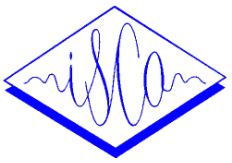

6th International Conference on Spoken Language Processing (ICSLP 2000)
Beijing, China
October 16-20, 2000

A NEW APPROACH FOR MODELING OOV WORDS

REN Weimin, WANG Chengfa, GAO Wen, Xu Jinpei

Speech Processing Laboratory, Department of Computer Science and Technology, Harbin Institute of Technology , Harbin
rwm@srg.hit.edu.cn, wcf@srg.hit.edu.cn

ABSTRACT

This paper addressed the problem of Out-Of-Vocabulary (OOV) utterance detection in small vocabulary telephone keyword spotting system. We propose a new approach for modeling OOV words in the scenario of a small vocabulary of telephone keyword spotting system. The paper adopt the semi-continuous Hidden Markov Model with multiple codebooks to modeling the keywords. We propose a two pass procedure to spot the real keyword occurrence. In the first pass, the normal viterbi search procedure is applied, with the appropriate defined and trained garbage models and silence models. The output of this stage produces the N-best word hypothesis The second pass, which can be seen as a verification procedure, take the first pass output as focuses. This approach is mainly constructing a "dynamic anti-model" based on the detected hypothesis keyword model and the current input acoustic information.

1. INTRODUCTION

It' s well known that the problem of garbage modeling, utterance verification, and OOV detection is important and difficult to be solved by automatic speech recognition systems, and it becomes more crucial when one have to deal with the speech recognition under spoken speech, telephone speech environment. There have been many researches and reports for solving this problem.

The template-based dynamic programming was proposed by bridle[1]. Each reference template matches with every part of the utterance, all putative hit are processed and some threshold is applied. In HMM-based methods, a background noise model is adopted to calculate likelihood score ratio of keyword[2]. For keyword spotting tasks, some researchers proposed garbage model or filler model. These model are used to model the non-keywords[3,4,5]. The detector' s output is composed of keywords and non-keywords string, each keyword occurrence is one "putative hit". This method need more "garbage" data to train the corresponding models. In addition, the speech space must be carefully treated. In order to resolve this problem, In our telephone keyword spotting system EasyTalk, we adopt the semi-continuous Hidden Markov Model with multiple codebooks to modeling the keywords. Each keyword was modeling by one SCHMM whole word model. The state output pdf is mixture of Gaussian functionusing three different codebook, and the feature consists of 12 cepstral plus 1 normalized energy, and their first and second dynamic features. We proposed a two pass procedure to spot the real keyword occurrence. In the first pass, the normal viterbi search procedure is applied, with the appropriate defined and trained garbage models and silence models. the output of this stage produce the N-best word hypothesis The second pass, which can be seen as a verification procedure, take the first pass output as focuses.

In this verification procedure, our new approach is mainly constructing a "dynamic anti-model" based on the detected hypothesis keyword model and the current input acoustic information. This procedure can be described as follow: For the k' th hypothesis keyword utterance, we recorded the viterbi path and the corresponding score of whole word. Then, a new scan is executed along exactly the recorded path. The scoring method for this "anti-model" pass is based on the current acoustic feature, instead of the hypothesis keyword model: In each state of the recorded path, the state emission probability can be calculated with a way like normal SCHMM state probability. For each input frames, first we calculate the probability over all code words, the sorted the top M probabilities. Next we compare the keyword top M code words with above Top M dynamic code words, and count the same code word numbers N. A "similarity weight function" W can be defined according to the two M sets and N, W=F(N/M). Now with this weight function, we can dynamic calculate each state' s mixture weight C: C = Fc(W, C'), in which C' is the keyword model' s mixture weight in this state. Go along the recorded path, a new dynamic viterbi-like score can be given for "anti-model". At last, based on the comparison of two score, some decision rule can be adopt to decided whether or not accept the hypothesis keyword spotting.

