A Robust Keyword Spotting System for Mandarin Speech

Chung-Hung Chien and Hsiao-Chuan Wang

Department of Electrical Engineering
National Tsing Hua University
Hsinchu, Taiwan, 30043

Abstract

This paper introduces a method for designing a robust Mandarin keyword spotting system. Keywords which will be extracted from an uttered sentence are modeled by sequences of states. These state models that represent the subsyllables of Mandarin speech are generated by using the existing speech database. The non-keyword portions of an input utterance are filtered out by filler models. A simplified signal bias removal technique is applied to overcome the influences due to channel distortion and speaker variation. State integrated Wiener filters are used for noise compensation. Proposed techniques are evaluated by several experiments to show their effectiveness for robust speech recognition.

1 Introduction

In many applications, an input utterance can be recognized by extracting its keywords without transcribing all the sentence. This keyword spotting technique allows a speaker to talk to a machine naturally. Without complicated recognition algorithm, such as the continuous speech recognition, the keyword spotting method provides an alternate implementation of speech input. For small vocabulary applications, keyword spotting is an effective method for implementing a voice input system. Many researchers have been attracted into this area and

---

This research has been sponsored by the National Science Council, ROC, under contract number NSC-862745-E-007-010.

developed some remarkable keyword spotting methods [Wilpon, Miller, and Modi 1991, Wilcox and Bush 1992, James and Young 1994, Huang, Wang, and Soong 1994, Sukkar and Lee 1996].

A typical keyword spotting method is based on hidden Markov model (HMM) technique [Wilpon, Rabiner, Lee, and Goldman 1990] where a word or a subword is modeled as a Markov chain of states. The filler models are used for filtering out the non-keyword portion of the utterance. Usually, the filler models can be generated by using speech data of selected non-keywords. A decision making scheme must be provided for discriminating those non-keyword speech and silence in an utterance. The performance of a keyword spotting algorithm depends on the effectiveness of screening the non-keyword portion.

In this paper, a keyword spotting method for Mandarin speech is introduced. The continuous density HMM technique is applied. An utterance is modeled by a finite state network composed of keyword models and filler models. Viterbi decoding algorithm is used to find the optimal state sequence and then the score of the utterance is calculated. A likelihood ratio test is applied to extract the keyword from an input utterance. Both keyword models and filler models are generated by using the state models trained by an existing speech database [Hwang, Cheng, and Wang 1996]. This implies that a keyword spotting system can be implemented by using the existing state models so that no training procedure is necessary in building an application system.

In order to overcome the channel distortion and the speaker variation, a simplified signal bias removal (SBR) algorithm [Rahim and Juang 1996] is developed. The background noise is compensated by using state integrated Wiener filters [Vaseghi and Milner 1997]. This robust keyword spotting system has been implemented on a personal computer to demonstrate its capability in voice response applications.

2 Mandarin Syllables and their Hidden Markov Models

Mandarin speech is a tonal and syllabic language. Each Chinese character corresponds to a syllable. There are about 1300 distinctive syllables in Mandarin speech. Without concerning the tones, the number of base syllables is 408. Usually we represent a Mandarin syllable as an Initial-Final model. The *final* portion is its rhyme part, and the *initial* portion is a consonant. Some of syllables are vowel only, and no consonant appears in the *initial* part. We refer these syllables as *null-initials*. Since the acoustic characteristic of *the initial* is affected by its following *final*, we consider the *initial* a context-dependent unit. Totally, there are 38 context-independent *finals* and 99 right-context-dependent *initials* in Mandarin speech. Also, there are 33 syllables of *null-initials*. In this study, the *initials* are modeled by 3-state HMMs, and the *finals* by 4-state HMMs. For a syllable of *null-initial*, its *initial* part id modeled by a 2-state HMM. Including a silence state, there are totally 498 states must be modeled.

The speech database for generating the state models consists of 5045 phonetically balanced Mandarin words spoken by 51 males and 50 females. It includes 408 base syllables in Mandarin speech. These speech data were recorded in an office room via a high-quality microphone. The speech signal was sampled at 8 kHz with 16 bits per sample. The speech signal is pre-emphasized and then a Hamming window of 256 sampling points is applied before calculating its cepstral coefficients. The frames are spaced by 128 sampling points. For each frame, a 12-order cepstrum is extracted. A feature vector consists of 12 cepstral coefficients, 12 delta cepstral coefficients, a delta log-energy, and a delta delta log-energy. Finally, an utterance is represented by a sequence of 26-dimensional feature vectors. The state model is a mixture of Gaussian densities. 498 state models are generated using the speech database described above.

3. Scoring Method for Keyword Spotting

In this study, we assume that an input utterance contains one keyword only. Then an utterance is modeled by a network structure such that a sequence of nodes representing a keyword is preceded by a filler model and followed by another filler model. The silence state is added to the beginning and ending nodes of this network for filtering out the silence portions before and after the speech. The filler model is for filtering out the garbage speech in the utterance.

