Computational Linguistics and Chinese Language Processing
Vol. 11, No. 1, March 2006, pp. 1-16

Using Duration Information in Cantonese Connected-Digit Recognition

Yu Zhu* **and Tan Lee***

Abstract

This paper presents an investigation on the use of explicit statistical duration models for Cantonese connected-digit recognition. Cantonese is a major Chinese dialect. The phonetic compositions of Cantonese digits are generally very simple. Some of them contain only a single vowel or nasal segment. This makes it difficult to attain high accuracy in the automatic recognition of Cantonese digit strings. Recognition errors are mainly due to the insertion or deletion of short digits. It is widely admitted that the hidden Markov model does not impose effective control on the duration of the speech segments being modeled. Our approach uses a set of statistical duration models that are built explicitly from automatically segmented training data. They parametrically describe the distributions of various absolute and relative duration features. The duration models are used to assess recognition hypotheses and produce probabilistic duration scores. The duration scores are added with an empirically determined weight to the acoustic score. In this way, a hypothesis that is competitive in acoustic likelihood, but unfavorable in temporal organization, will be pruned. The conventional Viterbi search algorithms for connected-word recognition are modified to incorporate both state-level and word-level duration features. Experimental results show that absolute state duration gives the most noticeable improvement in digit recognition accuracy. With the use of duration information, insertion errors are much reduced, while deletion errors increase slightly. It is also found that explicit duration models are more effective for slow speech than for fast speech.

Keywords: Explicit Duration Modeling, Duration Features, Connected-Digit Recognition, Cantonese, Hidden Markov Models

* Department of Electronic Engineering, The Chinese University of Hong Kong
Tel: 852-26098267 Fax: 852-26035558
E-mail: tanlee@ee.cunk.edu.hk
The author for correspondence is Tan Lee.

[Received June 15, 2005; Revised January 23, 2006; Accepted January 25, 2006]

1. Introduction

In the past two decades, automatic speech recognition (ASR) has advanced to a high performance level. The state-of-the-art technology predominantly uses hidden Markov models (HMM), which provide a nicely formulated framework for the modeling of speech signals. This framework is amenable to a set of mathematically rigorous algorithms for the estimation of model parameters and pattern classification. For ASR, an HMM consists of a number of states that are arranged into a left-to-right topology. The states can be thought of as a sequence of acoustic targets that constitute a speech segment. The output probability density functions (pdf) associated with individual states describe the spectral variability in the realization of these targets. The temporal structure is reflected mainly in the evolution of the states, which is governed by state transition probabilities.

It is widely acknowledged that an HMM does not impose effective control on the duration of the speech segment being modeled. HMM-based ASR systems frequently make errors. A significant portion of these recognition errors exhibit unreasonable time durations or duration proportions. For the task of connected-digit recognition in various languages in particular, a lot of errors are due to the insertion of short digits [Dong and Zhu 2002; Kwon and Un 1996]. The problem is extremely severe with noise-corrupted speech [Yang 2004].

Connected-digit recognition has many useful applications that often require very high recognition accuracies. Despite its limited vocabulary size, it is not straightforward to attain the desired performance level because the combination of digits is unrestricted. Knowledge sources like lexical constraints and word-level language models are not applicable in this case. Therefore, it becomes particularly important to fully exploit the information embedded in the acoustic signals. Other than the spectral features, prosodic features, like pitch and duration, can be considered.

In this paper, we focus on the use of duration information for Cantonese connected-digit recognition. Our approach uses a set of statistical duration models that are built explicitly from automatically segmented training data. The duration models are used to assess the recognition hypotheses, based on the measured duration at the either state or the model levels. As a result, a probabilistic duration score is generated and added with an empirically determined weight to the conventional acoustic score. In this way, a hypothesis that is competitive in acoustic likelihood, but unfavorable in temporal organization, is pruned.

There have been many studies on explicit duration modeling for ASR. Recognition performance can be improved to various extents. The most commonly used duration features include whole-model duration [Lee *et al*. 1989], absolute state duration [Russell and Moore 1985; Levinson 1986] and normalized (relative) state duration [Rabiner 1989; Power 1996]. The design of duration models has been application-dependent. In most cases, parametric

distributions have been used so that each duration model can be represented by a few parameters.

HMM based speech recognition is formulated as a process of searching for the optimal path among many possibilities. The optimality is measured in terms of the path's accumulated probability or likelihood. With the duration models, the conventional probabilistic path score can be modified to include the duration scores. Unlike the acoustic likelihood, duration scores are not computed on a short-time frame basis. There may be cases in which, when a path extension decision is made, some of the competing paths involve duration scores and others do not. Thus, the search is only sub-optimal. Examples of such sub-optimal methods can be found in [Power 1996].

In this work, we adopt the one-pass approach and aim for an optimal search. The conventional Viterbi search algorithm for connected-word recognition is modified to facilitate the incorporation of explicit duration models at both the state and the model levels. The effectiveness of different duration features is evaluated through recognition experiments.

In the next section, a brief introduction to the Cantonese dialect is given and the task of Cantonese connected-digit recognition is described. Baseline recognition performance is also presented. Statistical modeling of various types of duration features is described in Section 3. The ways of integrating duration models into the speech recognition processes are explained in Section 4. Experimental results are presented and discussed in Section 5. Conclusions are given in Section 6.

2. Cantonese Connected-Digit Recognition

2.1 About Cantonese

Cantonese is one of the major dialects of Chinese. It is the mother tongue of over 60 million people in Southern China and Hong Kong. Like Mandarin, Cantonese is a monosyllabic and tonal language. A Cantonese utterance is considered a string of monosyllabic sounds. Each Chinese character is pronounced as a single syllable that carries a specific tone. A character may have multiple pronunciations, and a syllable typically corresponds to a number of different characters. As shown in Table 1, each Cantonese digit is pronounced as a monosyllable sound.

Table 1. Phonetic transcriptions of the 10 Cantonese digits

Digit	IPA	LSHK
0	lɪŋ	ling4
1	jɐt	jat1
2	ji	ji6
3	sam	saam1
4	sei	sei3
5	ŋ	ng5
6	luk	luk6
7	ts^{h}ɐt	cat1
8	pat	baat3
9	kɐu	gau2

2.2 Baseline System

Our baseline system for Cantonese connected-digit recognition was trained with the CUDIGIT database, which is part of a whole series of Cantonese spoken language corpora developed at the Chinese University of Hong Kong [Lee *et al.* 1998a]. CUDIGIT is a collection of Cantonese digit strings. The data collected were all read speech. Speakers were prompted with onr digit string at a time, with Chinese characters and Arabic digits displayed in parallel on a computer screen. The recordings were carried out in a closed quiet room using a high-quality microphone. The speech signal was sampled at 16 kHz. The database contains an exhaustive permutation of digit strings from one to four syllables long. There are also randomly generated strings that are of 7, 8, and 16 digits long. A total of 25 male and 25 female speakers were recorded. Each speaker spoke about 570 digit strings.

For the acoustic models of the baseline system, the training data included 11,387 utterances from 20 male speakers. In addition, 2,847 utterances from the other 5 male speakers in CUDIGIT were reserved as development data, which were used as the estimation of the weighting factor for the duration models (see Section 5.1).

The utterances for performance evaluation were from a different database, which was recently collected for speaker recognition research. It contains Cantonese digit strings recorded under the same acoustic conditions as CUDIGIT. About 900 utterances from 5 male speakers were used in this study. In terms of the total number of digit occurrences, the amount of the evaluation data is similar to the development data.

Feature extraction was done with a 20-msec Hamming window and 10-msec window overlapping. 32 nonlinearly spaced (Mel-scale) filter banks were used to cover the bandwidth of 8 kHz and the first 12 cepstral coefficients were computed. Each feature vector had 39 components, including the 12 Mel-Frequency Cepstral Coefficient (MFCC), log-energy, and their first and second order derivatives. Cepstral liftering was applied to the cepstral coefficients.

