sequence $O(b_1,b_2)=\{o_t, b_1 \le t \le b_2\}$ located between the interval (b_1,b_2), given keyword W_k, can be denoted as :

$$L_k(b_1,b_2)=\sum_{i=1}^{I_k} l_i(O(b_1,b_2)|s_i) \tag{1}$$

where

$$l_i(O(b_1,b_2)|s_i)=\sum_{t=t_{i-1}+1}^{t_i}\log p(o_t|s_i), \quad t_0=b_1-1, \quad t_{I_k}=b_2 \tag{2}$$

I_k represents the syllable number of the $k-th$ keyword, and s_i is the $i-th$ syllable, t_i denotes the ending frame of s_i in a speech segment. In order to fairly compare candidates with different lengths of duration, we define the normalized likelihood score as:

$$\Phi_k(b_1,b_2)=L_k(b_1,b_2)/(b_2-b_1+1) \tag{3}$$

Now, the problem can be defined as:

$$(W_k^*,b_1^*,b_2^*)=\arg\max_{(W_k,b_1,b_2)}\Phi_k(b_1,b_2) \tag{4}$$

That is, given an input speech, for all keywords at all possible segments of speech, we will try to find their normalized likelihood scores. The keyword W_k with the highest score will be picked out to be the final result, and its location (b_1^*,b_2^*) will also be identified.

3. Speed improvement

From the description of the three-phase framework in last section, we can see that the computation time needed for the second phase is approximately proportional to the total number of possible boundaries B in the input utterance. Because the more syllable boundaries are found, there will be more segments that may include a syllable each, and more computation will be needed for the syllable and tone recognition. Another observation is that the searching space for the third phase is approximately proportional to $B\cdot\sum_{k=1}^{K}n^{I_k}$, where n is the average number of possible ending frames for the preceding syllable corresponding to a given ending frame, and K is the total number of keywords. Note that for those keywords with large I_k, we have to search

through a very large tree expanding from every possible boundary frame. Several techniques are therefore developed to improve the efficiency of our framework, and they are thus summarized below.

First, we should try to prune the unlikely paths expanding in the tree as early as possible. To reduce the searching space, three criteria are used to decide whether to prune a path or not. One is the score ratio and the others are appearing ratios of syllable and sub-syllable units of the keyword in a path. For example, when the accumulated score obtained in the middle of a path is too small compared to that of the Top 1 keyword which is already in the stack, or when there are already too many syllables or sub-syllable units in the keyword which are not within the top N candidates of the corresponding segments along the path, we will prune the path as early as possible. By properly adjusting the thresholds for these pruning strategy, the searching space can be reduced accordingly.

Secondly, another more efficient technique is using additional information to reduce the number of possible syllable boundaries B. This not only helps to speed up both the second and the third phases, but can also further improve the accuracy. Instead of obtaining possible syllable boundaries simply by energy dips, here a multi-pass method is used to obtain the syllable boundaries through better use of energy contour, zero crossing rate, pause duration, and phrase-final lengthening. All boundaries will be classified into three types of boundaries, including soft syllable boundaries, hard syllable boundaries, and phrase boundaries. Boundaries with higher probability, e.g. some phrase boundaries and hard syllable boundaries, will be picked out first and they will divide the utterance to some shorter segments. The soft syllable boundaries in those shorter segments are then picked out. When the acoustic recognition process is performed, a syllable is not allowed to cross any hard syllable boundaries and more phrase boundaries can be estimated by comparing the recognized syllable length to the statistical syllable length. And when the keyword-matching process is performed, it is not allowed for a short keyword to cross any phrase boundary. After better estimating syllable boundaries and some reasonable constrains, the searching space will be reduced significantly.

The third method makes use of the special structure of Chinese keywords to improve the efficiency. Instead of searching through all the keywords one by one, we try to reject whole groups of impossible keywords in the early stage. For example, in the second task, the 2611 keywords are actually the titles of banking/financing organizations in Taipei. It can be found quite often that the last two syllables are some common ending sub-words representing a "bank (In-Hang, 銀行)", a "corporation (Gung-Sr, 公司)", ... etc., so we can in fact classify these 2611 keywords into about 100 groups based on the common ending sub-words. In this way the search algorithm can be modified to have two stages. The first stage simply quickly reject whole groups of keywords by searching for the short common ending sub-words, and reserve only the possible candidates. In the second stage, we only have to search keywords with the same ending sub-words reserved after the first stage, which are often very small part of the total keywords.

