AN IMPROVED MANDARIN KEYWORD SPOTTING SYSTEM USING MCE TRAINING AND CONTEXT-ENHANCED VERIFICATION*

JiaEn Liang, Meng Meng, XiaoRui Wang, Peng Ding and Bo Xu

Institute of Automation, Chinese Academy of Sciences, Beijing, 100080
E-Mail: {jeliang, mmeng, xrwang, pding, xubo}@hitic.ia.ac.cn

ABSTRACT

The task of keyword spotting is to detect a set of keywords in the input continuous speech. The main goal of this work is to develop an improved mandarin keyword spotting (KWS) system for conversational telephone speech (CTS). In this paper, we propose an efficient online-garbage model based KWS system, which integrated with a word-level minimum classification error (MCE) training method and a novel context-enhanced verification method. Experiment showed that the proposed methods can reduce the Equal-Error-Rate (EER) of the system by 13.8% in relative.

1. INTRODUCTION

The task of keyword spotting is to detect a set of keywords in the input continuous speech. A typical keyword spotting system is shown in Fig. 1, where both *"Keywords"* and *"Fillers"* are often modeled by hidden Markov models (HMM) [1]. Then a Viterbi search algorithm is used to spot keywords from the input continuous speech. Suppose some keyword *W* is detected in some time interval $[T_s, T_e]$, which is called a *putative keyword*, a confidence score will be calculated to verify it [2]. The *"Fillers"* here serves as two functions: first as garbage models to filter out non-keyword speech intervals; second as background models (or anti-models) to calculate the confidence measures for putative keywords.

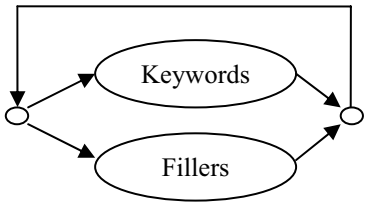

Fig. 1. Typical Keyword Spotting System

For each confidence threshold *C*, the system will report some putative keywords with confidence score greater than *C*. Among these putative keywords, some of them are incorrect, which are called *false alarms (FA)*; and still some keywords are not reported due to low confidence or beam pruning, which are called *false rejects (FR)*. The false alarm ratio and false reject ratio are usually defined as:

$$fa = (FA / KW / HR / M) * 100\% \quad (1)$$

$$fr = (FR / N) * 100\% \quad (2)$$

Where *KW* is the size of keyword set to be detected; *HR* is the duration (in hour) of speech to be processed; *N* is the total number of keywords exist in the test speech; *M* is the expected maximum number of average false alarms. It can be seen that *fa* may be greater than 100%, which means that when the system process one hour speech, the false alarm of each keyword will be greater than *M* in average. *M=10* is used in this paper.

For a given acoustic HMM model, the performance of KWS system mainly depends on the method used for confidence calculations. Significant work has been done to obtain better confidence measures for KWS systems. In [3], the online dynamically constructed filler model was used for confidence score computing. In [4, 5], a discriminatively trained, vocabulary independent utterance verification using anti-subword models was proposed. In [6], techniques were proposed to discriminatively training the weighting factors in the word level confidence scores for word rejection in large vocabulary continuous speech recognition system (LVCSR). Other works such as [7], the rejection threshold was adaptively tuned to improve the rejection performance.

In this paper, we addressed the issue of improving verification performance by two methods: a) inspired by the idea that confidence scores for some phones being more reliable than others of the authors of [6], we introduced the techniques into KWS framework, and proposed a practical implementation scheme as well; b) to alleviate the phenomenon that shorter keywords are less reliable to spot out than the longer ones, we proposed a novel technique to enhance the acoustical characteristics of the shorter key-words by statistically incorporating the context words. When combining the two methods, a significant improvement can be obtained in our Mandarin KWS system.

The paper is organized as follows: Section 2 introduces an online-garbage model based KWS system as a baseline;

* The work is supported by National High Technology Research & Development program 863 of China under contract 2005AA114070.