For an input utterance, Viterbi decoding is applied to calculate the maximum likelihood score and to obtain its corresponding optimal state sequence. Along the optimal state sequence, the local likelihood scores belonging to the keyword states are accumulated as a keyword score.

$$L(O^v, S_k^v) = \log P(O^v \left| S_k^v \right.) = \sum_{j=j}^{j+M_v-1} \log P(o_j^v \left| s_{k,j}^v \right.), \qquad (1)$$

where $O^v = \{o_j^v, o_{j+1}^v, \cdots o_{j+M_v-1}\}$ are the feature vectors of the frames belonging to keyword v, $S_k^v = \{s_{k,j}^v, s_{k,j+1}^v, \cdots s_{k,j+M_v-1}^v\}$ is the corresponding states belonging to keyword v, M_v is the number of frames belonging to keyword v, and j is the starting frame of the keyword. If the likelihood scores of those decoded keyword states are calculated based on filler models, we obtain a normalization score.

$$L(O^v, S_f) = \log P(O^v \left| S_f \right.) = \sum_{j=j}^{j+M_v-1} \max_{s_j \in \{S_f\}} \log P(o_j^v \left| s_j \right.), \qquad (2)$$

where S_f is a set of the filler states. Then we define a likelihood ratio as follows,

$$L(O^v) = (\log P(O^v \left| S_k^v \right.) - \log P(O^v \left| S_f \right.)) / M_v. \qquad (3)$$

This ratio will be used for determining the recognized keyword,

$$v^* = \max_v \{L(O^v)\}. \qquad (4)$$

In order to screen out those cases of abnormal keyword duration, we set a limit to bound the keyword duration in a reasonable range. When a keyword detected in an utterance is out of the bound, this keyword is wrong and the utterance is indicated as no keyword existing.

4. Compensation of Channel Effect

The channel effect may be due to a telephone line or a microphone. We can consider the channel effect a convolution noise. In frequency domain, the resulted speech signal is expressed as

$$Y(\omega) = H(\omega)X(\omega) \ , \qquad (5)$$

where $Y(\omega)$ is the distorted speech, $H(\omega)$ is the channel effect, and $X(\omega)$ is the original speech. When Eq.(5) is transformed into cepstral domain, the channel effect becomes an additive term,

$$c_y(n) = c_x(n) + \delta(n) . \qquad (6)$$

When an utterance is represented by a sequence of feature vectors, and the feature vector consists of cepstral coefficients and delta cepstral coefficients, the bias can be assumed to be an additive constant vector.

$$c_{y,t} = c_{x,t} + b \ , \qquad (7)$$

where $c_{y,t}$ is the feature vector of distorted speech in t-th frame, $c_{x,t}$ is the feature vector of original speech in t-th frame, and b is a bias vector. The procedure for finding the bias vector is as follows [Rahim and Juang 1996];

(a) Apply Viterbi decoding on the test utterance to find its optimal state sequence, $S = \{s_1, s_2, \cdots s_T\}$.

(b) Apply following equation to estimate the bias vector,

$$b = \frac{1}{T}\sum_{t=1}^{T}(c_{y,t} - m_i), \qquad (8)$$

where m_i is the mean of state model i corresponding to the decoded state s_t in t-th frame, and T is the number of frames in the utterance.

When the bias vector is obtained, we can apply this bias to adapt all the state models,

$$\tilde{m}_i = m_i + b \ . \qquad (9)$$

Viterbi decoding algorithm is applied again to the test utterance based on adapted models. This procedure, i.e. Eq.(8) and Eq.(9), can be iterated so that a converged bias vector is obtained and all the state models are adapted to new ones. Finally, the input utterance is recognized based on the new state models. Usually, two iterations is enough to obtain converged bias vector. If training utterances are used for finding this constant bias vector, the adaptation operation is not necessary during the recognition phase. Speaker adaptation is exactly similar to channel compensation with given training utterances by a specific speaker.

5. Compensation of Additive Noise

The additive noise can be modeled as adding a noise term to the clean speech in frequency domain.

$$Y(\omega) = X(\omega) + N(\omega), \tag{10}$$

where $X(\omega)$ is the clean speech, and $N(\omega)$ is the additive noise. Many noise compensation methods have been developed for compensating the additive noise. Here we use Wiener filter to minimize the effect of noise [Vaseghi and Milner 1997]. In frequency domain, Wiener filter is expressed as

$$W(\omega) = \frac{P_{xx}(\omega)}{P_{xx}(\omega) + P_{nn}(\omega)}, \tag{11}$$

where $P_{xx}(\omega)$ and $P_{nn}(\omega)$ are the power spectrum densities of original speech and noise, respectively. When a noisy speech is input to Wiener filter, the output would be

$$\tilde{X}(\omega) = W(\omega) Y(\omega). \tag{12}$$

In cepstral domain, Eq.(12) becomes

$$c_{\tilde{x}}(n) = c_w(n) + c_y(n), \tag{13}$$

where

$$c_w(n) = c_{P_{xx}}(n) - c_{P_{xx}+P_{nn}}(n) \tag{14}$$

In our speech recognition system, Wiener filter is estimated and applied to state models to adapt the models to noisy environment,