2. ACOUSTIC MODELS

The proposed OOV modeling method is based on semi-continuous hidden Markov models. We use three multiple codebooks SCHMM to model the basic word. Feature vector is 39-dimension RASTA-PLPs[7]. Model' s topology is left-to-right structure without state jump. In each state, the output is modeled by mixture Gaussian pdf' s. So the probability of observing vector

O_t in I' th state at time t can be given as follows:

$$\Pr(O_t|i)=\prod_{k=1}^{K}\sum_{m=1}^{M_k}c_{im}^{k}N^{k}\left(O_t^{k};m_m^{k},\Sigma_m^{k}\right) \qquad (1)$$

Where k denotes different feature stream, m is the indices of codeword in each codebook and M denotes the k' th codebook' s size.

In our system , for each keyword, there is a SCHMM model. In order to deal with silence segment , background noise and some typical non-speech signals in input speech utterance, we also build other HMM models, such as silence model, background noise model and cross word models. They are trained like normal keyword.

SCHMM models' training can be done using the following algorithms.

$$\bar{a}_{ij}=\frac{\sum_{t=1}^{T}g_t(i,j)}{\sum_{t=1}^{T}\alpha_{t-1}(i)\beta_{t-1}(i)} \qquad (2)$$

$$\bar{c}_{im}^{k}=\frac{\sum_{t=1}^{T}\sum_{k=1}^{N}\xi_t(k,i,m)}{\sum_{t=1}^{T}\sum_{j=1}^{N}g_t(j,i)} \qquad (3)$$

$$\bar{m}_m^k=\frac{\sum_{t=1}^{T}\sum_{k=1}^{N}\sum_{i=1}^{N}\xi_t(k,i,m)\cdot O_t^k}{\sum_{t=1}^{T}\sum_{k=1}^{N}\sum_{i=1}^{N}\xi_t(k,i,m)} \qquad (4)$$

$$\bar{\Sigma}_m^k=\frac{\sum_{t=1}^{T}\sum_{k=1}^{N}\sum_{i=1}^{N}\xi_t(k,i,m)\cdot\left(O_t^k-m_m^k\right)\left(O_t^k-m_m^k\right)'}{\sum_{t=1}^{T}\sum_{k=1}^{N}\sum_{i=1}^{N}\xi_t(k,i,m)} \qquad (5)$$

3. SPOTTING KEYWORD

In keyword spotting procedure, we used two kind of model as system reference model. By choosing non-keyword model, we can easily apply continuous speech recognition algorithm as the first pass searching procedure. This pass can give us a string of keyword and non-keyword output string. We call these non-keyword models as garbage models or filler models. The keyword detector is illustrated as figure 1.

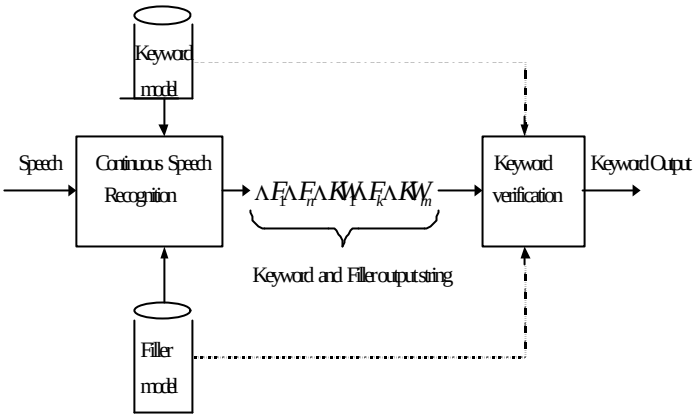

figure 1. the keyword detector

The input utterance first be processed by a continuous speech recognizer. The recognizer' s vocabulary is composed of keyword and filler models, and the recognition result is keyword and filler word string. Actually, this output can be N-best word or word lattice.

Suppose there is N keywords and M filler models. The continuous speech recognizer' s task is to find a path from the starting utterance to the end utterance. All the keyword and filler in this path composed of the output string.