Section 3 presents an MCE optimized word-level confidence, and Section 4 introduce a novel context-enhanced keyword verification method; Section 5 is some experiments and results; Section 6 will conclude this paper.

2. BASELINE SYSTEM

The baseline system in this paper is an online-garbage model based KWS system. The framework of this system is shown in Fig.1, in which the *"Keywords"* network consists of 100 keywords to be detected and the "Fillers" network consists of 1460 toned mandarin syllables. The *"keywords"* network and the *"Fillers"* network share the same tri-phone HMM acoustic models, and there is no explicit garbage model for the *"Fillers"* network.

The confidence score of the putative keyword $W\,[T_s, T_e]$ in the baseline system is calculated as:

$$CM(W) = \frac{1}{N_w}\sum_{i=1}^{N_w} CM(ph_i) \tag{3}$$

Where N_w is the number of tri-phones in keyword W, and $CM(ph_i)$ is the phone-level confidence for the *i-th* tri-phone ph_i of keyword W. It can be seen that the word-level confidence of keyword W is defined as the arithmetic mean of phone-level confidence. $CM(ph_i)$ is defined as

$$CM(ph_i) = \frac{1}{t_{ei} - t_{si} + 1}\sum_{t=t_{si}}^{t_{ei}} \log\frac{p(X_t \mid S_t)}{p(X_t)} \tag{4}$$

Where X_t is the feature vector at time t, t_{si} and t_{ei} are start and end time of the *i-th* tri-phone ph_i according to the Viterbi alignment, S_t is the aligned HMM state at time t. Note that $CM(ph_i)$ is the time mean of the frame-level logarithm posterior probabilities.

The computation method of $P(X_t)$ has remarkable influence on the performance of KWS systems. In paper [6], it is calculated by a so-called catch-all model, which is used to calculate

$$P(X_t) = \sum_{for\ \text{all}\ Q_t} P(X_t \mid Q_t) \tag{5}$$

Where Q_t is all possible states. In our implement, we modified Eq. (5) as

$$P(X_t) = \sum_{for\ \text{all active}\ Q_t} P(X_t \mid Q_t) \tag{6}$$

It seems that Eq. (5) should be better than Eq. (6) in theory, but experiment showed that it is not true. In fact, when we use Eq. (6) instead of Eq. (5), we get a better performance. We think it is because that Eq. (5) may include some "odd" states, which only give a significant observation probability in some time frames, but cannot survive the sequential beam pruning. So that such a state is not a really competing state and should not be included in $P(X_t)$.

The feature used in this paper was a 42-dimension vector, including 12 MFCC, 1 logarithm energy, 1 pitch and their 1st and 2nd derivatives. The analysis frame length and frame shift are 25ms and 10ms respectively. The acoustic HMM model used in this paper was trained by 300 hours of telephone speech corpus. All performance listed in this paper are tested by a one-hour conversational telephone speech test set with 100 keywords. The results of the baseline system are shown in Fig. 2, where *Tcm* means the scaled confidence threshold at EER point of that curve.

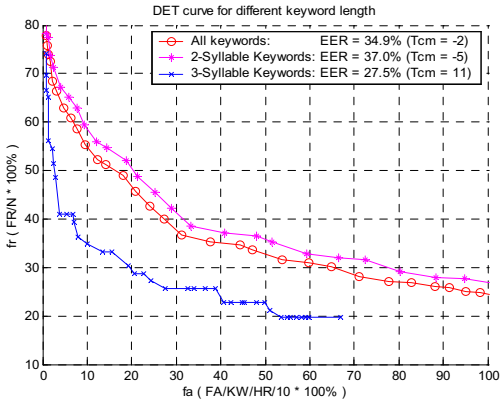

Fig. 2. DET Curves for Different Keyword Length

From Fig. 2 we can observe two facts: the first is that different keyword-length have different confidence threshold at EER point; the second is that long keywords have much better performance than short keywords. The following new approaches are based on these two facts. In the following sections, two approaches will be proposed to combat the abovementioned problems.

