Sensitive Keyword Spotting for Voice Alarm Systems

Chunlei Zhu, Qing-Jie Kong, Lucidus Zhou
Dongguan Research Institute of CASIA
Cloud Computing Center, Chinese Academy of Sciences
Dongguan, China
zhuchun_lei1988@126.com; {kongqingjie;
Zhoushiyu}@casc.ac.cn

Gang Xiong, Fenghua Zhu
The State Key Laboratory of Management and Control for
Complex Systems
Institute of Automation, Chinese Academy of Sciences
Beijing, China
{gang.xiong; fenghua.zhu}@ia.ac.cn

Abstract — **Keyword spotting is the task of identifying the occurrences of certain desired keywords in an arbitrary speech signal. Keyword spotting has many applications. One of them is emergency voice alarm systems, like command control and smart monitoring ATM etc. This paper presents a sensitive keyword spotting system applied to voice alarm occasions. By sensitive keyword retrieval, emergency warning systems can automatically monitor and follow up the situation, find the crisis occurs, and then give the alarm. Automatically retrieving and tracking for the particular occasion can not only save a lot of manpower and material resources, but is also more efficient. So the study of this subject has great reality and application value.**

Keywords—keyword spotting; automatic tracking; smart monitoring; voice alarm systems

I. Introduction

Keyword spotting (KWS), as a branch of automatic speech recognition technology, aims at testing and confirming a number of given specific keywords in continuous speech [1]. Keyword recognition technology compared with continuous speech recognition technology has a lot of advantages. First of all, keyword identification technology is a kind of isolated word recognition, which falls in between the technologies of continuous speech recognition and voice recognition. It does not require the entire speech flow to be all recognized and is rid of the limitations of isolated word speech recognition. It can save a lot of computing resources, so as to build a relatively simple and stable application system. Secondly, in order to obtain high quality voice, continuous speech recognition requires relatively quieter environment and better channels. However, in noisy environment, it has a significant reduction in the performance. The keyword recognition allows use in noisy environments. Even through channels with poor quality such as telephone lines, the system can automatically determine which the keywords are. It is suffice to say under the current technical level, many applications are not suitable for continuous speech recognition, and are requiring the keyword spotting to be in unconstrained voice conversation or natural speech data streams. The main applications of keyword detection include: voice monitoring, command control, voice dialing, intercom, etc. [2-6]. The particularity of keyword detection for testing technology and the many extensive applications determines the great research value and practical significance of keyword spotting research.

The traditional way of spotting keywords is to train individual models for the speech keywords and then represent non-keywords by “filler” or “garbage” models. These models are based on hidden Markov models. Some systems also add extra models for non-speech segments, such as coughing, laughing and silence. However, the HMM-garbage model based keyword spotting systems have some shortcomings:

(1) In order to achieve reliable and effective recognition rates, an HMM generally requires an enormous database for training. If the keyword spotting systems for small vocabulary keywords are required, training for the garbage models will become an extremely tedious work and will take a mass amount of training time.

(2) Viterbi decoding algorithm is a global optimal algorithm and is not for any keywords or garbage element specifically. Consequently, the score is not normalized with respect to a specific keyword, and the score threshold setting is not keyword-specific either. Therefore, the temporal outliers may affect the final global score, resulting in detection mistakes.

(3) Since the garbage model represents all non-keywords in acoustics feature space, it can model any speech vocabulary, including the keywords themselves. This can lead to spotting errors.

(4) A keyword spotting system based on keywords and garbage models based on HMM has difficulties in detecting the desired keywords correctly in real time because it requires a lot of spotting time to match garbage model.

In this paper we discuss the problem of rapidly recognizing a small set of prescribed vocabulary words spoken in the context of unconstrained speech for application-oriented emergency speech alarm systems. In the general case, the keyword recognition system is presented with continuous input and must decide whether or not any of the pre-defined vocabulary words is present anywhere in the speech. While much research has been performed on the general word spotting task, very little of it can apply in practice. Based on keyword spotting technology, the paper realizes the voice alarm system for practical applications. Our approach relies on sliding windows and Hidden Markov Model by HTK tools [7], and on performing keyword-specific threshold setting. For each keyword, the width of the sliding window is optimized specifically.

Research supported in part by National Natural Science Foundation of China grants 61174172, 61233001, and 71232006.

978-1-4799-0530-0/13/$31.00 ©2013 IEEE

The remainder of this paper is organized as follows: in section II, we present the implementation of keyword spotting system in detail. In section III, we present recognition algorithm based on Viterbi algorithm and sliding window. In section IV, we give some experimental results. Finally, a conclusion is drawn.