$$\tilde{m}_i = c_{P_{m_i}} - c_w, \tag{15}$$

where

$$c_w = c_{P_{m_i}} - c_{P_{m_i}+P_{nn}}. \tag{16}$$

P_{m_i} is the power spectrum density calculated for each state model during the training phase. Its corresponding cepstrum is $c_{P_{m_i}}$. P_{nn} is the power spectrum density of noise which is estimated under silence input. Once P_{nn} is obtained and P_{m_i} is available, $c_{P_{m_i}+P_{nn}}$ can be calculated. In our implementation, P_{nn} is calculated once in a stationary noisy environment.

During recognition phase, a signal-to-noise ratio (SNR) is calculated for each utterance to adjust P_{nn}.

6. Experiments

Some experiments were conducted to demonstrate the keyword spotting algorithm and the effectiveness of our channel and noise compensation methods.

Experiment 1

Twenty city names in Taiwan were designated as keywords embedded in the uttered sentences. Six speakers each provided 50 test utterances through microphones. Only one keyword was embedded in each utterance. The garbage speech might appear before and after the keyword. The accuracy was 94.7% for mixture number is 2 for each state model.

Experiment 2

Twenty city names in Taiwan were designated as keywords embedded in the uttered sentences. Thirteen speakers each provided 20 test utterances through telephone system. Only one keyword embedded in each utterance. The garbage speech might appear before and after the keyword. The mixture number was 2 for each state model. The accuracy was 57.9% without channel compensation. The accuracy increased to 82.6% when the proposed channel compensation method was applied. The improvement was 24.7%.

Experiment 3

Thirty city names in Taiwan were designated as keywords embedded in the uttered sentences. Fifteen speakers each provided 20 test utterances through microphones. Only one keyword embedded in each utterance. The garbage speech might appear before and after the keyword. The mixture number was 2 for each state model. In order to simulate various noise conditions, test utterances were added by different noises with specific SNRs. The types of noises included white noise, factory noise, car noise and babble noise. The accuracy for various SNRs was summarized in the following table.

Table: Recognition Accuracy (%)

noise type \ SNR	0dB		10dB		20dB	
	no compe	compens	no compe	compens	no compe	compens
white noise	8.57	13.2	37.0	45.4	68.2	77.5
factory noise	16.1	24.3	55.7	58.6	79.6	82.6
car noise	65.4	78.2	77.1	81.1	81.8	82.5
babble noise	10.7	19.6	48.9	59.3	73.9	78.9
AVERAGE	25.2	33.8	54.7	61.1	75.9	80.4

The effectiveness of noise compensation depends on the type of additive noise. The result shows that car noise does not influence to much on the recognition accuracy. White noise is the most serious one because it affects a wide band of signal spectrum. In average the proposed noise compensation method can gain an improvement of 4.5% to 8.6%.

7. Conclusion

A robust Mandarin keyword spotting system is presented. Keyword and filler models can be generated by using the existing speech database. This allows user to define their own application systems. A simplified signal bias removal technique is applied to overcome the influences due to channel distortion and speaker variation. State integrated Wiener filters are used for noise compensation. Experiments show that the channel compensation can gain an accuracy improvement of about 25%, and the noise compensation can improve 8.6% accuracy in average when the SNR is 0dB.

References

Huang, E.F., H.C. Wang, and F.K. Soong, "A fast algorithm for large vocabulary keyword

spotting application," IEEE Trans. SAP, vol. 2, no. 3, pp. 449-452, 1994.

Hwang, T.H., H.M. Cheng, and H.C. Wang, "Keyword spotting for Mandarin speech based on subsyllable models," Int. Conf. Multiple Information processing, Hsinchu, Taiwan, 1996.

James, D.A. and S.J. Young, "A fast lattice-based approach to vocabulary independent word spotting," ICASSP-94, Adelaide, Australia, 1994.

Rahim, M.G. and B.H. Juang, "Signal bias removal by maximum likelihood estimation for robust telephone speech recognition," IEEE Trans. SAP, vol. 4, no. 1, pp. 19-30, 1996.

Sukkar, R.A. and C.H. Lee, "Vocabulary independent discriminative utterance verification for nonkeyword rejection in subword based speech recognition," IEEE Trans. SAP, vol. 4, no.6, pp. 420-429, 1996.

Vaseghi, S.V. and B.P. Milner, "Noise compensation methods for hidden Markov model speech recognition in adverse environments," IEEE Trans. SAP, vol. 5, no. 1, pp. 11-21, 1997.

Wilcox, L.D. and M.A. Bush, "Trainig and search algorithm for interactive wordspotting system," ICASSP-92, San Francisco, CA, 1992.

Wilpon, J.G., L.G. Miller, and P. Modi, "Improvements and applications for keyword recognition using hidden Markov models," ICASSP-91, Toronto, Canada, 1991.