Each Cantonese digit was modeled by a whole-word HMM. The HMM had 6 left-to-right connected states. There was no state-skipping transition. Each state was associated with a mixture of 8 Gaussian distributions. Diagonal covariance matrices were assumed. There were also a six-state "silence" model and a one-state "sp" model for the non-speech signal. The baseline recognition performance is given in Table 2.

Table 2. Baseline performance for Cantonese connected-digit recognition

Digit accuracy	Deletions	Substitutions	Insertions
95.09%	82	116	418

2.3 Discussion

As shown in Table 2, insertions and deletions accounted for over 80% of the recognition errors. It must also be noted that 68.2% of the insertion and deletion errors were due to the digits "2" and "5" [Zhu 2005]. The phonetic compositions of Cantonese digits are generally very simple. This makes it difficult to attain high accuracy in the automatic recognition of Cantonese digit strings. For example, the digit "2" can be regarded as a single vowel segment. When this digit is repetitively spoken in a continuous utterance, the boundaries between them tend to be blurred because the signal's spectrum remains virtually unchanged. This will cause deletion and insertion errors in speech recognition. Moreover, "2" is phonetically very similar to the coda part of the digit "4". It is easily confused with this coda, and recognition errors will occur.

Figure 1 shows the spectrogram of an example utterance. It contains the digit string "22" during the period of 0.5 – 0.81 sec. There is no observable spectral discontinuity that signifies the boundary between the two digits. Similarly, in the example shown in Figure 2, the coda of digit "4" is likely to be recognized as an inserted "2".

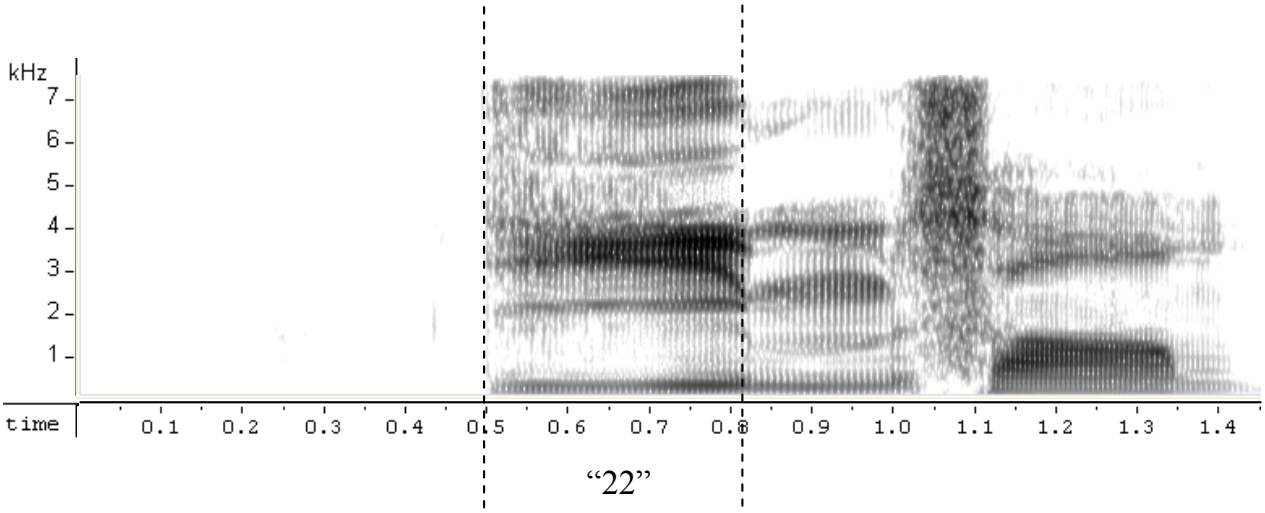


Figure 1. Spectrogram of an utterance that contains the digit string "22"

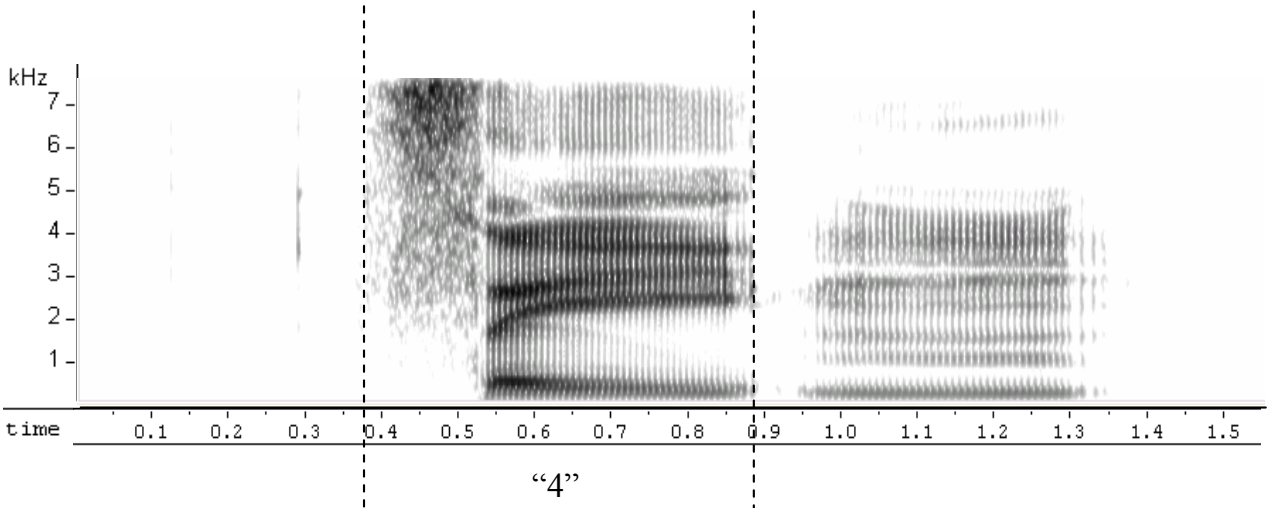


Figure 2. Spectrogram of an utterance that contains the digit "4"

Another problematic digit is "5", which can be approximated as a single nasal segment. Like "2", if the digit "5" is uttered repetitively in a continuous utterance, the spectral cues are not sufficient for detecting the digit boundaries. It is easily confused with the nasal codas of the digits "0" and "3."

Although the duration of a digit is affected by many different factors, it by no means has an unlimited range of variation, especially in those applications where the speaking style and the speaking rate are relatively stable. In the cases in which repetitive "2" or "5" segments are merged or a single segment is split, the durations of the recognized digit segments usually deviate much from their nominal values. Similar argument can be made when the string "42"

is recognized as a single digit "4" or vice versa. Prior knowledge about digit durations would be helpful to correct such errors. In addition to the absolute duration, relative duration features, e.g. the ratio between the duration of certain state(s) and that of the whole digit, are also useful. These features reflect the regularity that governs possible internal adjustments among the sub-components of a digit segment. In the next section, the statistical modeling of both absolute and relative duration features is discussed.

3. Duration Modeling for Cantonese Digits

3.1 Duration Features

Duration can be measured and modeled at segments of various lengths. The measurements of duration information are referred to as duration features. In an HMM-based system, HMMs are used to model and segment speech signals. In our baseline system, each Cantonese digit was modeled by a whole-word HMM. Given a digit string, the durations of individual digits were given directly by the model-level segmentation. State durations were derived from the state-level time alignment.

Both state duration and model duration have been found to be useful for speech recognition, but their effectiveness varies across applications. It was reported that the use of the relative state duration (with respect to the model duration) leads to better recognition performance than the use of the absolute state and model durations [Power 1996].

