A NOVEL PHONE-STATE MATRIX BASED VOCABULARY-INDENENDENT KEYWORD SPOTTING METHOD FOR SPONTANEOUS SPEECH*

Peng Gao, JiaEn Liang, Peng Ding, Bo Xu

Institute of Automation, Chinese Academy of Science, Beijing, 10080
{pgao, jeliang, pding, xubo}@hitic.ia.ac.cn

ABSTRACT

Keyword spotting (KWS) is an essential technique for speech information retrieval. When doing offline keyword query on large volume spontaneous speech data, fast and accurate KWS methods are required. In this paper, a novel phone-state matrix based vocabulary-independent KWS method is proposed, which has merits of both hidden Markov model (HMM) based and lattice-based methods. Four KWS systems are compared in our experiments on conversational telephone speech test set. Result shows that compared to the high precision HMM-based KWS system the proposed phone-state matrix system has better equal-error-rate (EER) and false-alarm (FA) performance than the other two lattice-based systems.

Index Terms—keyword spotting, spontaneous speech, spoken document search, speech recognition, confidence measure

1. INTRODUCTION

Keyword spotting is to detect target words in continuous speech input. Searching for keywords in speech corpus is an essential technique for spoken document retrieval, which is becoming a more and more important part of information retrieval with ever-increasing audio and multimedia data. When dealing with formal speech data, such as broadcast news, word retrieval can be done naturally by searching large vocabulary continuous speech recognition (LVCSR) results. However, when facing informal or spontaneous speech, teleconference record or conversational telephone speech (CTS) for example, out of vocabulary (OOV) problem greatly degrades KWS performance. Recently, new approaches were explored to solve the KWS problems in spontaneous speech, most of which are based on HMM or lattice search. Both techniques are trying to deal with ambiguity in speech, whereas their methods are quite different. The following paragraphs will discuss these two techniques and their advantages/drawbacks. To overcome these drawbacks, our new KWS method is proposed.

This work is supported by National High Tech R&D program 863 of China under contract 2006AA010103.

HMM-based KWS method is a direct derivative from LVCSR approach, which changes the purpose of Viterbi search process from sentence decoding to keyword detection and verification. A typical HMM-based KWS system is composed of "*Keywords*" and "*Fillers*" which are often modeled by HMMs [1]. If a keyword W is spotted in some time span, it is called a *putative keyword*. The "*Fillers*" serve as garbage models to match non-keyword speech segment and as background models to compute confidence measure for putative keywords. HMM-based KWS system can achieve high precision and low miss rate [2]. However, this method is vocabulary-dependent. If keywords change, HMM decoding must run again on all input speech. This makes it not applicable for the scenario of fast keyword query on hundreds of audio data.

Lattice based KWS approach, on the other hand, concentrates on the reuse of first pass decoding results. It's reasonable to use word lattice as keyword search space [3] for it is more robust than one best result. Many kinds of sub-word units such as syllable or phonemes also can be adopted in lattice generation, which may handle OOV problem [4]. Compared to HMM decoding, lattice search is very fast and vocabulary-independent. The process of lattice search is mainly pattern matching, in which many information retrieval techniques can be used, such as indexing [5][8]. This makes it very attractive for keyword searching in large volume speech corpora. When processing spontaneous speech, however, performance of lattice method is commonly not as good as HMM-based method [3]. Moreover, insertion, deletion and substitution problems must be considered carefully in lattice search process [6].

In this paper, we propose a novel vocabulary-independent KWS method based on phone-state matrix. This method tries to find a trade-off between HMM and lattice approaches. When processing spontaneous CTS speech, the proposed method outperforms lattice methods on EER and FA evaluations. It also has no OOV and insertion/deletion problems which lattice-based methods must deal with.

This paper is organized as follows: In section 2, phone-state matrix based KWS method is fully explored, including matrix generation and search algorithm; Section 3 gives

some experiments on CTS speech, and results are discussed; Conclusion is drawn in Section 4.

2. PHONE-STATE MATRIX BASED KWS

Phone-state matrix KWS method is inspired by the thought of saving intermediate information of HMM decoding for later use. The following figure shows the framework of this method.