3. MCE OPITMIZED WORD-LEVEL CONFIDENCE

MCE optimized word-level confidence score is based on the fact that confidence scores for some phones being more reliable than others [6]. In this section, we proposed a revised scheme to implement the technique under KWS scenario.

The word-level MCE training method has been proposed in [6] for LVCSR systems. The authors suggested that different tri-phones should have deferent weights and biases in calculating the word-level confidence. Compared with Eq. (3), the confidence score should be modified as:

$$CM(W) = \frac{1}{N_W}\sum_{i=1}^{N_W}(a_{ph_i} CM(ph_i) + b_{ph_i}) \tag{7}$$

Where a_{phi} and b_{phi} are the weight and bias for tri-phone ph_i. In KWS systems, Eq. (7) can be treated as an implicit normalization method to solve the different threshold problem as studied in [7], since a_{phi} and b_{phi} can be used to normalize each $CM(ph_i)$ of word W.

In order to train the parameters in MCE framework, a misclassification measure function should be defined:

$$d(W) = (CM(W) - C) \times Sign(W) \tag{8}$$

Where C is confidence threshold to be trained, $Sign(W)$ is a supervised annotation function, which is defined as:

$$Sign(W) = \begin{cases} 1 & \text{if } W \text{ is incorrect} \\ -1 & \text{if } W \text{ is correct} \end{cases} \tag{9}$$

It can be seen that $d(W) > 0$ means an annotation error, and $d(W) \le 0$ means a correct annotation. The goal of MCE training procedure is to minimize the misclassification measure function $d(W)$, that is to minimize the loss function:

$$l(W) = \frac{1}{1+\exp(-\gamma \cdot d(W))} \quad (10)$$

Where γ is a positive real number, which is used to control the smoothness of the sigmoid loss function $l(W)$.

To implement the algorithm under KWS framework, some revise will be made compared with [6]: in LVCSR systems [6], each recognized word can be used for MCE training; while in KWS system, there are always no enough instances for a fixed keyword list. To solve this problem, we changed the keyword list frequently according to the training scripts to ensure enough putative hits. So that the spotting process during MCE training can simulate the process in run time and get enough putative hits for training.

To minimize the loss function $l(W)$, We first derive $l(W)$ over a_{phi}, b_{phi} and C:

$$\frac{\partial l(W)}{\partial a_{ph_i}} = K(W) CM(ph_i) \quad (11)$$

$$\frac{\partial l(W)}{\partial b_{ph_i}} = K(W) \quad (12)$$

$$\frac{\partial l(W)}{\partial C} = -K(W) \quad (13)$$

Where $K(W)$ is defined as:

$$K(W) = \frac{\gamma}{N_W} l(W) \cdot (1 - l(W)) \cdot Sign(W) \quad (14)$$

It is evident that a_{phi} should be a positive number, so we can define:

$$a_{ph_i} = \exp(\tilde{a}_{ph_i})$$

Then Eq. (11) can be modified as

$$\frac{\partial l(W)}{\partial \tilde{a}_{ph_i}} = K(W) CM(ph_i) \exp(\tilde{a}_{ph_i}) \quad (15)$$

With the generalized probabilistic descent (GPD) algorithm, we can easily get the update equation as:

$$\tilde{a}_{ph_i}(n+1) = \tilde{a}_{ph_i}(n) - \varepsilon_n K(W) CM(ph_i) \exp(\tilde{a}_{ph_i}(n)) \quad (16)$$

$$b_{ph_i}(n+1) = b_{ph_i}(n) - \varepsilon_n K(W) \quad (17)$$

$$C(n+1) = C(n) + \varepsilon_n K(W) \quad (18)$$

Where ε_n is the iteration step for *n-th* training sample, which should satisfy:

$$\varepsilon_n > 0, \quad \sum_{n=1}^{+\infty} \varepsilon_n = \infty, \qquad \sum_{n=1}^{+\infty} \varepsilon_n^2 < \infty \quad (19)$$