In this study, both the absolute state duration and the absolute digit duration were investigated. As for the relative duration features, the relative state duration (with respect to the digit duration) and the so-called tail part ratio were used. The tail part ratio measures the relative duration of the tail part of a digit. The tail part is defined to cover the last two states of an HMM. The tail part ratio can be considered a variation of normalized state duration. From the baseline recognition results, it is observed that the tail part corresponds roughly to the last phonetic unit of the digit. As mentioned in Section 2.3, the two mono-phone digits, i.e., "2" and "5", are easily confused with the tail part of other digits. When the tail part is deleted or prolonged, the tail part ratio becomes unreasonable.

3.2 Statistical Modeling

In [Russell and Moore 1985], Poisson distribution was used to model state duration. While the model is simple to estimate (only one free parameter), it is not generally applicable because it demands that the variance be equal to the mean. It was found that Gaussian and Gamma distributions are more appropriate [Levinson 1986]. In [Gadde 2000], a mixture of Gaussian distributions was used to model multivariate duration features. In [Burshtein 1995], it was shown that Gamma distribution fits the empirical data better than the Gaussian distribution for

both state and model durations. In [Dong and Zhu 2002], it was also found that duration models using Gamma distributions are superior to other parametric distributions in terms of speech recognition accuracy.

Figure 3 shows the empirical distribution of the absolute duration of digit "0" as well as the corresponding Gamma fit. The empirical distribution was obtained through supervised segmentation (also known as forced alignment) of the training data in CUDIGIT. It can be seen that the Gamma distribution fits the empirical measurements quite well. This is also true for all other digits [Zhu 2005].

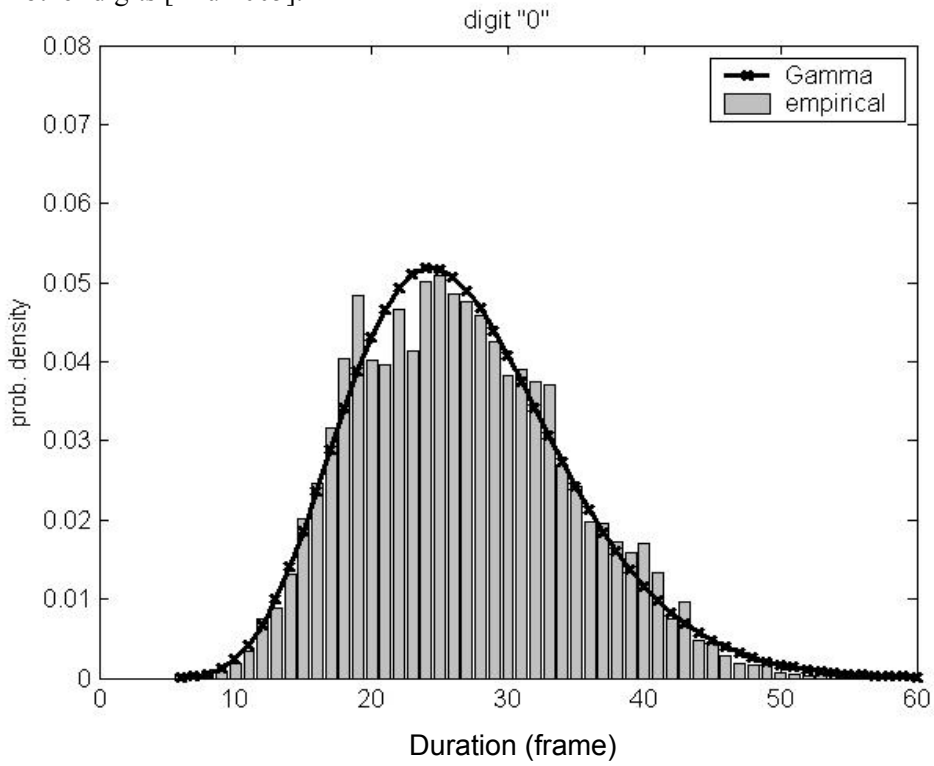


Figure 3. Distribution for the absolute digit duration for the digit "0"

For each HMM state, there is one distribution for absolute state duration and one distribution for relative state duration to be modeled. Thus, the total number of state duration distributions is 120. More than 70% of these empirical distributions can be approximated quite well as Gamma functions [Zhu 2005]. The distributions that do not fit well have complicated shapes, e.g., multi-modal. Similar observations are made concerning the modeling of relative state duration. For simplicity, uni-modal Gamma distribution is used in all state duration models.

As for the tail part ratios, the empirical distributions can all be nicely modeled with uni-modal Gamma functions. One of the examples is given in Figure 4.

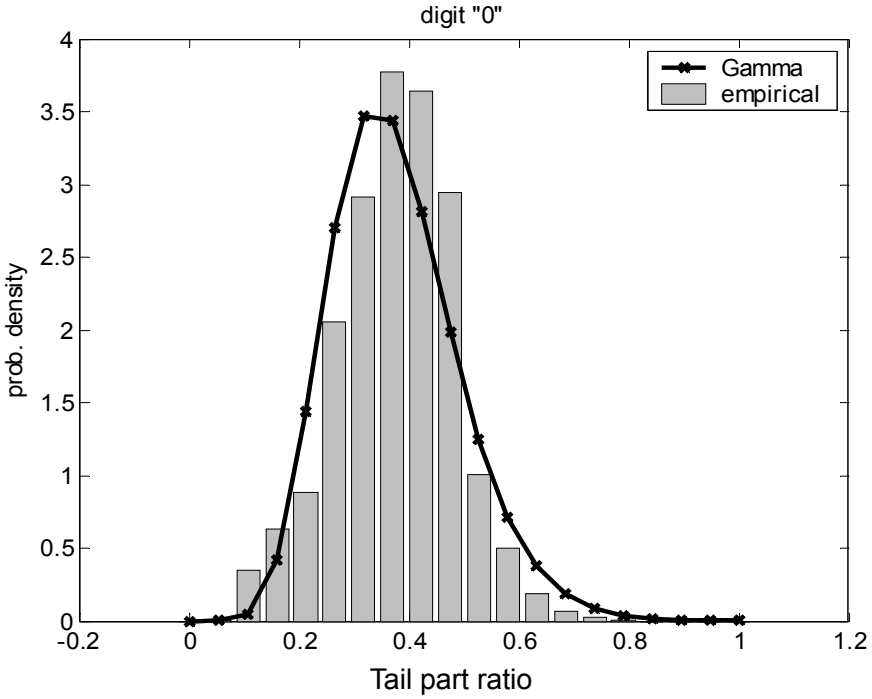


Figure 4. Distribution of the tail part ratio for the digit "0"

3.3 Training of Duration Models

In this study, Gamma distribution was used for the statistic modeling of duration features. Ther training of a duration model refers to the process of estimating the parameters of the Gamma distribution from segmented training utterances. Given a large amount of training utterances, manual segmentation at the word (digit) level is not realistic, let alone at the state level. Supervised automatic segmentation can be done with a set of acoustic models (HMMs) that are trained beforehand. This is referred to as the multi-pass training approach.

To obtain a truly optimal solution, the parameters of duration models must be estimated jointly with the HMM parameters, because they depend on each other [Russell and Moore 1985; Levinson 1986]. This one-pass approach is computationally expensive. Moreover, it is not applicable when sophisticated duration features, like relative state duration, are being modeled. Experimental results also showed that multi-pass training can be just as effective as one-pass training in terms of recognition performance [Rabiner 1989]. In this study, the duration models were trained through the multi-pass approach.

In summary, for our study regarding Cantonese connected-digit recognition, explicit duration models were established for the absolute digit duration, the absolute state duration, the relative state duration, and the tail part ratio. Each duration model was represented by a Gamma distribution, which was trained with CUDIGIT training data through the multi-pass

approach. In the subsequent discussion, the abbreviations in Table 3 are used to refer to the different duration features.