II. Implementation of Keyword Spotting Systems

In keyword spotting task, the speech signal is assumed to be composed of a combination of keyword and non–keyword speech. One of the most common approaches for the keyword spotting implementation is to consider individual models for the keywords and to represent other words by "filler" or "garbage" models. A classical spotting scheme is performed using hidden Markov models.

Whereas, sometimes, speech corpus is available for a particular domain, so, word spotting can benefit from the more detailed background model. When non-keywords are not present in the training data, Rohlicek [8] suggested to model non-keywords as segments of the keywords. A lot of other suggestions were given [9]. In this paper, we suggest only the modeling of keywords.

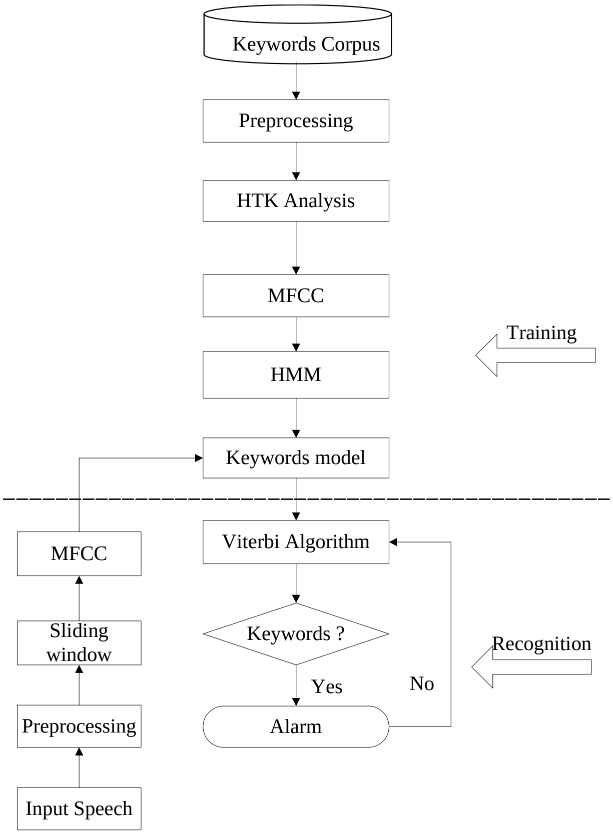

Fig. 1 framework of voice alarm systems.

The system aims to achieve emergency voice alarm systems for some particular applications based on speech keyword spotting technology. Due to the specific application, for instance ATM monitoring, a small set of sensitive vocabulary words will be monitored, and real-time spotting and high accuracy are necessary. In this case, vocabulary pronunciation similarity is not important. Moreover, false detection is preferable to leak detection for safety's sake. Consequently, local prior knowledge and keyword-specific sliding windows are applied to the sensitive vocabulary spotting. The block diagram of our keyword spotting system is shown in Fig. 1.

In the training phase, every keyword template is created using Hidden Markov Model by HTK tools. HMM model state number is set to 6, and Gaussian mixture number is 1.

In the spotting stage, for the testing speech, the whole sequence of speech is segmented as local portions by using sliding windows with a width of a fixed number of frames T. Then, feature vectors of a local portion looking like a keyword are extracted frame-by-frame. For each keyword the width of the sliding window is optimized specifically. Finally, it is decided whether or not the keyword has occurred in the input speech by Viterbi algorithm.

A. Feature Extraction

The first stage in a speech recognizer is the features extraction. The features used in the system are Mel-frequency cepstral coefficients. The set of 26 MFCCs is extracted from the signal using a 20 ms hamming window and a 10 ms shift. It is implemented with HTK tools. Its parameter settings are as follows:

```
#NATURALREADORDER = TRUE
SOURCEKIND = WAVEFORM
SOURCEFORMAT = WAV
TARGETKIND = MFCC_E_D_Z
TARGETRATE = 100000.0
WINDOWSIZE = 200000.0
PREEMCOEF = 0.975
NUMCHANS = 26
CEPLIFTER = 22
NUMCEPS = 12
USEHAMMING = T
DELTAWINDOW = 2
ACCWINDOW= 2
```

Some parameters are interpreted as follows: "TARGETKIND" is the identifier of the coefficients to use; "WINDOWSIZE" represents the length of a time frame (20ms); "TARGETRATE" means frame periodicity(10ms); "NUMCEPS" is the Number of MFCC coeffs; "USEHAMMING" declares whether or not to use Hamming windowing; "PREEMCOEF" is the Pre-emphasis coefficient; "NUMCHANS" represents the Number of filterbank channels; "CEPLIFTER" is the Length of cepstral liftering.