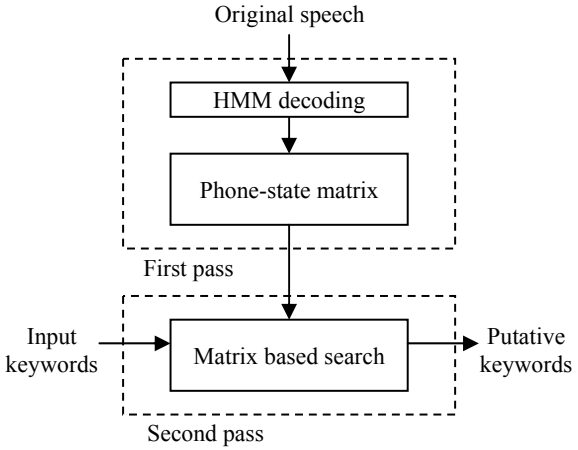

Fig. 1. Phone-state matrix based KWS

Fig. 1 shows that the keyword search process is composed of two passes. The first pass runs HMM decoding with phone-loop filler model on original speech, and outputs a phone-state matrix. In this matrix, one dimension is time and another dimension is phone-state, and each matrix element is a specific phone-state max-observation score. The first pass only runs one time. The second pass is the search pass based on the phone-state matrix generated from the first pass. If input keywords change, only the second pass needs to run again, which is much faster than the first pass. The second pass needs no time-consuming observation score computation while maintaining dynamic phoneme matching capability. No language model (LM) is used in both passes. In the following sections, phone-state matrix generation method is presented and search algorithm based on it is discussed in detail.

2.1. Phone-state Matrix Generation

The phone-state matrix is calculated in the first pass HMM decoding process. Here the max-score of all tri-phones of a phoneme is stored. At time *t*, the likelihood score of j-th state of i-th phoneme $L(Ph_{ij})_t$ is calculated as:

$$L(Ph_{ij})_t = \max_{ph_j \in Ph_i} \{P(X_t \mid ph_j)\} \quad (1)$$

Where X_t is the feature vector at time *t*, ph_j the tri-phone on state j belonging to i-th phoneme, $P(X_t|ph_j)$ is observation score of X_t on ph_j. It is possible to use sum or average function instead of max function in equation (1). Preliminary experiments show that max representation has the best performance. This is probably due to the dynamic programming nature of decoding process.

We also need to store the maximal filler reference observation score at time *t* for later search use:

$$L^t_{\max} = \max_{\forall j} \{P(X_t \mid ph_j)\} \quad (2)$$

Where ph_j is any one of tri-phone on any state j, $P(X_t| ph_j)$ is the observation score of X_t on ph_j.

In order to calculate confidence measure (CM) score in second pass keyword search, we have to do more preparation work. In this paper we use the same CM score calculation method as in [2] [7], and we have to save the $P(X_t)$ at time *t* as:

$$P(X_t) = \sum_{all\ active\ Q_t} P(X_t \mid Q_t) \quad (3)$$

Where Q_t is all possible states, $P(X_t| Q_t)$ is observation score of X_t on Q_t.

Now that we get a phone-state score and additional reference data vector at time *t*, if we calculate and store this vector from time 0 to speech end, a phone-state matrix is generated. This matrix then can be used as following fast keyword search.

If we ignore tri-phone state information in equation (1), we can get a phone matrix:

$$L(Ph_i)_t = \max_{ph \in Ph_i} \{P(X_t \mid ph)\} \quad (4)$$

This may reduce storage consumption at the cost of precision loss.

2.2. Phone-state Matrix Based Search

On the basis of phone-state matrix, KWS task is to find all time segments that match searched keywords and have larger CM score than a preset threshold. Search algorithm is a simulation of HMM-based KWS viterbi search on phoneme HMMs which are already pre-calculated as phone-state matrix. We still use the search model of "*Keywords*" and "*Fillers*" introduced by HMM-based method. The computation of "*Fillers*" part can be ignored in the search. Instead, we use the pre-stored maximal reference score in equation (2) to replace it. Thus, we only need to search on the "*keywords*" part.

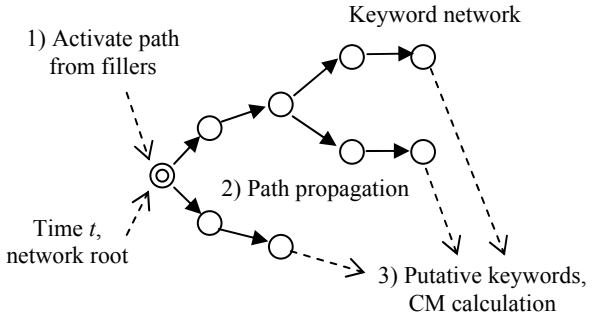

Fig. 2. Search Algorithm

Fig. 2 illustrates the search algorithm. The main steps are described as follows:

1) At time *t*, activate search paths at the beginning of "*keyword*" network, using stored maximal reference score L^{t}_{max} in equation (2) as path initial score;
2) Propagate all search paths on "*keyword*" network, using the pre-calculated phone-state score in equation (1) as observation output score (Here we lose tri-phone sequence information, which leads to precision loss);
3) If a path reaches keyword end, calculate its CM score, output this putative keyword if the CM score is larger than a preset threshold;
4) Repeat step1 to step 3 until time end.