1) . Initialize

For k=1 To K

$\alpha(1,k)=\pi(1,k)=1.0$

$B(1,k)=0$

For j=2 To J(k)

$\alpha(j,k)=\pi(j,k)=0.0$

$B(j,k)=0$

End j

End k

T(0)=0, F(0)=0, $\alpha_{\max}=0$

2). iteration

For t=1 To T_0

For k=1 To K

For j=1 To J(k)

a) inner-word rules

if j=1

if($\alpha(1,k)\cdot a_{11}(k)<\alpha_{\max}$)

$\alpha_t(1,k)=\alpha_{\max}\cdot b_j^k(O_t)$

$B_t(1,k)=t-1$

else

$$\alpha_t(1,k) = \alpha_{\max}$$

$$B_t(1,k) = B_{t-1}(1,k)$$

End if

b) Inter-word rules

if (j>1)

$$j^* = \arg_j \max\left\{\alpha(j,k)a_{ji}(k), \alpha(j-1,k)a_{j-1,j}\right\}$$

$$\alpha_t(j,k) = \alpha(j^*,k)a_{j^*j}(k)b_j^k(O_t)$$

$$B_t(j,k) = B(j^*,k)$$

End if

End k loop

T=0, $\alpha = 0$

For k=1 To k

if ($t - B_t(J(k),k) > l_k$ and $\alpha_t(J(k),k)$)

T = k ; $\alpha = \alpha_t(J(k),k)$

End if

End k

T(t)=T ; $\alpha_{\max} = \alpha$

$$F(t) = B_t(J(T),T)$$

▪ update $\alpha(j,k)$

$$\alpha(j,k) = \alpha_t(j,k) \quad ; \quad B(j,k) = B_t(j,k) \qquad \forall j,k$$

End for t loop

3). Traceback for best path

$r = T_0$; $R = 0$

while F(r) != 0

$U_R = T(r)$

$r = F(r)$

$R = R+1$

End while

And the best path is:

$$U = U_1 \oplus U_2 \oplus \Lambda \oplus U_R$$

Because this algorithm need only scan the input utterance one pass, it's called one-pass decode algorithms. Our experiment shows that it has good performance for small vocabulary decode tasks

4. SECOND PASS FOR OOV VERIFICATION PROCEDURE

The second pass search is based on the first pass output string. It aims at picking up the real keyword and discarding false hypothesis. This is an OOV verification procedure. The simplest way is applying threshold to make decision. But the first pass don't give an absolute score for each hypothesis, the threshold decision method really give poor performance. A modified version of the threshold is logarithm likelihood ratio method. In this approach, a background model is build. The ratio of scores for keyword model and background model is used to determine acceptance or refusal of a hypothesis. Obviously, the system's performance is greatly affected by the background model.

In this section, we proposed a new modeling method for out-Of-Vocabulary word. This algorithm is based on keyword model knowledge and the current input utterance. A new 'model' is constructed dynamically, and the current utterance segment is scored by this model. We call this new model "dynamic anti-model". This algorithm can be described as following figure 2.

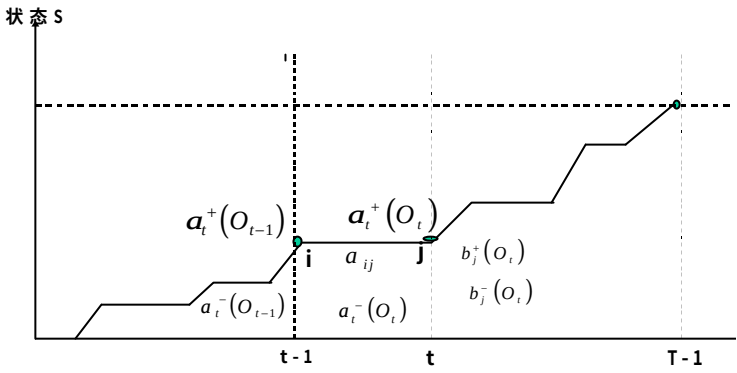

Figure 2 Viterbi decoding path

(1). In figure 2, suppose the frame 0 to frame T-1 belong to some keyword, say k' th keyword. For this T frames segment, we can get its viterbi decoding path $Path = s_1 s_2 \Lambda\ s_{t-1} s_t \Lambda\ s_{T-1}$, and $s_t \in [0, J(k)-1]$. This path' s probability score, we denote it as $s^+(O)$. Similarity, we will denote all symbols related to keyword' s path as a plus + symbal.

(2) Along this viterbi path, we can calculate the dynamic anti-model' s score as follows: At t-1 time, the partial path for dynamic model is $\alpha_t^-(o_{t-1})$, then at time t, path score can be calculated by $\alpha_t^-(o_t) = \alpha_{t-1}^-(o_{t-1}) \cdot a_{ij} \cdot b_j^-(o_t)$.