Table 3. Different duration features

AD	Absolute digit duration
AS	Absolute state duration
RS	Relative state duration
TR	Tail part ratio

4. Integrating Duration Models into Speech Recognition

As described earlier, the problem of connected-word recognition concerns the search for an optimal word string among many possibilities. The search space is formed by the HMM states, and a word string is in this way essentially a path connected by the states. The basic idea behind incorporating duration models into the search process is to make the duration probabilities contributive to the path probability. The challenge is to ensure that each path extension decision is optimal, considering that the duration probability is computed in a different time scale from the acoustic probability.

In the conventional Viterbi algorithm [Ney 1984], the problem of searching for an optimal complete path can be decomposed into many sub-problems at the frame level. The sub-problem at a particular frame t is to find the optimal partial path extended to each legitimate state. Let (t, v, j) denote the optimal partial path extended to state j of model v and at frame t. The accumulated path score is denoted by $L(t, v, j)$. The sub-problem at frame t can be solved given the solutions to the sub-problems at $t-1$, i.e., the immediately preceding frame. The path extension algorithm is explained as follows:

1) If the path is extended to the first state of an HMM, the predecessor can be the last state of any HMM or the current state itself. The path extension is done by,

$$L(t,v,1) = \max_{u} \left\{ L(t-1,u,N) \times a_{N,N+1}, L(t-1,v,1) \times a_{11} \right\} \times b_1(o_t), \tag{1}$$

where N is the number of states in the model and $a_{N,N+1}$ is the probability of exit from state N. Here we assume that all HMMs have the same number of states.

2) For a path extended to state j of a model, where $j \neq 1$, we have

$$L(t,v,j) = \max_{\substack{i=j \text{ or} \\ i=j-1}} \left\{ L(t-1,v,i) \times a_{ij} \right\} \times b_j(o_t). \tag{2}$$

That is, the predecessor can be either state j itself or state j-1 of the same HMM, because we have assumed there is no state skipping.

The path extension is performed with a step size of one frame. To incorporate the duration model scores, the path extension needs to cover a longer time span. For state-level duration features, it should cover the time span of an HMM state. For word-level features, it should cover the span of a model.

4.1 Incorporation of State-Level Duration Model

For state-level duration features, the duration scores can only be computed if there is a state transition. In this case, the notion of path extension is defined differently. The step size of the path extension is a state instead of a frame. The path extension stretches from the beginning frame of one state to the beginning frame of another state. The state duration is a variable that affects the path extension decision.

Let (t,v,j) denote the optimal partial path that extends to state *j* of model *v* at frame *t*, and $L(t,v,j)$ be the corresponding accumulated path score. Accordingly, the path extension algorithms are modified as follows:

1) When the path gets to the first state of an HMM, its predecessor can be the last state of any other HMM. For each possible predecessor $(t-d,u,N)$, the duration score $D_{u,N}(d)$ is computed, where d is the duration of staying at state N. $D_{u,N}(d)$ is incorporated into the path extension decision as

$$L(t,v,1)=\max_{\substack{u \\ d_{\min}\le d\le d_{\max}}}\left\{L(t-d,u,N)\times a_{N,N+1}\times\prod_{t-d<\tau<t}b_N(o_\tau)\times\left[D_{u,N}(d)\right]^w\right\}\times b_1(o_t), \qquad (3)$$

where d_{max} and d_{min} are the upper and lower bounds, respectively, of the state duration value, and *w* is an empirically determined weighting factor that controls the relative contribution of the duration scores.

2) For the path extension from state $j-1$ to state j of an HMM, where $j\neq 1$, we have

$$L(t,v,j)=\max_{d_{\min}\le d\le d_{\max}}\left\{L(t-d,v,j-1)\times a_{j-1,j}\times\prod_{t-d<\tau<t}b_{j-1}(o_\tau)\times\left[D_{u,j-1}(d)\right]^w\right\}\times b_j(o_t), \qquad (4)$$

In this case, all competing path extensions are from state $j-1$ to state j. They differ from each other in terms of the time instant at which the extension occurs, which is specified by the value of d.

The above formulation is referred to as the 3-dimensional optimal decoder, because the token (t,v,j) has three elements. As seen in Eqs. (3) and (4), each possible path extension involves the computation of $\prod_{t-d<\tau<t}b_i(o_\tau)$. If the paths are evaluated individually, there are a lot of duplicated computations. To alleviate this problem, the search algorithm is

re-formulated. A new dimension "*d*" is introduced into the path token. The token (t, v, j, d) refers to a path that has stayed at state *j* in HMM *v* for *d* frames at frame *t*. Equations (3) and (4) can be written as

$$L(t,v,1,1) = \max_{\substack{u \\ d_{\min} \le d \le d_{\max}}} \left\{ L(t-1,u,N,d) \times a_{N,N+1} \times \left[D_{u,N}(d) \right]^{w} \right\} \times b_1(o_t), \tag{5}$$

$$L(t,v,j,1) = \max_{d_{\min} \le d \le d_{\max}} \left\{ L(t-1,v,j-1,d) \times a_{j-1,j} \times \left[D_{u,j-1}(d) \right]^{w} \right\} \times b_j(o_t), \tag{6}$$

$$L(t,v,j,d) = L(t-1,v,j,d-1) \times b_j(o_t). \tag{7}$$

Such a 4-dimensional formulation is equivalent to the decoding framework in [Gu *et al.* 1991]. The computation cost of this decoder is d_{max} times that of the baseline decoder.

4.2 Incorporation of Word-Level Duration Models

To incorporate word-level duration scores, the step size of a path extension is defined to be a word (an HMM). A path extension is from the beginning frame of one word to that of another word. Let (t,v) denote the optimal partial path that extends to HMM *v* at frame *t*, and let $L(t,v)$ be its path score. The path extension decision is obtained as follows:

$$L(t,v) = \max_{\substack{u \\ d_{\min} \le d \le d_{\max}}} \left\{ L(t-d,u) \times warp(u,t-d,t-1) \times \left[D_u(d) \right]^{w} \right\}, \tag{8}$$

where $warp(u,t-d,t-1)$ is the probability that the sub-sequence of feature vectors from *t-d* to *t-1* is generated by HMM *u*, and d_{max} and d_{min} are the upper and lower bounds, respectively, of a word duration. $D_u(d)$ is the word-level duration score given by HMM *u*. It can be contributed by one or more duration features, including AD, RS, and TR as described in Section 3.1. For RS, it is assumed that the relative durations of individual states are independent of each other and the overall duration score is given by the multiplication of the probabilities obtained at all states.

Similar to the state-level case, the 4-dimensional formulation of the above algorithm is given as

$$L(t,v,1,1) = \max_{\substack{u \\ d_{\min} \le d \le d_{\max}}} \left\{ L(t-1,u,N,d) \times a_{N,N+1} \times \left[D_u(d) \right]^{w} \right\} \times b_1(o_t), \tag{9}$$

$$L(t,v,j,d) = \max_{\substack{i=j \text{ or} \\ i=j-1}} \left\{ L(t-1,v,i,d-1) \times a_{ij} \right\} \times b_j(o_t), \tag{10}$$

where (t,v,j,d) refers to a path that has stayed at state j of HMM v for d frames. The computation cost of this decoder is d_{max} times that of the baseline. Since word duration is much larger than state duration, the computation load of integrating word-level duration features is much heavier than that with state-level features. Such a 4-dimensional formulation is equivalent to the decoding framework in [Kwon and Un 1996].

