1998 International Computer Symposium
Workshop on Artificial Intelligence
December 17-19, 1998, N.C.K.U., Tainan, Taiwan, R.O.C.

Robust Multi-Keyword Spotting of Telephone Speech Using Stochastic Matching

Chung-Hsien Wu, Yeou-Jiunn Chen, and Yu-Chun Hung

Department of Computer Science and Information Engineering,
National Cheng Kung University, Tainan, Taiwan, R.O.C.
Email:{chwu,chenyj}@csie.ncku.edu.tw

ABSTRACT

In telephone speech recognition, the acoustic mismatch between the training and the test environment often causes severe degradation due to the channel distortion and ambient noise. In this paper, a two-level codebook-based stochastic matching (CBSM) is proposed to deal with the acoustic mismatch. For multi-keyword detection, we define a keyword relation table and a weighting function for reasonable keyword combinations. In the multi-keyword spotting system, 94 right context-dependent INITIAL's, 37 context-independent FINAL's and 1 silence model are adopted. In order to evaluate the multi-keyword spotting system, 1275 faculty names and department names are selected as the keywords. Using a testing set of 2400 conversional speech utterances from 8 speakers, the proposed two-level CBSM can reduce the recognition error rate from 36.52% to 13.4%.

1. INTRODUCTION

In the last decade, many approaches have been developed for the speech recognition over telephone network [1, 2]. When Hidden Markov Model (HMM)-based speech recognizer is applied to telephone network, the performance often makes great degradation. This is because of the acoustic mismatch between training and testing environment. For telephone speech recognition, the variant environment becomes much more serious due to the channel distortion effect, the ambient background noise and the variant speaker. For the keyword spotting, many algorithms have been proposed to spot keywords from continuous speech [3, 4]. In Mandarin speech all syllables are monosyllabic and the unvoiced part of syllable is difficult to be correctly recognized for telephone speech. Of these, most of them use template-based or HMM-based continuous speech recognition algorithms. In these cases, filler templates or HMM's are used to match non-keyword speech and non-speech sounds. Thus, the utterance will be decoded in terms of a sequence of filler and keyword models. Therefore, some syntactical constraints are usually applied to the keyword spotting system such as constraining the system to recognize at most one keyword per utterance. Given this formalism, it is difficult to design the filler models and the performance of keyword spotting will heavily depend on the filler models. It is also difficult to extend single keyword spotting to multi keywords spotting.

Many keyword spotting systems not only use word/subword models but also filler models to decode an input speech utterance into a sequence of keywords and non-keywords. However, it is always very difficult to train filler models for the non-keyword speech, and it is even more difficult to model the lower level events such as non-speech noise. In addition, most reported keyword spotting techniques only consider the small vocabulary problem [3]. Also, for multi-keyword spotting, to train filler models is a difficult work. Therefore, in our multi-keyword spotting recognition system, 94 right context-dependent INITIAL's, 37 context-independent FINAL's in Mandarin speech, and one background/silence model are used as the basic recognition units and the filler model is not included in the basic recognition units.

In this paper, a two-level CBSM is proposed to deal with channel effect. For each mixture components of HMM's, we construct a two-level codebook which was employed to estimate the bias between training and testing environment. Then the bias is used to adopt the acoustic space of speech models to the input corrupted speech. For multi-keyword detection, a fuzzy search algorithm is also proposed to deal with the recognition error problem. For extending single keyword spotting to multi-keyword spotting, a keyword relation table is constructed to record the relationship among keywords. According to the positions of keyword candidates in a sentence and the keyword relation, we can find reasonable keyword combinations. Also, a weighting method is used to decide the final weighted distance of keyword combinations.

2. SYSTEM DESCRIPTION

A block diagram of the multi-keyword system is shown in Fig. 1. The architecture of the keyword spotting includes 4 parts: the two-level CBSM, phonetic recognition, fuzzy research and multi-keyword spotter.