(3) calculating $b_j^-(o_t)$

For the hypothesis model k, we could know its probability in state I at time t :

$$b_j^+(o_t) = \sum_{m=1}^{M} c_{jm} N(o_{jt} u_{jm} \Sigma_{jm}) \qquad (6)$$

Here we only consider one codebook situation. $M^{'}$ is integer less than codebook' s size M. In general, $M^{'}$ is far less than M. c_{jm} Is the jm' th Gaussian pdf ' s mixture weight, and

$$0 \le c_{jm} \le 1 \wedge \sum_{m=1}^{M^{'}} c_{jm} = 1$$

For current input vector 0_{tj} , calculate all its probabilities over all codewords, $p_1, p_2, \Lambda\ p_M$

$$p_m = N(o_{tj} u_m \Sigma_m) \qquad (7)$$

Thus we can get M Ps. For $p_1 \sim p_m$, we sort them and take top M' Ps set: $j^- = \left\{ j_1^-, j_2^- \Lambda\ j_{M^{'}}^- \right\}$. j is index of codeword in codebook.

Suppose the hypothesis keyword k' s Top M' j is:

$$j^+ = \left\{ j_1^+, j_2^+ \Lambda\ j_{M^{'}}^+ \right\}$$

Denote above two sets intersection set as C:

$$C = j^+ \cap j^-$$

At this time , we can calculate the "dynamic model" probability of producing vector O in state j at time t as:

$$b_j^-(o_t) = \sum_{m=1}^{M} c_{jm}^- N(o_{jt} u_{jm}\ \Sigma_{jm}) \qquad (8)$$

Assume the number of elements in set C is N' , and the numeber of elements in Set J^+ is N, then a similarity weight can be define:

$$W_s = f(N' / N) \qquad (9)$$

With this Similarity weight function, C_{jm}^- can be determined using the formula 10:

$$C_{jm}^- = \begin{cases} C_{jm}^+ \cdot W_s & \text{if } j_m^- \in J \\ C_{jm}^+ (1 - W_s) & \text{if } j_m^- \notin J \end{cases} \qquad (10)$$

After normalization of C_{jm}^-, $b_j^-(o_t)$ can be calculated. With them, the overall dynamic model' s score along the keyword k model' s viterbi path $S^-(o)$ is available now. At this point, we just decide whether accept this hypothesis or not according to the following comparison:

If $\delta = S^-(o) - S^+(o) \le \varepsilon_k$ accept

If $\delta = S^-(o) - S^+(o) > \varepsilon_k$ refuse

5. EXPERIMENTS

The EasyTalk system is designed to deal with 50 keywords, and trained with 60 minutes telephone speech database. All models are initialized with K-means method. Three codebooks are trained with LBG algorithms.

With 4fa/hr/kw the correct detection rate of 91.9% can be achieved. The further works will be done to deal with general HMM scenario using this method.

6. REFERENCES

[1] J.S. Bridle, An Efficient elastic template method for detecting keywords in running speech, British Acoustic Society Meeting, April, 1973, pp. 1-4.

[2] J.R. Rohlicek, et al., Continuous Hidden Markov Modeling for Speaker-Independent Word Spotting, ICASSP' 89, pp. 627-630.

[3] R.C. Rose, D.B. Paul, A Hidden Markov Model Based Keyword Recognition System, ICASSP' 90, pp. 129-132

[4] J.G. Wilpon, C.H. Lee and L.R. Rabiner, Application of Hidden Markov Models for Recognition of a Limited set of Words in Unconstrained Speech, ICASSP' 89, pp. 254-257

[5] J.C. Wilpon, L.R. Rabiner and C.H. Lee, Automatic Recognition of Keywords in Unconstrained Speech Using Hidden Markov Models, IEEE Trans. ASSP, 1990, pp. 1870-1878.

[6] Yang Ziyun, Han Jiqing, et al ., Study on the method and system implementation of an isolated word recognition system used in noisy environment, Chinese Journal of Acostics, 1996, Vol.15 No.2, pp123-132.

[7] H. Hermansky, N. Morgan, et al, RASTA-PLP Speech Analysis Technique, ICASSP' 92, pp121-124

[8] X. D. Huang, Phoneme Classification Using Semicontinuous Hidden Markov Models, IEEE Trans. Signal Processing, Vol. 40, No. 5, 1992, pp1062-1067.