According to [2], the CM score of putative keyword *W* in step 3 is calculated as:

$$CM(W) = \frac{1}{N_w}\sum_{i=1}^{N_w} CM(Ph_i) \qquad (5)$$

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

$$CM(Ph_i) = \frac{1}{t_{ei} - t_{si} + 1}\sum_{t=t_{si}}^{t_{ei}} \log\frac{P(X_t \mid Ph_{ij})}{P(X_t)} \qquad (6)$$

Where X_t is the feature vector at time *t*, t_{si} and t_{ei} are start and end time of the i-th phone Ph_i according to the Viterbi alignment, Ph_{ij} is is the aligned state *j* of Ph_i at time *t*, $P(X_t|Ph_{ij})$ and $P(X_t)$ are pre-calculated in equation (1) and (4) respectively and stored in phone-state matrix. Note that $CM(Ph_i)$ is the time mean of the frame level logarithm posterior probability.

3. EXPERIMENTS AND RESULTS

In this experiment, a one hour self-recorded mandarin conversational telephone speech test set is used. All speeches are very spontaneous utterances, including laugh, stop, repeat, hesitation, etc. Since the two speakers' voices are recorded in the same channel, there are lots of speech overlaps in the recordings. Male and female speeches are both recorded, and chat topic has no any limitation.

The keyword test set consists of 100 words, including 79 short words (2 syllable) and 21 long words (3 syllable), and there are 405 keyword instances in the test speech. We do not choose the most frequently occurring words as test keywords because this is not the case in real scenarios. All selected keywords have very specific meanings.

Three kind of different KWS methods are implemented in our experiment, including HMM-based, lattice-based and the phone-state matrix method proposed in this paper. Because of the high WER of the test speech set, LVCSR and word lattice KWS methods are not considered, and we test both phoneme and syllable lattice with no language model when decoding.

All methods use same speech feature and acoustic model (AM) when spotting keywords or generating lattices. The feature is a 42-dimension vector, including 12 MFCC, 1 log-energy, 1 pitch and their 1st and 2nd derivatives. The analysis frame length and shift are 25ms and 10ms respectively. The AM is trained by 300 hours of telephone speech corpus (not including the one hour test speech).

We mainly use *false alarms* (*FA*) and *false rejects* (*FR*) to evaluate different KWS systems. When KWS system detected a putative keyword, a CM score is given with it. If we preset a CM threshold score *C*, only some putative keywords with CM score greater than C are reported. Among these reported keywords, some are incorrect, which are called *false alarms*; and still some keywords are not reported due to low CM score or beam pruning, which are called *false rejects*. The *FA* ratio and *FR* ratio are usually defined as:

$$FA = (fa/kw/hr/m)*100\% \qquad (7)$$

$$FR = (fr/n)*100\% \qquad (8)$$

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 the false alarms of each keyword will be greater than *m* in average when the system processes one hour speech. In this experiment, we use *m*=10.

3.1. Performance Comparison

Fig. 3 illustrates the detection error tradeoff (DET) curves of the four KWS systems tested in our experiments.

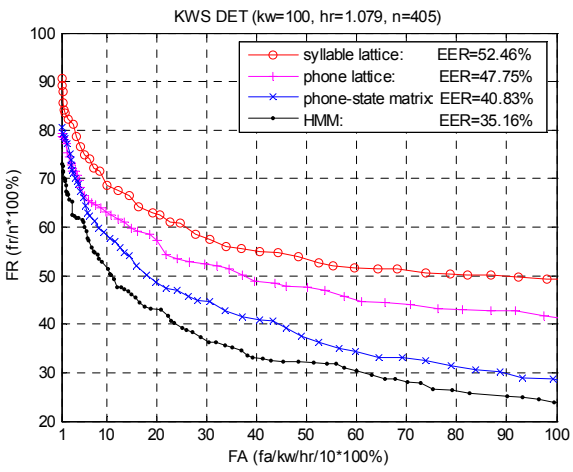

Fig. 3. DET Curves of the four KWS systems

It can be seen from this figure that phone-state matrix approach has much better performance than syllable and phoneme lattice ones. Phone-state matrix method is an approximation of HMM-based method, and precision loss is reasonable.