With Eq. (16) ~ (18), we can iteratively find parameters for Eq. (7), which is a discriminative word-level confidence measure. The main steps of word-level MCE training are listed as follows:

1) Dynamically select a fixed number of high-frequency words according to current training scripts in training process, which is used as keywords for step 2 and 3. In this paper, 100 keywords were dynamically selected for every 100 training sentences.
2) Use Viterbi alignment to segment all training corpus and record the keywords as reference.
3) Run KWS system with the keywords selected in step 1, record all putative keywords and give a supervised annotation according to Eq. (9) and reference given in step 2.
4) Use GPD algorithm to iteratively update a_{phi}, b_{phi} and C according to Eq (16) ~ (18) until converged.

4. CONTEXT-ENHANCED VERIFICATION

A novel context-enhanced verification approach, which can statistically "extend" the keyword length, is proposed in this section.

From the results in Fig. 2, we found that long keywords have much better performance than shorter ones. So we introduce a novel context-enhanced verification method into the baseline KWS system. The main idea of this method is to statistically extend the short keyword to the longer one by using N-Gram language model.

The context-enhanced verification is implemented by adding confidence score to keywords which are detected in a proper N-Gram context. By this method, the false alarms of short keywords can be suppressed to some extent. The main steps of context-enhanced verification are:

1) Training an N-Gram language model with CTS corpus like LVCSR systems.
2) For any given keywords $V_1, \ldots, V_m$, check relevant words $U_1, \ldots, U_k$ and $W_1, \ldots, W_n$ from the trained N-Gram language model, and build an statistically connected keyword phrases network as below:

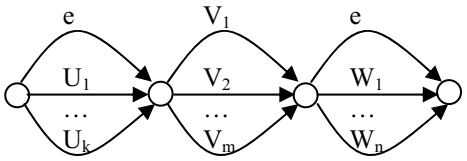

Fig. 3. Keyword phrase network

in the above figure, *e* represent empty word. When some keywords are not found in N-Gram vocabulary, empty word will be used as their grammar context.

3) Replace the *"Keywords"* network in Fig. 1 with keyword phrase network in Fig. 3 and run the KWS system, then the result of keyword-phrase spotting is *(U, V, W)*, where *V* is keyword, and *U, W* are the contexts of *V*, which may be empty words.
4) Let *C(U), C(V), C(W)* be the confidence score of word *U, V,* and *W* define by Eq. (3) or Eq. (7). Then the confidence score of keyword *V* can be calculated as

$$\tilde{C}(V) = \log(P(V) + aP(U)P(V|U) + bP(W)P(W|V)) \quad (20)$$

Where *a, b* is positive real number, used to control the enhancement of confidence. In our experiment, *a=b=1* are used. *P(U), P(V), P(W)* are probabilities of *U, V, W*, which are calculated by $P(X) = exp(C(X))$, note that if *X* is equal to *e*, *P(X)* is defined as 0.

5. EXPERIMENTS AND RESULTS

A one hour mandarin conversational telephone speech test set is used in this experiment, and 100 words are selected as keywords, including 75 short words (2-syllable words) and 25 long words (3-syllable words). Among these 100 keywords, there are 12 keywords which are not in the vocabulary of N-Gram, and there are 398 keyword instances exist in the test set.

The word-level MCE training set is a 90 hours training corpus. Total 2,810,314 putative keywords are detected in the training step 3 of section 3, including 2,651,061 false alarms and 159,253 correct hits. Fig. 4 shows the loss and EER curves of MCE iterations:

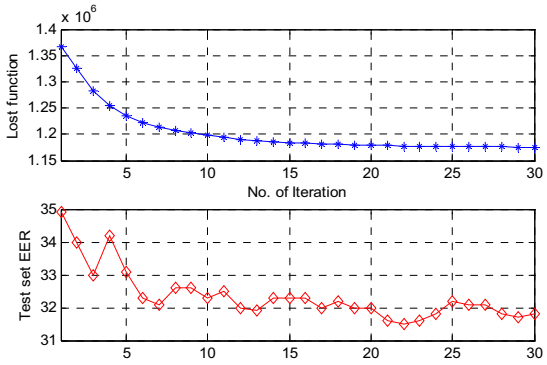

Fig. 4. Loss and EER Curves of MCE Iterations

It can be seen from Fig. 4 that after 5 iterations, both loss function and test set EER are converged slowly, but the loss function curve is much smoother than the EER curve.