5. Experimental Results and Discussion

5.1 Effectiveness of Different Duration Features

Experiments on Cantonese connected-digit recognition were carried out to evaluate the use of different duration features and their combinations. In all the experiments, the acoustic models were the same as those in the baseline system. The features and weights in the experiments are listed in Table. It is observed that the acoustic scores produced by the HMMs have a much wider dynamic range than the duration scores. Therefore, the effect of duration models tends to be overshadowed by that of HMM. In this work, a positive weighting factor *w* is used to balance the situation. For each of them, the weighting factor *w* for the duration scores was empirically determined from the development data (see Section 2.2). Different values of weights were tested and the one with the best results are shown as in Table 4. The values of d_{max} are 15 and 80 for state-level and word-level models, respectively.

Table 4. List of duration features and the respective weights for duration scores

Duration features		*w*
State-level	AS	3
Word-level	AD	6
	RS	4
	TR	4
	AD+RS	6, 2
	AD+TR	6, 4

In addition, an experiment was performed using the word insertion penalty method, which is commonly used to reduce insertions [Huang *et al.* 2001]. The penalty value was also determined empirically from the development data.

The experimental results are given in Table 5. In all cases, the recognition accuracy is improved compared with the baseline system. The most significant improvement is 2.36% in terms of digit accuracy, which is attained by using the absolute state duration. The performance improvement results mostly from the reduction in insertion errors, and the

substitution errors also decreased. Meanwhile, more deletion errors are produced. The use of the word insertion penalty method can also improve recognition accuracy. However, it is not as effective as the explicit duration models.

Table 5. Recognition performance with different duration features

Method of duration control		Accuracy	Deletions	Substitutions	Insertions
Baseline		95.09%	82	116	418
State-level	AS	97.45%	105	88	127
Word-level	AD	96.70%	132	100	182
	RS	96.74%	98	108	203
	TR	96.11%	81	100	308
	AD+RS	97.22%	142	90	116
	AD+TR	97.24%	133	90	124
Insertion penalty		96.37%	124	117	215

The absolute state duration (AS) gives a better recognition performance than any of the word-level features. Since the incorporation of a state-level duration model requires much less computation, it is more preferable than the word-level duration models.

Among the three word-level features, the relative state duration (RS) is the most effective, while the tail part ratio (TR) gives little improvement. The combined use of word-level features, e.g., AD+RS and AD+TR, attains a similar performance to AS. This implies that RS and TR carry certain complementary information to AD.

5.2 The Effect of the Speaking Rate

It is obvious that duration features depend greatly on the speaking rate. We divided the evaluation utterances evenly into three categories based on their speaking rates. The speaking rate was defined based on normalized word duration as described in [Lee *et al.* 1998b]. For each category, a set of speaking-rate dependent duration models were built.

Table 6 shows the recognition performance for each speaking rate category. It is noted that the use of duration models is most effective for slow utterances, though improvement is observed in all categories.

Table 6. Recognition accuracy (%) for different speaking rates

Method of duration control		Fast	Medium	Slow
Baseline		96.19%	94.79%	93.57%
State-level	AS	96.77%	97.96%	97.40%
Word-level	AD+RS	96.45%	97.70%	97.40%
	AD+TR	96.42%	97.89%	96.92%

6. Conclusions

HMM does not give effective control over duration. For speech recognition tasks in which high-level linguistic constraints are not applicable, the duration of speech segments is a useful cue that supplements the conventional spectral features. In this work, we have shown how duration features can be used to improve the accuracy of Cantonese connected-digit recognition.

Among all of the duration features investigated, the absolute state duration gave the most noticeable performance improvement. A similar level of performance was also achieved with the combined use of absolute digit duration and relative state duration. With the use of duration information, insertion errors were much reduced, while deletion errors increased slightly. The reduction in insertion errors is particularly critical for Cantonese speech recognition because many of the short syllables in Cantonese are likely to be inserted if there is no duration control. Our experimental results also revealed that explicit duration models were more effective for slow speech than fast speech.

To incorporate duration models into the speech recognition process, the standard Viterbi search algorithm has to be modified. To ensure that the search is optimal, a larger step size for path extension is needed so as to accommodate the long time-span required for computing the duration scores. This leads to a significant increase in the computation load. To reduce the computation load, a sub-optimal search can be considered.

Acknowledgement

This research was partially supported by a research grant from the Hong Kong Research Grants Council (Ref: CUHK4206/01E).

References

Burshtein, D., "Robust parametric modeling of durations in Hidden Markov Models," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1995, pp 548-551.

Dong, R. and J. Zhu, "On use of duration modeling for continuous digits speech recognition," In *Proceedings of the International Conference on Spoken Language Processing,* 2002, pp. 385-388.

Gadde, V. R. R., "Modeling word durations," In *Proceedings of the International Conference on Spoken Language Processing*, 2000, pp. 601-604.

Gu, H.Y., C.Y. Tseng and L.S. Lee, "Isolated-Utterance Speech Recognition Using Hidden Markov Models with bounded State Duration," *IEEE Trans. Signal Processing*, 39(8), pp. 1743-1751, Aug. 1991.

Huang, X. D., A. Acero and H. W. Hon, Spoken language processing: A Guide to Theory, Algorithm and system development. Carnegie Mellon University, 2001.

Kwon, O.W. and C. K. Un, "Performance of connected digit recognizers with context-dependent word duration modeling," In *Proc. APCCAS*, 1996, pp. 243-246.

Lee, C. H., and L. Rabiner, "A frame-synchronous network search algorithm for connected word recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, 37, pp. 1649-1658, Nov. 1989.

Lee, T., W.K. Lo and P.C. Ching, "Development of Cantonese spoken language corpora for speech applications," In *Proceedings of the International Symposium on Chinese Spoken Language Processing,* 1998, pp. 102-107.

Lee, T., R. Carlson and B. Granström, "Context-dependent duration modeling for continuous speech recognition," In *Proceedings of the International Conference on Spoken Language Processing,* 1998, pp.2955-2958.

Levinson, S.E., "Continuously Variable Duration Hidden Markov Models for Speech Analysis," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1986, 2, pp. 1241-1244.

Ney, H., "The use of a one-stage dynamic programming algorithm for connected Word Recognition," *IEEE Trans. Acoust., Speech, Signal Processing*, vol. ASSP-32 (2), pp. 263-271, April. 1984.

Power, K., "Duration modeling for improved connected digit recognition," In *Proceedings of the International Conference on Spoken Language Processing*, 1996, pp. 885-888.

Rabiner, L.R. "A tutorial on Hidden Markov Models and selected applications in speech recognition," In *Proceedings of the IEEE*, 77, pp. 257-286, Feb. 1989.

Russell, M. and R. Moore, "Explicit modeling of state occupancy in hidden Markov models for automatic speech recognition," In *Proceedings of the International Conference on Acoustics, Speech and Signal Processing*, 1985, pp. 2376-2379.

Yang, C., "On the robustness of static and dynamic spectral information for speech recognition in noise," PhD dissertation, The Chinese University of Hong Kong, 2004.

Zhu, Y., "Using Duration Information in HMM-based Automatic Speech Recognition" MPhil Thesis, The Chinese University of Hong Kong, 2005.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 1, March 2006, pp. 17-36

Modeling Cantonese Pronunciation Variations for Large-Vocabulary Continuous Speech Recognition

Tan Lee*, Patgi Kam* and Frank K. Soong*+

Abstract

This paper presents different methods of handling pronunciation variations in Cantonese large-vocabulary continuous speech recognition. In an LVCSR system, three knowledge sources are involved: a pronunciation lexicon, acoustic models and language models. In addition, a decoding algorithm is used to search for the most likely word sequence. Pronunciation variation can be handled by explicitly modifying the knowledge sources or improving the decoding method. Two types of pronunciation variations are defined, namely, phone changes and sound changes. Phone change means that one phoneme is realized as another phoneme. A sound change happens when the acoustic realization is ambiguous between two phonemes. Phone changes are handled by constructing a pronunciation variation dictionary to include alternative pronunciations at the lexical level or dynamically expanding the search space to include those pronunciation variants. Sound changes are handled by adjusting the acoustic models through sharing or adaptation of the Gaussian mixture components. Experimental results show that the use of a pronunciation variation dictionary and the method of dynamic search space expansion can improve speech recognition performance substantially. The methods of acoustic model refinement were found to be relatively less effective in our experiments.