2.1 Two-level CBSM

For deriving the two-level CBSM, we find that the speech segments and the non-speech segments suffer different level of effects in the environment. The differences are as follows:

(1) When the signal-to-noise ratio (SNR) is high, the speech segments are primarily interfered by the channel effect.

(2) When the SNR is low, the distortions of speech segments are both from ambient noise and channel effect.

(3) The ambient noise affects the non-speech segments in

all kinds of SNR.

These imply

(1) The primary distortion resource of speech segments is the channel effect and

(2) The primary distortion resource of non-speech segments is the ambient noise.

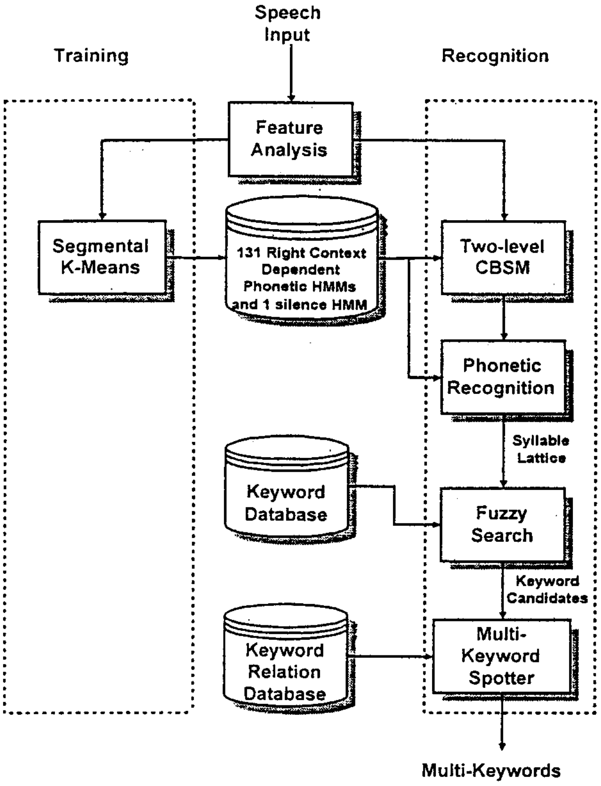


Fig. 1 A block diagram of the multi-keyword spotting

According to the above derivation, we estimate the biases for speech and non-speech separately, and compensate different segments respectively to increase the recognition rate. Additionally, because the energy of the initial part is low and the noise affects the initial part more than the final parts, we separate the speech segment into initial parts and final parts. For different characteristics, we classified the models into different parts firstly. Then K-means clustering algorithm [5] is applied to cluster the mixtures in each part. This method is modified from codebook-based stochastic matching (CBSM)[1]. It is named as **two-level CBSM**. The idea of two-level CBSM is to classify the mixtures according to the characteristics of speech.

The procedure of the two-level CBSM algorithm is shown in Fig. 2, and the algorithm is described as follows:

Construction of two-level codebook

Step 1 : Clustering the models into d classes according to the characteristics of speech and non-speech.

Step 2 : For each class of models, using K-means clustering algorithm to cluster the mixture component, $\{\mu_{n,m}\}$.

The two-level codebook, Ω with k classes can be clustered into d groups expressed as following

$$\Omega = \bigcup_{i=1}^{d} \bigcup_{j=1}^{k_i} \Omega_j^i \,, \qquad (1)$$

where k_d represents the number of classes in group d and $k_1+k_2+...+k_d=k$. A codebook with 7 classes can be obtained by separating models into 3 groups, including the initial part, the final part, and the silence part. Then the initial part is separated into 2 classes, the final part is separated into 4 classes, and the silence is separated into only one class. A diagram of the two-level codebook with 7 classes is shown as Fig.3.