Method	EER	Loss	FR (FA=100)	Loss
syllable lattice	52.46%	17.30%	49.3%	25.4%
phoneme lattice	47.75%	12.59%	41.4%	17.5%
phn-state matrix	40.83%	5.67%	28.5%	4.6%
HMM	35.16%	-	23.9%	-

Table 1. EER and FR Comparison of the four systems

Table 1 shows EER and FR (FA=100) performance of four systems. It can be seen that phone-state matrix system has quite less EER and FR loss than lattice ones.

3.2. Time Performance Comparison

Method	time/kw/hr (seconds)
syllable lattice	0.2 s
phoneme lattice	0.48 s
phone-state matrix	0.5 s
HMM	-

Table 2. Time Comparison of the four systems

Time performance of each KWS system is shown in Table 2. The average time of searching one keyword in one hour speech is recorded. Result shows that the phone-state matrix system is fast enough comparable with the lattice-based systems. Note that time performance of lattice-based systems can be improved by adopting indexing techniques [8]. In this test, only linear search is used with no indexing.

The HMM-based system is relatively slow for its vocabulary-dependent nature. In this experiment, it runs at about 0.5 real times, which is also the running speed of the first pass of phone-state matrix system. Only considering the second search pass, the phone-state matrix system runs at 1/720 real time speed when keyword number is 10, which is about 360 times faster than the HMM-based system.

3.3. Storage Consumption

The original speech data is about 57.6 megabytes per hour which is in raw wave format with 8k/16bit sample rate. HMM-based method has no need for external storage support if the original speech is always available, whereas Phone-state matrix, syllable and phoneme lattice methods have to save intermediate results for second pass search.

The size of the phone-state matrix is related to phoneme table size and HMM output state number. We can calculate the size as:

$$M = (P * S + 2) * T \quad (9)$$

Where M is the matrix size, P is phoneme number, S is output state number of every phoneme HMM, T is total analysis frame number. P and S are AM parameters. In our tests, P is 61 and S is 3.

In order to reduce the matrix and lattice size, vector quantization (VQ) and compression are used. In this experiment, we use codebook size of 127 with almost no performance loss. Some simple data compress methods are also applied to save more space.

Method	size/hr
syllable lattice	11.6 Mb
phoneme lattice	29.4 Mb
phone-state matrix	40.7 Mb
HMM	0
original speech	57.6 Mb

Table 3. Storage Comparison of the four systems

Storage consumptions of the four KWS systems are listed in table 3. It can be seen from the table that storage requirements of all lattice and matrix methods are less than the original speech data size. Though phone-state matrix system has the largest extra storage cost, it's acceptable in some practical environments such as TV program searching.

4. CONCLUSION

In this paper, we introduce a novel vocabulary-independent KWS method based on phone-state matrix for fast keyword spotting in spontaneous speech. Experiment results show that the proposed method provides very fast keyword searching while maintaining respectable performance. Compared to the high precision HMM-based system, the phone-state matrix system has far less EER and FA precision loss than syllable and phoneme lattice KWS systems.

5. 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] J.E. Liang and M. Meng et al, "An Improved Mandarin Keyword Spotting System Using MCE Training and Context-enhanced Verification", *Proc. ICASSP*, SLP-P19, pp I-1145-1148, 2006.

[3] I. Szoke and P. Schwarz et al, "Comparison of Keyword Spotting Approaches for Informal Continuous Speech", *Proc. 9th Eurospeech*, pp. 633-636, 2005.

[4] M. Saraclar and R. Sproat, "Lattice-based Search for Spoken Utterance Retrieval", *Proc. HLT-NAACL*, 2004.

[5] O. Siohan and M. Bacchiani, "Fast Vocabulary-Independent Audio Search Using Path-Based Graph Indexing", *Proc. 9th Eurospeech*, pp. 53-56, 2005.

[6] K. Thambiratnam and S. Sridharan, "Dynamic Match Phone-lattice Searches for Very Fast and Accurate Unrestricted Vocabulary Keyword Spotting", *Proc. ICASSP*, SP-P5, pp. I-465-468, 2005.

[7] 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.

[8] P. Yu and K. Chen et al, "Vocabulary-Independent Indexing of Spontaneous Speech", *IEEE Transactions on Speech and Audio Processing*, Vol.13,No.5, Sep 2005.