Keywords: Automatic Speech Recognition, Pronunciation Variation, Cantonese

1. Introduction

Given a speech input, automatic speech recognition (ASR) is a process of generating possible hypotheses for the underlying word sequence. This can be done by establishing a mapping between the acoustic features and the yet to be determined linguistic representations. Given

* Department of Electronic Engineering, The Chinese University of Hong Kong, Shatin, N. T., Hong Kong Tel: 852-26098267 Fax: 852-26035558
E-mail: tanlee@ee.cuhk.edu.hk
The author for correspondence is Tan Lee.
+ Micrpsoft Research Asia, 5th Floor, Sigma Center, 49 Zhichun Road, Haidian, Beijing 100080, China

[Received June 16, 2005; Revised January 20, 2006; Accepted January 23, 2006]

the high variability of human speech, such mapping is in general not one-to-one. Different linguistic symbols can give rise to similar speech sounds, while the same linguistic symbol may also be realized in different pronunciations. The variability is due to co-articulation, regional accents, speaking rate, speaking style, etc. Pronunciation modeling is aimed at providing an effective mechanism by which ASR systems can be adapted to pronunciation variability.

Pronunciation variations can be divided into two types: phone change and sound change [Kam 2003] [Liu and Fung 2003]. In [Saraçlar and Khudanpur 2000] [Liu 2002], they are also referred to as complete change and partial change, respectively. A phone change happens when a *baseform* (canonical) phoneme is realized as another phoneme, which is referred to as its *surface-form*. The baseform pronunciation is considered to be the "standard" pronunciation that the speaker is supposed to use. Surface-form pronunciations are the actual pronunciations that different speakers may use. A sound change can be described as variation in phonetic properties, such as nasalization, centralization, voicing, etc. Acoustically, the variant sound is considered to be neither the baseform nor any surface-form phoneme. In other words, we cannot find an appropriate unit in the language's phoneme inventory to represent the sound. In terms of the scope of such variations, pronunciation variations can be divided into word-internal and cross-word variations [Strik and Cucchiarini 1999].

There have been many studies on modeling pronunciation variations for improving ASR performance. They are focused mainly on two problems: 1) prediction of the pronunciation variants, and 2) effective use of pronunciation variation information in the recognition process [Strik and Cucchiarini 1999]. Knowledge-based approaches use findings from linguistic studies, existing pronunciation dictionaries, and phonological rules to predict the pronunciation variations that could be encountered in ASR [Aubert and Dugast 1995] [Kessens *et al*. 1999]. Data-driven approaches attempt to discover the pronunciation variants and the underlying rules from acoustic signals. This is done by performing automatic phone recognition and aligning the recognized phone sequences with reference transcriptions to find out the surface forms [Saraçlar *et al*. 2000] [Wester 2003]. Some studies used hand-labelled corpora [Riley *et al*. 1999].

The key components of a large-vocabulary continuous speech recognition system are the acoustic models, the pronunciation lexicon and the language models [Huang *et al.* 2001]. The acoustic models are a set of hidden Markov models (HMM) that characterize the statistical variations of input speech. Each HMM represents a specific sub-word unit, e.g. a phoneme. The pronunciation lexicon and the language models are used to define and constrain the ways sub-word units can be concatenated to form words and sentences. They are used to define a search space from which the most likely word string(s) can be determined with a computationally efficient decoding algorithm. Within such a framework, pronunciation

variations can be handled by modifying one or more of the knowledge sources or improving the decoding algorithm. Phone changes can be handled by replacing the baseform transcription with surface-form transcriptions, i.e. the actual pronunciations observed. In an LVCSR system, this can be done by either augmenting the baseform lexicon with the additional pronunciation variants [Kessens *et al.* 1999] [Liu *et al*. 2000] [Byrne *et al*. 2001], or expanding the search space during the decoding process to include those variants [Kam and Lee 2002]. In order to deal with sound changes, pronunciation modeling must be applied at a lower level, for example, on the individual states of a hidden Markov model (HMM) [Saraçlar *et al*. 2000]. In general, acoustic models are trained solely with baseform transcriptions. It is assumed that all training utterances follow exactly the canonical pronunciations. This convenient, but apparently unrealistic, assumption renders the acoustic models inadequate in representing the variations of speech sounds. To alleviate this problem, various methods of acoustic model refinement were proposed [Saraçlar *et al*. 2000] [Venkataramani and Byrne 2001] [Liu 2002].

In this paper, the pronunciation variations in continuous Cantonese speech are studied. The linguistic and acoustic properties of spoken Cantonese are considered in the analysis of pronunciation variations and, subsequently, the design of pronunciation modeling techniques for LVCSR. Like in most conventional approaches, phone changes are anticipated by using an augmented pronunciation lexicon. The lexicon includes the most frequently occurring alternative pronunciations that are derived from training data. We also describe a novel method of dynamically expanding the search space during decoding to include pronunciation variants that are predicted with context-dependent pronunciation models. For sound changes, we propose to measure the similarities between confused baseform and surface-form models at the Gaussian mixture component level and, accordingly, refine the models through sharing and adaptation of the relevant mixture components.

In the next section, the properties of spoken Cantonese are described and the fundamentals of Cantonese LVCSR are explained. In Section 3, different methods of modeling pronunciation variations at the lexical level are presented in detail and experimental results are given. The techniques for handling sound changes through acoustic model refinement are described in Section 4. Conclusions are given in Section 5.

2. Cantonese LVCSR

2.1 About Cantonese

Cantonese is one of the major Chinese dialects. It is the mother tongue of over 60 million people in Southern China and Hong Kong [Grimes *et al.* 2000]. The basic unit of written Cantonese is a Chinese character [Chao 1965]. Chinese characters are ideographic, meaning that they contain no information about pronunciation. There are more than ten thousand

distinctive characters. In Cantonese, each of them is pronounced as a single syllable that carries a specific tone. A sentence is spoken as a string of monosyllabic sounds. A character may have multiple pronunciations, and a syllable typically corresponds to a number of different characters.

A Cantonese syllable is formed by concatenating two types of phonological units: the *Initial* and the *Final*, as shown in Figure 1 [Hashimoto 1972]. There are 20 Initials (including the null Initial) and 53 Finals in Cantonese, in contrast to 23 Initials and 37 Finals in Mandarin. Table 1 and Table 2 list the Initials and Finals of Cantonese. They are labeled using *Jyut Ping*, a phonemic transcription scheme proposed by the Linguistic Society of Hong Kong [LSHK 1997]. In terms of the manner of articulation, the 20 Initials can be categorized into seven classes: null, plosive, affricate, fricative, glide, liquid, and nasal. The 53 Finals can be divided into five categories: vowel (long), diphthong, vowel with nasal coda, vowel with stop coda, and syllabic nasal. Except for [m] and [ng], each Final contains at least one vowel element. The stop codas, i.e., -p, -t and -k, are unreleased. In Cantonese, there are more than 600 legitimate Initial-Final combinations, which are referred to as *base syllables*.

BASE SYLLABLE		
Initial	*Final*	
[Onset]	Nucleus	[Coda]

Figure 1. The composition of a Cantonese syllable.* [] *means optional.