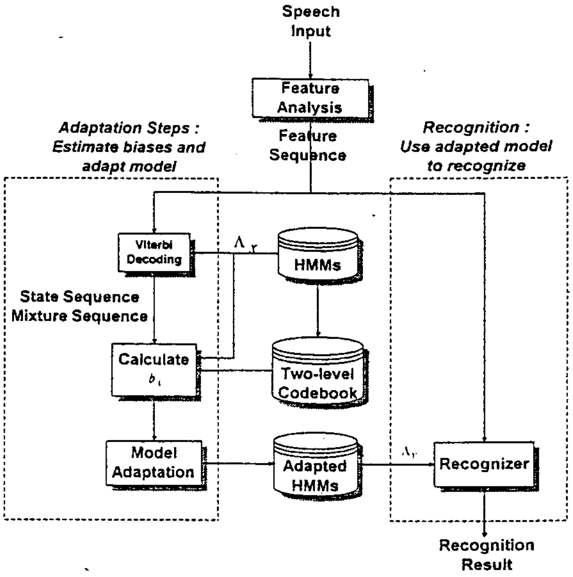


Fig. 2 A block diagram of the procedure of two-level CBSM

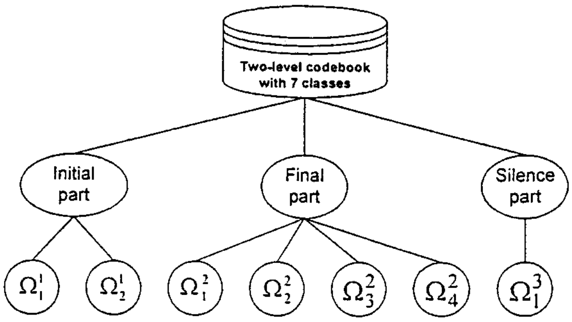


Fig.3 A diagram of a two-level codebook with 7 classes.

Adaptation steps

Step 1 : Viterbi decoding algorithm is used to find the state and mixture sequences with the maximum likelihood of the observation.

Step 2 : The biases, $\left\{b_k^{(i)}\right\}_{i=1,\dots,D;k=1,\dots,K}$, are calculated as

$$b_k^{(i)} = \frac{\sum_{t=1}^{T} \sum_{(n,m)\in\Omega_k} \gamma_t(n,m) \frac{y_{t,i} - \mu_{n,m,i}}{\sigma_{n,m,i}^2}}{\sum_{t=1}^{T} \sum_{(n,m)\in\Omega_k} \frac{\gamma_t(n,m)}{\sigma_{n,m,i}^2}} \qquad (2)$$

where $i=1,\dots,D$ is the dimension of vector y_t. $\gamma_t(n,m)$ is the joint likelihood of observing Y and the m-th mixture component of the n-th state producing the observation y_t, and $\gamma_t(n,m) = p\left(s_t = n, c_t = m \middle| \hat{B}, \Lambda_X, Y\right)$.

Step 3 : Adapt the original models, Λ_X, to the adapted model, Λ_Y, by

$$\hat{\mu}_{n,m} = \mu_{n,m} + \sum_{k=1}^{K} b_k I_{\Omega_k}(n,m), \qquad \forall (n,m) \qquad (3)$$

where $b(t,n,m) = \sum_{k=1}^{K} b_k I_{\Omega_k}(n,m)$. $b(t,n,m)$ is the bias estimate associated with frame t, state n, and mixture m. $I_{\Omega_k}(\bullet)$ is the indicator function for the group Ω_k.

Recognition

Using Λ_Y and Y to generate the string output.

2.2 Recognition

For a given input speech utterance, we firstly find the syllable lattices in the recognition phase. Each Chinese character corresponds to a syllable. Usually we represent a Mandarin syllable as an INITIAL-FINAL model. Some syllables are vowels only, and no consonant appears in the INITIAL part. Since the acoustic characteristic of the INITIAL is affected by its following FINAL, we consider the INITIAL a context-dependent unit. The grammar used in the transition between subsyllable HMM's allows for an INITIAL HMM in Group i followed by a FINAL HMMs in Group i or allows only for a FINAL HMM to form a syllable unit. Each syllable unit can then be followed by any syllable unit. This is shown in Fig. 4.