The process of MFCC feature extraction is as follows [10], [11]:

(1) Read normalized speech signal.

(2) Extract the signal section by endpoint detection.

(3) Apply the pre-emphasis of normalized speech signal using formula

$$H(z) = (1 - \mu \cdot z^{-1}) \tag{1}$$

where $\mu \approx 1$.

(4) Apply signal windowing with hamming window to framing.

(5) Apply FFT extraction.

The computation of the FFT is as follow:

$$X[k] = \sum_{n=0}^{N-1} x(n) e^{\frac{-j2\pi nk}{N}}, 0 \le k \le N \quad (2)$$

where N is the frame size in samples, $x(n)$ is the input speech signal, and $X[k]$ is the corresponding FFT spectrum.

(6) Compute Log-energy with formula

$$S[m] = \log[\sum_{k=0}^{N-1} (X(k))^2 \cdot H_m(k)], 0 < m < M \quad (3)$$

where M is the number of filters in the Mel-filter bank, $H_m(k)$ is the triangular M-band filter.

(7) Perform discrete cosine transforms to obtain MFCC.

The formula for discrete cosine transform (DCT) is shown next:

$$C[n] = \sum_{m=0}^{M-1} s(m) \cos(\frac{\pi n(m-0.5)}{M}) \quad (4)$$

where C is the MFCC coefficient desired.

B. *Acoustic Modeling*

Speaker-independent (SI) acoustic models were trained with keyword speech of 20 persons. The acoustic model consists of 6-state HMMs with 1 Gaussian component per state.

III. Recognition Algorithm

Firstly, sliding window were used to cut input test voice into many isolated parts, with each part seen as an isolated word. Then, MFCC features were extracted from each section. At last, the Viterbi algorithm is used to find the most likely sequence of hidden states. We determine whether or not the input speech contains the sensitive keywords by the threshold setting.

Through prior knowledge of corpus, the pronunciation time of each syllable is obtained on average under the background of the application. For each keyword the width of the sliding window is optimized specifically. The shift size of the sliding window used is 235 ms, estimated from the training data in this paper.

Given a sequence of observations $o = o_1 o_2 ... o_T$, and an HMM $H = (p_{i,j}, e_i(a), \omega_i)$, p_{ij} is the state transition probabilities, the emission probability for the observable *a* from state i is $e_i(a)$,and the initial state probabilities are $\omega_i = 1/M$. M is the number of all possible observables.

We wish to find the maximum probability state path $Q = q_1 q_2 ... q_T$. This can be done recursively using the Viterbi algorithm.

Let $v_i(t)$ be the probability of the most probable path ending in state i at time i, i.e.,

$$v_i(t) = \max_{q_1, q_2, ..., q_{t-1}} P(q_1, q_2, ..., q_{t-1}, q_t = i, o_1 o_2 ... o_t \mid H) \quad (5)$$

and let ω_i be the initial probabilities of the states i at time $t = 1$.

Then $v_j(t)$ can be calculated recursively using

$$v_j(t) = \max_{1 \le i \le N} [v_i(t-1) p_{ij}] e_j(o_t) \quad (6)$$

together with initialization

$$v_i(1) = \omega_i e_i(o_1) \qquad 1 \le i \le N \quad (7)$$

where N is the number hidden Markov states.

Finally,

$$P_S = \max_{1 \le i \le N} [v_i(T)] \quad (8)$$

The P_S is desired.

The Viterbi algorithm is a dynamic programming algorithm for finding the most likely sequence of hidden states –called the Viterbi path– that result in a sequence of observed events. For each keyword, the corresponding phonemes will be searched from pronunciation dictionary first. Then Viterbi algorithm is used to compute the probability $P(W/O)$ of phonemes, where O is the observations represented by the extracted features from the incoming signal and W is a given word. At last, keyword similarities are achieved by the logarithmic arithmetic mean of all the underlying phonemes. A global threshold is to decide whether it is a keyword or not, accepting it if it is bigger than the threshold and rejecting it if it is smaller than the threshold. During the searching process, beam strategies are adopted to control the keywords which occur in 1-best phoneme sequences.