4. Likelihood score modification

To further improve the system performance, we develop some techniques to modify the likelihood scores. We have known that the likelihood score over the best path often varies with time, this makes it difficult to compare words at different time. To deal with this problem, we try to find an adequate scoring method to reduce the variation of the likelihood. The concept of background models has been introduced[6], we will apply it to our approach. In equation (2), we only count the conditional probability of keyword models, which are constructed by vocabulary independent sub-syllable models. Here we modify it to be

$$l_i'(O(b_1,b_2)|s_i) = \sum_{t=t_{i-1}+1}^{t_i} [\log p(o_t|s_i) - \log p(o_t|g)], \quad t_0 = b_1 - 1, \quad t_{I_k} = b_2 \tag{5}$$

where g is background model. After modifying equation (2) to equation (5), the equations (1), (3), and (4) are still adequate for problem definition.

Two different types of background models are designed in this paper. The first approach simply uses only one background model which is trained by all of training data. After considering the structure of the acoustic models, i.e. initial/final models, used in this paper, the second approach adopts three background models, one for background noise g_n, and the other two, g_i and g_f, for initials and finals. When the acoustic recognition process is performed for a syllable, the scoring method is shown in Figure 3. The score for the

initial part of a syllable will be normalized by the score for g_n or g_i, whichever is larger, and the score for final part will be normalized by the score for g_n or g_f, whichever is larger. This easy modification process will normalize the likelihood score to an adequate level.

5. Experimental results

5.1 Baseline experiments

Two tasks with completely different characteristics were tested in this research. The first one is a task with vocabulary size of 9 keywords, and each keyword includes 2 syllables. A spotting rate of 92.52% is obtained while the conventional Viterbi decoding algorithm with filler models gives only 88.46%. The second task has 2611 keywords, and each keyword includes 2 to 20 syllables. A 82.57% spotting rate for the top 10 candidates is attained but the speed requires more than 10 times of real-time when the test was performed on a Sparc 20 workstation.

5.2 Experiments on speedup techniques

Table 1 shows the spotting rates (S.R.) for the correct keywords to appear within the top 10 candidates for the second task mentioned previously with 2611 keywords: when the pruning criteria are getting more strict, the searching space is reduced step by step. In this table, 'Node no.' is the average number of nodes to be searched through for an input utterance whose average length is 2.5 seconds. It can be found from the table that the spotting rate is degraded only slightly when the search space is reduced from 7.0×10^7 to 4.5×10^6, but further reducing the search space will cause large performance degradation. We will choose the parameters at the point that search space equals to 4.5×10^6 for further experiments.

As can be found in the middle column of Table 2, for the same task with better syllable boundaries and some reasonable searching constrains, the average searching space is further reduced to 7.5×10^5, but the spotting rate can be significantly improved to 83.33% simultaneously. It is because the better syllable boundaries are estimated, the less confusing phenomena caused by recognition error will occur. Therefore, the spotting rate can be improved even though the searching space is further reduced.

In the last column of Table 2, it can be found that if we taking advantage of common ending sub-words, the same spotting rate can be maintained while the searching space can be further reduced significantly. In the preliminary tests on a Sparc 20 workstation, for this case, it takes in average only 3.5 seconds to spot a keyword out of 2611 candidates from an utterance with average length of 2.5 seconds. It needs about 1.4 times of utterance length to process an utterance.

5.3 Experiments on likelihood score modification

Table 3 shows the results of using background models to modify the likelihood scores. After carefully choosing numbers of state and mixture of background models, the spotting rates for the first task will further rise to 93.07% and 93.73% for the cases of one background model and three background models. For the second task, the spotting rate can be risen to 83.85% when only one background model is used. And 84.52% of spotting rate will be achieved when three background models are used, but the processing speed will be a little reduced to 1.6 times of utterance length.

6. Conclusion

In this paper, we present an local-segment-based approach for spotting large vocabulary Chinese keywords from a spontaneous Mandarin speech utterance. Taking advantage of the mono-syllabic structure of Chinese language and carefully considering the phenomena of spontaneous speech, a three-phase framework is designed to spot keywords directly from the local segments within a speech utterance. A special scoring method and some techniques are also developed to further improve the efficiency and accuracy of the approach. Very attractive performance was demonstrated in the experiments.

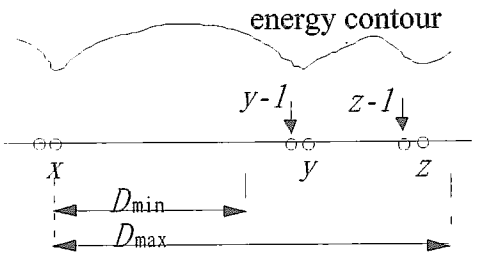

Figure 1. A section of an example utterance. x, y, and z are possible beginning frames, while y-1 and z-1 are ending frames corresponding to x.