In the recognition pass, we use the Viterbi decoding algorithm by 94 right context-dependent INITIAL's, 37 context-independent FINAL's and one silence HMM to segment the input utterance into syllable lattice. In the recognition process, Viterbi algorithm is employed to find the most likely subsyllable string $S=s_1s_2\dots s_p$, where

$$S = \arg\max_{S_i} L(O|S_i) \qquad (4)$$

and $L(O|S_i)$ is the likelihood of the observation sequence O given subsyllable string S_i. Based on the most likely subsyllable string S, we combine the INITIAL's and FINAL's to a syllable string, $\tilde{S}$. Then the boundaries of syllables in $\tilde{S}$ are used to find the N-best candidates by Viterbi algorithm and form a syllable lattice.

2.3 Fuzzy Search Method

In the continuous speech recognition process, the Viterbi-Parallel Backtracking algorithm (VPB) algorithm [6] is adopted to generate a syllable lattice. The grammar used in the VPB is called no-grammar where any syllable can follow any syllable. The syllable boundaries associated with the optimal syllable string can be generated with a simple backtracking procedure. Therefore, the work of segmentation can be done by the Viterbi search. According to the syllable boundaries, we can backtrack separately to find the N-best syllables to construct syllable lattice.

With the syllable lattice, a fuzzy search method is used to extract the possible keyword candidates. In the fuzzy search algorithm, each syllable is decomposed into two subsyllables, namely INITIAL part and FINAL part in Mandarin speech. The likelihood between every two subsyllables is calculated in the training process. The nearest neighbor for each subsyllable is determined and used to compensate the substitution, insertion, or deletion errors. Using the fuzzy search algorithm, we can find the most likely keyword $K(i)$, where

$$K(i) = s_1^i s_2^i \cdots s_L^i . \qquad (5)$$

The subsyllable string $s_1^i s_2^i \cdots s_L^i$ is the subsyllable lexical representation of the i-th possible keyword, $K(i)$, and L is the number of subsyllables which are included in $K(i)$.

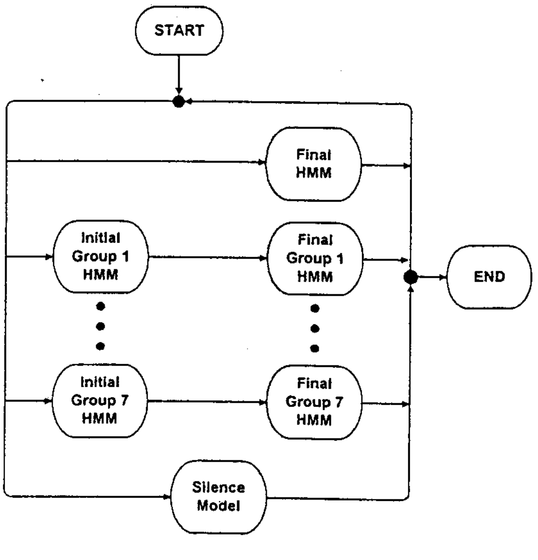


Fig. 4 The grammar used in the transition between subsyllable and silence models

2.4 Multi-Keyword Spotter

After the fuzzy search phase, the keyword candidates from an utterance are obtained. These candidates could be the result of single keyword spotting system, and the result includes only one keyword. For getting more information from the utterance, multi-keyword spotting is a method to achieve this purpose. In this section, we use the information of keyword candidates from fuzzy search to find all the possible combinations among the keyword candidates. In our system, multi-keyword spotting is integrated to process the keyword candidates and output the result of multi-keywords, which is allowed to include one or more keywords in an utterance.