Table 1.The Cantonese Initials

Jyut Ping symbols	Manner of Articulation	Place of Articulation
[b]	Plosive, unaspirated	Labial
[d]	Plosive, unaspirated	Alveolar
[g]	Plosive, unaspirated	Velar
[p]	Plosive, aspirated	Labial
[t]	Plosive, aspirated	Alveolar
[k]	Plosive, aspirated	Velar
[gw]	Plosive, unaspirated, lip-rounded	Velar, labial
[kw]	Plosive, aspirated, lip-rounded	Velar, labial
[z]	Affricate, unaspirated	Alveolar
[c]	Affricate, aspirated	Alveolar
[s]	Fricative	Alveolar
[f]	Fricative	Dental-labial
[h]	Fricative	Vocal
[j]	Glide	Alveolar
[w]	Glide	Labial
[l]	Liquid	Lateral
[m]	Nasal	Labial
[n]	Nasal	Alveolar
[ng]	Nasal	Velar

Table 2.The 53 Cantonese Finals

		CODA								
		Nil	-i	-u	-p	-t	-k	-m	-n	-ng
NUCLEUS	-aa-	[aa]	[aai]	[aau]	[aap]	[aat]	[aak]	[aam]	[aan]	[aang]
	-a-		[ai]	[au]	[ap]	[at]	[ak]	[am]	[an]	[ang]
	-e-	[e]	[ei]				[ek]			[eng]
	-i-	[i]		[iu]	[ip]	[it]	[ik]	[im]	[in]	[ing]
	-o-	[o]	[oi]	[ou]		[ot]	[ok]		[on]	[ong]
	-u-	[u]	[ui]			[ut]	[uk]		[un]	[ung]
	-yu-	[yu]				[yut]			[yun]	
	-oe-	[oe]	[eoi]			[eot]	[oek]		[eon]	[oeng]
								[m]		[ng]

From phonological points of view, Cantonese has nine tones that are featured by differently stylized pitch patterns. They are divided into two categories: entering tones and non-entering tones. The entering tones occur exclusively with syllables ending in a stop coda (-p, -t, or –k). They are contrastively shorter in duration than the non-entering tones. In terms of pitch level, each entering tone coincides roughly with a non-entering counterpart. In many transcription schemes, only six distinctive tone categories are defined. They are labeled as Tone 1 to Tone 6 in the *Jyu Ping* system. If tonal difference is considered, the total number of distinctive *tonal syllables* is about 1,800.

Table 3 gives an example of a Chinese word and its spoken form in Cantonese. The word 我們 (meaning “we”) is pronounced as two syllables. The first syllable is formed from the Initial [ng] and the Final [o], with Tone 5. The second syllable is formed from the Initial [m] and the Final [un], with Tone 4.

Table 3.An example Chinese word and its Cantonese pronunciations

Word	Chinese characters	Base syllables	Initial & Final	Tone
我們	我	ngo	[ng] [o]	5
	們	mun	[m] [un]	4

2.2 Linguistic Studies on Pronunciation Variations in Cantonese

Over the past twenty years, there have been sociolinguistic studies on how phonetic variations in Cantonese are related with social characteristics of speakers such as sex, age, and educational background. They have revealed some systematic patterns underlying the phonetic variations [Bauer and Benedict 1997] [Bourgerie 1990] [Ho 1994]. Table 4 gives a summary of the major observations in these studies.

Table 4. Major phonetic variations in Cantonese observed by sociolinguistic studies

Initial consonants	[n] ~ [l]	Inter-change between nasal and lateral Initials
	[ng]~ null	Inter-change between velar nasal and null Initial.
	[gw] → [g]	Change from labialized velar to delabialized velar before back-round vowel [o]
Syllabic nasal	[ng] → [m]	Change from velar nasal to bilabial nasal
Final consonants	-ng → -n	Change from velar nasal coda to dental nasal coda
	-k ~ -t -k ~ -p	Inter-change between velar stop coda and dental or glottal stop coda

It was found that [n]→[l], [ng]→null, and [gw]→[g] correlate with the sex and age of a speaker [Bourgerie 1990]. Older people make these substitutions much less frequently than younger generations. Female speakers tend to substitute [n] with [l], and delete [ng] more frequently than males. A correlation with the formality of the speech situation was also observed [Bourgerie 1990]. In casual speech, [l], null Initial, and [g] occur more frequently. According to [Bauer and Benedict 1997], the variations are also related to the development of neighboring dialects in the Pearl River Delta.

When the preceding syllable ends with a nasal coda, there is a tendency to substitute the Initial [l] of the succeeding syllable with [n] [Ho 1994]. Labial dissimilation is probably the cause of the change [gw]→[g], when the right context is -o, for example "gwok" 國 (country), pronounced as "gok" 角 (corner). The sequence of the two lip-rounded segments -w- and -o- become redundant or unnecessary with the second one driving out the first. The change [ng]→[m] is due to the fact that when [ng] occurs in the presence of a bilabial coda, its place of articulation changes to bilabial. For example, "sap ng" 十五 (fifteen) becomes "sap m" through the perseverance of the bilabial closure of the coda -p into the articulation of the following syllabic nasal. This is referred to as perseveratory assimilation [Bauer and Benedict 1997].

Other pronunciation variations are due to the dialectal accents of non-native speakers, who may have difficulties mastering some of the Cantonese pronunciations. They sometimes use the pronunciation of their mother tongue to pronounce a Cantonese word, for example, "ngo" 我 (me) is pronounced as "wo" by a Mandarin speaker.

2.3 Cantonese LVCSR: the Baseline System

Figure 2 gives the functional block diagram of a typical LVCSR system. At the front-end processing module, the input speech is analyzed and converted into a sequence of acoustic feature vectors, denoted by O. The goal of speech recognition is to determine the most probable word sequence W, given the observation O. With the Bayes' formula, the decision

can be made as

$$W^* = \arg\max_W P(W \mid O) = \arg\max_W P(O \mid W)P(W) . \quad (1)$$

Usually the acoustic models are built at the sub-word level. Let B be the sub-word sequence that represents W . Eq. (1) can be written as

$$W^* = \arg\max_W P(O \mid B)P(B \mid W)P(W) , \quad (2)$$

where $P(O \mid B)$ and $P(W)$ are referred to as the (sub-word level) acoustic models and the language models, respectively. $P(B \mid W)$ is given by a pronunciation lexicon.

In the case of Chinese speech recognition, the sub-word units can be either syllables, Initials and Finals, or phone-like units. The recognition output is typically represented as a sequence of Chinese characters. The details of our baseline system for Cantonese LVCSR are given below.

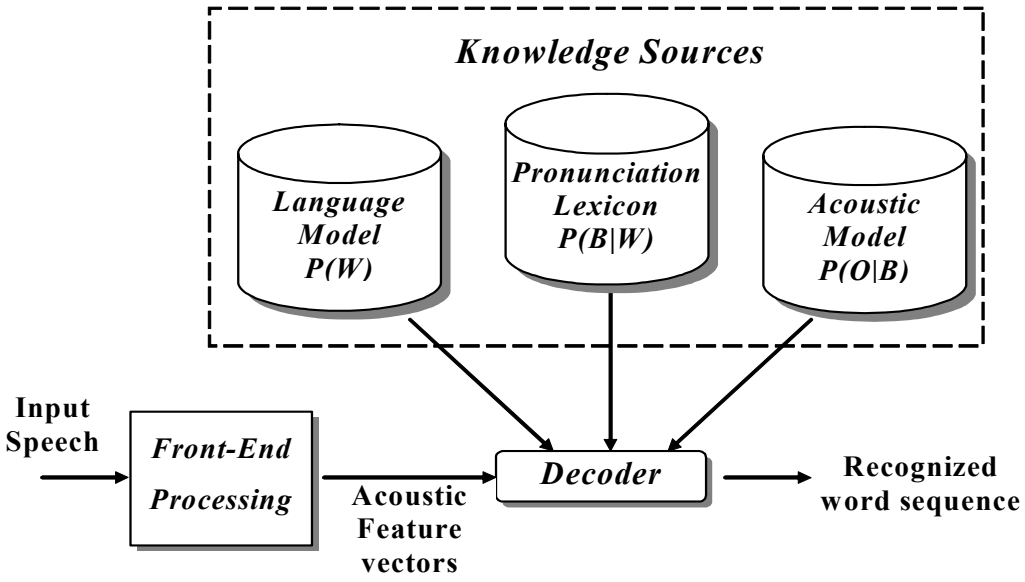


Figure 2. A typical LVCSR system

Front-end processing

Acoustic feature vectors are computed every 10 msec. Each feature vector is composed of 39 elements, which includes 12 Mel-frequency cepstral coefficients, log energy, and their first-order and second-order derivatives. The analysis window size is 25 msec.