Fig. 5 shows the DET curves of the baseline system, as well as the two approaches and their combination.

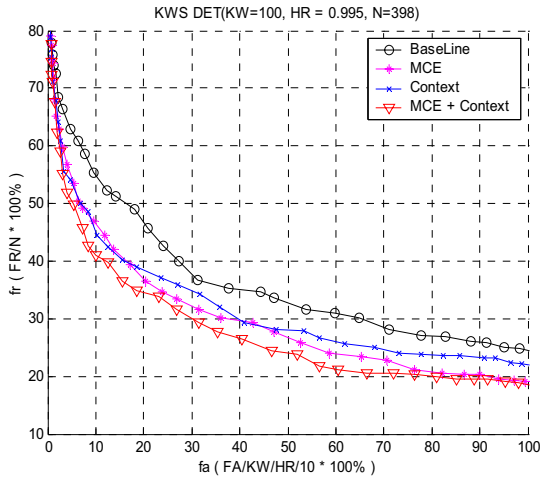

Fig. 5. DET Comparison of the Three Methods

From Fig. 5 we can see that both MCE training and context-enhanced method can improve the performance of the system significantly. Combining the two methods above, the performance can be further improved.

To compare the performance of the three methods in details, we list the EER of the three methods in Table 1, including the corresponding EER for short keywords and long keywords.

Table 1. EER Comparison for the Three Methods

Method	Short Words	Long Words	All Words	Relative improvement
Baseline	37.0%	27.5%	34.9%	-
MCE	33.4%	22.7%	31.5%	9.7%
Context	35.2%	27.0%	33.0%	5.4%
MCE + Context	31.8%	22.7%	30.1%	13.8%

It can be seen from table 2 that the MCE method can improve the performance of all keywords significantly, but the context-enhanced keyword verification method can only have significant improvement on short keywords.

6. CONCLUSIONS

In this paper, we introduce two efficient approaches to the mandarin keyword spotting system: the MCE optimized word-level confidence score and the context-enhanced verification method. They produce relative improvements of 9.7% and 5.4% respectively on conversational telephone speech test set. By combining the two methods, we achieved a total relative improvement of 13.8% in EER reduction.

7. REFERENCES

[1] R. C. Rose and D. B. Paul, "A hidden Markov model based keyword recognition system", *Proc. ICASSP*, Vol. 2.24, pp. 129-132, 1990.

[2] K. M. Knill and S. J. Young, "Fast Implementation Methods for Viterbi-based Word-spotting", *Proc. of ICASSP*, pp. 522-525, 1996.

[3] H. Bourlard, B. D'hoore, and J.M. Boite. "Optimizing recognition and rejection performance in word spotting systems", *Proc. of ICASSP*, Vol. 1, pp. 373-376, 1994.

[4] R.A. Sukkar and C.H. Lee, "Vocabulary Independent Discriminative Utterance Verification for Non-Keyword Rejection in Subword Based Speech Recognition", *IEEE Trans. on Speech and Audio Proc.*, Vol. 4, No. 6, pp. 420-429, 1996.

[5] R. A. Sukkar, "Subword-based Minimum Verification Error (SB-MVE) Training for Task Independent Utterance Verification", *Proc. ICASSP*, pages 229-232, 1998.

[6] S. Abdou and M.S. Scordilis, "Beam search pruning in speech recognition using a posterior-based confidence measure", *Speech Communication*, Vol.42, pp. 409-428, 2004.

[7] D. Jouvet, K. Bartkova, G. Mercier, "Hypothesis Dependent Threshold Setting for Improved Out-of-Vocabulary Data Rejection", *Proc. ICASSP*, vol. 2, pp. 709-712, 1999.