2.4.1 Keyword Relation Table

For extending the single keyword spotter to a multiple keyword spotter, we need to construct the relation among the keywords, namely *keyword relation table*. This table is used to decide the correction when two or more keywords exist in the same utterance. The structure of the *keyword relation table* is denoted as *(PK, SK)*. The meaning of *PK* is *Primary Keyword*, which is defined that the keyword is unique and necessary in user's inquiry. In our system, we define the name of a person who is in National Cheng Kung University (NCKU) as the *PK* in *keyword relation table*. The meaning of *SK* is *Secondary Keyword*, which is defined to be non-unique or unnecessary in user's inquiry. *SK* is expandable when new relation is established. In our system, we define the department in NCKU as the *SK* names in *keyword relation table.*

2.4.2 Multi-Keyword Spotting

In the multi-keyword spotting phase, we use the keyword candidates extracted from the fuzzy search phase to combine the multi-keywords in user's inquiry. According to two conditions, keyword sequence and keyword relation, all of the possible multi-keyword candidates are combined.

In the condition of keyword sequence, we have to find the combination of keywords that keywords are not conflictive in the same keyword sequence. The beginning frame and end frame of all keyword candidates which are extracted from the fuzzy search phase are used to consider the keyword sequence. The keyword combinations that overlap in time are deleted. The result of possible multi-keywords are expressed as

$$\{(K(i),K(j))\}, \text{ where } B_{K(i)} > E_{K(j)} \text{ or } E_{K(i)} < B_{K(j)}. \qquad (6)$$

where $B_{K(i)}$ and $E_{K(i)}$ are the beginning and end frame of *i*-th keyword, *K(i)* respectively. $B_{K(j)}$ and $E_{K(j)}$ are the beginning and end frame of *j*-th keyword, *K(j)*, respectively.

The relations in the *keyword relation table* are also considered to find the possible combinations from the results of the above multi-keywords, $\{(K(i),K(j))\}$. After finding out all possible combinations from keyword candidates, every possible path is given a weighted distance, *WD*. The weighting method that uses secondary keyword distance to calculate the weight of the primary keyword distance is expressed by the equation,

$$WD_{path(p)} = W_p Dist_{PK(P)} \qquad (7)$$

where $WD_{path(p)}$ is the weighted distance given to the path *p* and $Dist_{PK(p)}$ denotes the PK's distance. W_p is calculated by a sigmoid function :

$$W_p = \begin{cases} 1, & \text{if no SK in path p} \\ \left(\prod_{i=1}^{N} \frac{k}{1+e^{-\lambda \times (Dist_{SK,(p)} - DN)}} \right)^{1/N} + (1-k), & \text{otherwise} \end{cases} \qquad (8)$$

where *N* is the number of SK in path *p*. And *DN* is defined as follows

$$DN = \alpha \times Dist_{Min} + (1-\alpha) \times Dist_{Max} \qquad (9)$$

where $Dist_{Min}$ and $Dist_{Max}$ are minimum and maximum word distance among primary keywords respectively in possible paths. The steep part of the slope is between $Dist_{Min}$ and $Dist_{Max}$ in order to separate the different SK weights. The slope is affected by the parameter λ, and the value of W_p is between *(1-k)* and *1*.

3. Experimental Results

In the multi-keyword spotting system, a 26-dimension feature vector was extracted. 12 Mel-Frequency Cepstrum Coefficient (MFCC), 12 delta MFCC, delta log energy, and delta delta log energy are adopted. 94 INITIAL and 37 FINAL HMM's are used to construct the phonetic recognizer. Each INITIAL HMM consists of 3 states and each FINAL HMM consists of 5 states, each with 10 Gaussian mixture densities. In general, for every subsyllable model in the model set, a corresponding anti-subsyllable model for every subsyllable is trained specifically for the verification task. A continuous telephone speech database which is a part of Mandarin Speech Database Across Taiwan (MAT) with 295 speakers (192 males and 103 females) is employed to train the HMM's of this system. All speech data were recorded via public telephone lines in 8KHz using a Dialogic D/41D telephone card and a 16-bit Soundblaster card.

In the multi-keyword spotting system, 1275 faculty names and department names in National Cheng Kung University, Taiwan were selected as the keywords. A continuous telephone-speech database was employed to train the system. The database is part of the MAT (Mandarin Speech Database Across Taiwan) speech database and is composed of short spontaneous speech, number, syllables, words, and sentences. The total number of files is 12368. This database was pronounced by 295 speakers (192 males, 103 females). All speech data were recorded via public telephone lines in 8KHz using a Dialogic D/41D telephone card and a 16-bit Soundblaster card.