IV. Experiments

A. *Experimental Setup*

In this work, we recorded 15 sentences in mandarin and simulated a set of conditions in dangerous environment in studio environment (10 female and 10 male speakers). There are three Chinese keywords: “jiuming’a”, “baojing” and "qiangjie", which in Chinese means "help", "call the police", and "robbery", respectively. Each sentence is uttered 20 times; every keyword is treated as an acoustic example. The ground truth of the time region of each keyword is manually labeled. The test speech is uttered by the same speakers. The total number of the occurrences of the keyword is 1200. All speech data are digitized into 16-bit samples at a sampling rate of 16 kHz. The utterances are pre-emphasized with $\left(1-0.975z^{-1}\right)$, followed by feature analysis using a 20ms Hamming window.

B. Setting Thresholds

Finally the thresholds are set by

$$T = k \cdot m_s \tag{9}$$

where m_s is the averaged score value in the training data. k denotes a constant and is set experimentally. The thresholds are set to control the number of correct keyword detections and false alarms.

C. Results and Discussions

In this section, we present experiment results for keyword spotting based on sliding window and HMM. Table I shows the results of the keyword recognition experiments.

TABLE I. KEYWORD RECOGNITION RESULTS

Keywords	**Recognition Rate**	**Mean Recognition Rate**
'jiuminga'	95%	94%
'qiangjie'	93%	
'baojing'	94%	

From test results, the mean recognition rate of all keywords reached 94%. The recognition speed of the system is relatively high. It is thus clear that the proposed approach has rather good performance.

V. CONCLUSIONS

Keyword spotting is an innovative research area which has many applications, such as emergency voice alarm systems described in this paper. This paper has presented an approach for detection of some particularities with keywords. The system can produce the alarm when a sensitive keyword is detected. This alarm is confirmed by comparing the sequence inside the boundaries to HMM reference models.

In this paper, we have presented our keyword spotting method for spontaneous speech using the keywords predefined by a set of acoustic examples. In addition, we have proposed to train the keyword model using the acoustic examples based on HMM, score local portions using sliding windows and perform threshold setting in the test part. The keyword spotting experiments demonstrate the effectiveness of the proposed method with results of detection performance for sensitive vocabulary recognition systems.

REFERENCES

[1] W. Li and Q. Liao, "keyword-specific normalization based keyword spotting for spontaneous speech," in Proc. 2012 8th Int. Sym. Chinese Spoken Language Processing, 2012, pp. 233-237.

[2] P. Zhang, J. Han, J. Shao, and Y. Yan, "A new keyword spotting approach for spontaneous mandarin speech," in Proc. 2006 8th Int. Conf. Signal Processing, 2006, pp. 16-20.

[3] J.G. Lawrence, R. Rabiner, C.-H. Lee, and E.R. Goldman, "Automatic recognition of keyword in unconstrained speech using hidden markov models," IEEE Trans. Acoustics Speech Signal Processing, vol. 38, no. 11, pp. 1870-1878, November 1990.

[4] J. Nouza and J. Silovsky, "Fast keyword spotting in telephone speech," Radioengineering, vol. 18, no. 4, pp. 665-670, December 2009.

[5] S. Zhang, Z. Shuang, Q. Shi, and Y. Qin. "Improved mandarin keyword spotting using confusion garbage model," in Proc. 2010 20th Int. Conf. Pattern Recognition, 2010, pp, 3700-3703.

[6] M.S. Barakat, C.H. Ritz, and D.A. Stirling, "Keyword spotting based on the analysis of template matching distances," in Proc. 2011 5th Int. Conf. Signal Processing Communication Syst., 2011. pp: 1-6.

[7] S. Young, G. Evermann, M. Gales, T. Hain, D. Kershaw, G. Moore, *et al.*, The HTK Book (for HTK version 3.3). 2005.

[8] J.R. Rohlicek, P. Jeanrenaud, K. Ng, H. Gish, B. Musicus, and M. Siu, "Phonetic training and language modeling for word spotting," in Proc. IEEE Int. Conf. Acoustics, Speech, and Signal Processing, 1993, pp. II 459-II 462.

[9] H. Bahi and N. Benati, "A new keyword spotting approach," in Proc. Int. Conf. Multimedia Computing Syst., 2009, pp. 77-80.

[10] Z. Jiang, H. Huang, S. Yang, S. Lu, and Z. Hao, "Acoustic feature comparison of MFCC and CZT-based cepstrum for speech recognition," in Proc. 2009 5th Int. Conf. Natural Computation, 2009, pp. 56-59.

[11] C.K. On, P.M. Pandiyan, S. Yaacob, and A. Saudi, "Mel-frequency cepstral coefficient analysis in speech recognition," in Proc. Int. Conf. Computing Informatics, 2006, pp. 1-5.