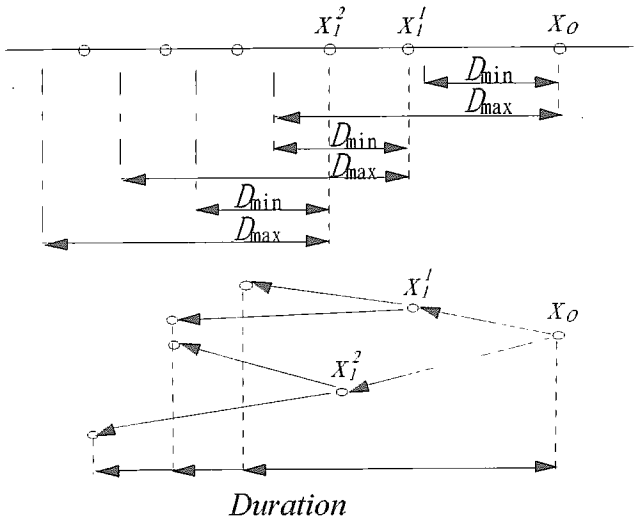

Figure 2. An example of a searching tree for a two-syllable keywords ending with X_0

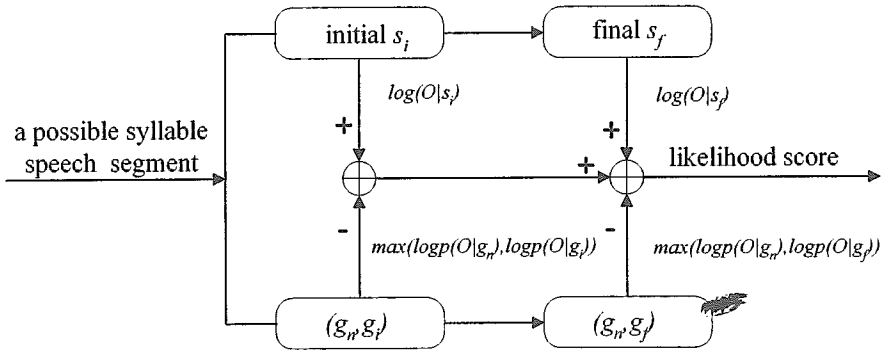

Figure 3. Block diagram for performing the likelihood score modification

Node no.	7.0x10 7	1.8x10 7	4.5x10 6	1.0x10 6
S.R.(%)	82.57	82.20	81.82	75.73

Table 1. Spotting rates under different search spaces by pruning the tree paths.

	path pruning	better boundaries	common sub-words
Node no.	4.5x10 6	7.5x10 5	1.5x10 5
S.R.(%)	81.82	83.33	83.33

Table 2. Spotting rates and search spaces after the three steps of improvements.

	no background model	one background model	three background model
Task 1	92.52	93.07	93.73
Task 2	83.33	83.85	84.52

Table 3. Spotting rates improovement by modifying likelihood score using background models.

References

[1] Lin-Shan Lee, et al, "Golden Mandarin (III) - A User-Adaptive Prosodic-Segment-Based Mandarin Dictation Machine for Chinese Language with Very Large Vocabulary", *Proc. 1995 IEEE Int. Conf. on Acoust., Speech and Signal Processing*, Vol. 1, pp. 57-60, 1995.

[2] Hsin-Min Wang, Jia-Lin Shen, Yen-Ju Yang, and Lin-Shan Lee, "Complete Recognition of Continuous Mandarin Speech for Chinese Language with Very Large Vocabulary but Limited Training Data", *Proc. 1995 IEEE Int. Conf. on Acoust., Speech and Signal Processing*, Vol. 1, pp. 61-64, 1995.

[3] Ron Cole, et al, "The Challenge of Spoken Language Systems: Research Directions for the Nineties", *IEEE Trans. on Speech and Audio Processing*, Vol. 3, No. 1, January 1995.

[4] Eng-Fong Huang, Hsiao-Chuan Wang, and Frank K. Soong, "A Fast Algorithm for Large Vocabulary Keyword Spotting Application", *IEEE Trans. On Speech and Audio Processing*, Vol. SAP-2, No.3, pp. 449-452, July 1994.

[5] Donia R. Scott, "Duration as a Cue to the Perception of a Phrase Boundary", *Journal of the Acoustic Society of American*, Vol. 71, No.4 , pp. 996-1007, April 1982.

[6] Richard C. Rose and Douglas B. Paul, "A Hidden Markov Model Based Keyword Recognition System", *Proc. 1990 IEEE Int. Conf. on Acoust., Speech and Signal Processing*, Vol. 1, pp. 129-132, 1990.