In order to evaluate the performance of the multi-keyword spotting system, 1275 faculty names and department names in National Cheng Kung University, Taiwan were selected as the keywords. We also recorded 2400 utterances with five categories which are listed below collected from 8 different people.

- In-vocabulary names, spoken in isolation primary keyword (PK) : 8%
- In-vocabulary names, spoken in primary keyword which is embedded in a sentence (PK+N) : 24%

- In-vocabulary names, spoken in primary keyword and secondary keyword (PK+SK) : 16%
- In-vocabulary names, spoken in primary keyword and secondary keyword which are embedded in a sentence (PK+SK+N) : 44%
- Out-vocabulary names, spoken without any keywords (N) : 8%

3.1 Experiment of Single Keyword Spotting and Multi-Keyword Spotting

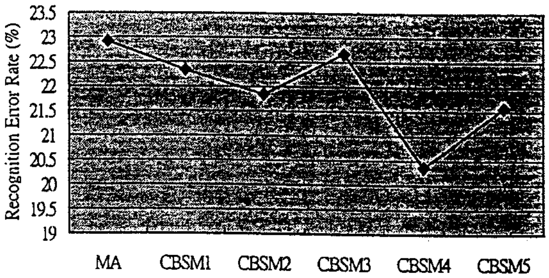
In order to deal with single keyword spotting and multi keyword spotting, the cepstral mean subtraction (CMS) technique is used as the front-end processing. For the multi-keyword spotting, the speech utterances divided into four categories were experimented upon to evaluate the effects of the locations of the primary keyword and secondary keyword in an utterance. The experimental results for four categories are listed in Table I. From table I, we can see that the fuzzy search algorithm can reduce the error rates from 36.52 to 33.45. In the single keyword spotting, the recognition error rate for the first category (PK) and the second category (PK+N) was 16.2% and 39.8%, respectively. In the multi keyword spotting, a weighting function is used to extend single keyword to multi keyword and the performance of single keyword was not degraded. However, in the third and fourth categories, the recognition error rates were reduced from 34.9% and 32.6% to 24.6% and 26.7%. This is because the secondary keyword is a helpful information for the keyword spotting. Consequently, we can obtain better performance in these two categories.

Table I Speech recognition error rate (%) for four categories

	Speech utterance category				
	PK	PK+N	PK+ SK	PK+ SK+N	Averag e
Single Keyword without Fuzzy Search	18.6	42.5	37.4	36.2	36.52
Single Keyword with Fuzzy Search	16.2	39.8	34.9	32.6	33.45
Multi Keyword Spotting	16.2	39.8	24.6	26.7	28.84

3.2 Experiment of Codebook Size of Two-Level CBSM

The relationship of codebook size for the two-level CBSM and the recognition error rates are shown in Figure 5. The experiment was conducted with class number equal to 0, 1, 2, 3, 5, and 7. When the class number is 5 (2 classes for initial, 2 classes for final, and 1 class for silence), the recognition error rate is the lowest. From the results, we can see that the recognition error rate is not proportional to the codebook size. This is because that large codebook size results in small adaptation data for each class, and therefore degrades the performance.



MA: One bias for the utterance.

CBSM1: Two-level CBSM with 2 classes of biases, one for speech and one for silence.

CBSM2: Two-level CBSM with 3 classes of biases, one for final, one for initial, and one for silence.

CBSM3: Two-level CBSM with 4 classes of biases, 2 for final, 1 for initial, and 1 for silence.

CBSM4: Two-level CBSM with 5 classes of biases, 2 for final, 2 for initial, and 1 for silence.

CBSM5: Two-level CBSM with 7 classes of biases, 4 for final, 2 for initial, and 1 for silence.

Fig. 5 The performance of Codebook size of Two-level CBSM

3.3 Experiment of Channel Effect Cancellation Methods

Using different channel effect cancellation methods, the experiment results are shown in Table III. In the CBSM, the experiment that was conducted with 1, 2, 4, 8, 16 and 32 bias indicate that a CBSM codebook size of 8 and beyond produces the lowest error rate. Thus we construct a CBSM codebook with size 8 to compare with other channel effect cancellation methods.

In Table II, we can see that the two-level CBSM with 2 biases achieve the best recognition accuracy, but the recognition time is about two times of the baseline. The ML, CBSM and two-level CBSM need Viterbi decoding processes two times, one is used to estimate the bias and another is used to recognize again by new models. In our experiments, the computation load of Viterbi decoding needs 97% CPU time to recognize an utterance and the post-processing such as fuzzy search and multi-keyword spotting is about 3%. The most time of recognition is used in Viterbi decoding and the time consuming is a major disadvantage by using MA and two-level CBSM. The major difference between CBSM and two-level CBSM is the architecture of the codebook. The codebook of two-level considers the different effect of channel noise, and the classes of biases can be estimated closer the distortion. Thus it can achieve better performance when it computes multiple biases for each utterance.

3.4 Experiment of Utterance Verification

For verification, keywords and anti-keywords were modeled using continuous density context-dependent subword HMM's. Table III presents the experiment results of utterance verification for robust multi-keyword recognition. It is clear that the proposed method outperforms the baseline system. For instance, at 4.5% false rejection, the proposed system resulted in 8.9% false alarm rate. Furthermore, for the fifth category (N), the proposed method was able to correctly reject 90.8% of nonkeywords.

Table II. Comparison of channel-effect cancellation methods

	Baseline	CMS	SBR	ML	CBSM	Two-level CBSM
Recognition Error Rate (%)	28.84	23.64	19.89	20.17	17.28	15.36
CPU times	1.68	1.70	1.79	3.23	3.35	3.35

Table III. False alarm rates (%) for five speech utterance categories, at a false rejection rate of 4.5%

	Speech utterance category				
	PK	PK+N	PK+SK	PK+SK+N	N
False alarms (%)	9.4	9.2	9.7	8.3	9.2
Average (%)	8.9				9.2

4. CONCLUSIONS

In this paper, we propose a new method for robust multi-keyword spotting. In this system, 94 right context-dependent INITIAL's and 37 context-independent FINAL's are used as the basic recognition units. We apply the two-level CBSM to remove the effect of mismatch between testing utterance and models in telephone environment. A fuzzy search algorithm is proposed to extract keywords from syllable lattice. According to the keyword relation table, we utilize a weighting function for combining keywords. Experimental results show that the two-level CBSM outperforms the baseline system.

5. REFERENCES

[1] Craig Lawrence and Mazin Rahim, "Integrated bias removal techniques for robust speech recognition", in *Proceedings of Eurospeech-97*, pp. 2567-2570.

[2] Mazin G. Rahim, and Biing-Hwang Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition", *IEEE Transactions on Speech and Audio Processing*, Vol. 4, No. 1, January 1996, pp. 19-30.

[3] Bo-Ren Bail, Chiu-Yu Tseng and Lin-Shan Lee, "A multi-phase approach for fast spotting of large vocabulary Chinese keywords from Mandarin speech using prosodic information" , in *Proc. IEEE Int. Conf. Acoust., Speech, Signal Processing*, 1997, pp. 903-906.

[4] Chun-Jen Lee, Eng-Fong Huang, and jung-Kuei Chen, "A Multi-Keyword Spotting for the Application of the TL Phone Directory Assistant Service," *Proceeding of 1997 Workshop on Distributes System Technoloties & Applications*, pp. 197-202.

[5] L. Rabiner and B-H Juang, "Fundamental of Speech Recognition," Prentice Hall, 1993.

[6] E. F. Huang and H. C. Wang, "An efficient algorithm for syllable hypothesization in continuous Mandarin speech recognition," IEEE Trans. On Speech and Audio Processing, July 1994.