Acoustic models

The acoustic models are right-context-dependent cross-word Initials and Finals models [Wong 2000]. The number of HMM states for Initial and Final units are 3 and 5, respectively. Each state is represented by a mixture of 16 Gaussian components. The decision tree based state clustering approach is used to allow the sharing of parameters among models.

Pronunciation lexicons and language models

The lexicon contains about 6,500 entries, among which 60% are multi-character words and the others are single-character words [Wong 2000]. These words were selected from a newspaper text corpus of 98 million Chinese characters. The out-of-vocabulary percentage is about 1% [Wong 2000]. For each word entry, the canonical pronunciation(s) is specified in the form of Initials and Finals [CUPDICT 2003]. The language models are word bi-grams that were trained with the same text corpus described above.

Decoder

The search space is formed from lexical trees that are derived from the pronunciation lexicon. One-pass Viterbi search is used to determine the most probable word sequence [Choi 2001]. The acoustic models were trained using CUSENT, which is a read speech corpus of continuous Cantonese sentences collected at the Chinese University of Hong Kong [Lee *et al.* 2002]. There are over 20,000 gender-balanced training utterances. The test data in CUSENT consists of 1,200 utterances from 6 male and 6 female speakers. The performance of the LVCSR system is measured in terms of word error rate (WER) for the 1,200 test utterances. The baseline WER is 25.34%.

3. Handling Phone Change with Pronunciation Models

The pronunciation lexicon used in the baseline system provides only the baseform pronunciation for each of the word entries. In real speech, the baseform pronunciations are realized differently, depending on the speakers, speaking styles, etc. Phone change means that the pronunciation variation can be considered as one or more Initial or Final (IF) unit in the baseform pronunciation being substituted by another IF unit. Note that the substituting surface-form unit is also one of the legitimate IF units, as listed in Tables 1 and 2.

A pronunciation model (PM) is a descriptive and predictive model by which the surface-form pronunciation(s) can be derived from the baseform one. There have been three different types of models proposed by previous studies. They are: 1) phonological rules for generating pronunciation variations [Wester 2003] [Kessens *et al*. 2003], 2) a pronunciation variation dictionary (PVD) that explicitly lists alternative pronunciations [Aubert and Dugast 1995] [Kessens *et al*. 1999] [Liu *et al*. 2000], and 3) statistical decision trees that predict pronunciation variations according to phonetic context [Riley *et al*. 1999] [Fosler-Lussier

1999] [Saraçlar *et al*. 2000]. In this study, two different approaches to handling phone changes in Cantonse ASR are formulated and evaluated. The first approach uses a probabilistic PVD to augment the baseform lexicon. This is a straightforward and commonly used method that has been proven effective for various tasks and languages [Strik and Cucchiarini 1999]. In the second approach, pronunciation variation information is introduced during the decoding process. Decision tree based PMs are used to dynamically expand the search space. In [Saraçlar *et al*. 2000], a similar idea was presented. Decision tree based PMs were applied to a word lattice to construct a recognition network that includes surface-form realizations.

3.1 Use of a Pronunciation Variation Dictionary (PVD)

In this study, the information about Cantonese pronunciation variations is obtained through the data-driven approach. This is done by aligning the baseform transcriptions with the recognized surface-form IF sequences for all training utterances. For each training utterance, the surface-form IF sequence is obtained through phoneme recognition with the acoustic models as described in Section 2.3. To reflect the syllable structure of Cantonese, the recognition output is constrained to be a sequence of Initial-Final pairs. With this approach, only substitutions at the IF level are considered pronunciation variations. Partial change of an IF unit and the deletion of an entire Initial or Final are not reflected in the surface-form IF sequences.

The surface-form phoneme sequence is then aligned with the baseform transcription. This gives a phoneme accuracy of 90.33%. The recognition errors are due, at least partially, to phoneme-level pronunciation variation. For a particular baseform phoneme *b* and a surface-form phoneme *s*, the probability of *b* being pronounced as *s* is computed based on the number of times that *b* is recognized as *s*. This probability is referred to as the variation probability (VP). As a result, each pair of IF units is described with a probability of being confused. This is also referred to as a confusion matrix [Liu *et al*. 2000]. It is assumed that systematic phone change can be detected by a relatively high VP, while a low VP is more likely due to recognition errors. A VP threshold is used to prune those less frequent surface-form pronunciations. As a result, for each baseform IF unit, we can find a certain number of surface-form units, each with a pre-computed VP.

A straightforward way of handling pronunciation variation is to augment the basic pronunciation lexicon with alternative pronunciations [Strik and Cucchiarini 1999]. Such an augmented lexicon is named a pronunciation variation dictionary (PVD). In the PVD, each word can have multiple pronunciations, each being assigned a word-level variation probability (VP). The PVD can be obtained from the IF confusion matrix. The word-level VP is given by multiplying the phone-level VPs of all the individual phonemes in the surface-form pronunciation. With the use of the PVD, the goal of speech recognition is essentially to search

for the most probable word sequence by considering all possible surface-form realizations. This can be conceptually illustrated by modifying Eq. (2) as

$$W^* = \arg\max_{W,k} P(O \mid S_{W,k}) P(S_{W,k} \mid W) P(W), \tag{3}$$

where $S_{W,k}$ denotes one of the surface-forms realizations of W. $P(S_{W,k} \mid W)$ are obtained from the word-level VPs.

3.2 Prediction of Pronunciation Variation during Decoding

The PVD includes both context-independent and context-dependent phone changes. Since each word is treated individually, the phonetic context being considered is limited to within the word. To deal with cross-word context-dependent phone changes, we propose applying pronunciation models at the decoding level. Our baseline system uses a one-pass search algorithm [Choi 2001]. The search space is structured as lexical trees. Each node on a tree corresponds to a baseform IF unit. The search is token based. Each token represents a path that reaches a particular lexical node. The propagation of tokens follows the lexical trees, which cover only the legitimate phoneme sequences as specified by the pronunciation lexicon. The search algorithm can be modified in a way that the number of alive tokens is increased to account for pronunciation variations. When a path extends from a particular IF node, its destination node can be either the legitimate node (baseform pronunciation) or any of the predicted surface-form nodes. In other words, the search space is dynamically expanded during the search process.

In this approach, a context-dependent pronunciation model is needed to predict the surface-form phoneme given the baseform phoneme and its context. It is implemented using the decision tree clustering technique, following the approaches described in [Riley *et al*. 1999] [Fosler-Lussier 1999]. Each baseform phoneme is described using a decision tree. Given a baseform phoneme, as well as its left context (the right context is not available in a forward Viterbi search), the respective decision-tree pronunciation model (DTPM) gives all possible surface-form realizations and their corresponding VPs [Kam and Lee 2002].

Like the confusion matrix, the DTPM is trained with the phoneme recognition outputs for the CUSENT training utterances. The training involves an optimization process by which the surface-form phonemes are clustered based on phonetic context. At a particular node of the tree, a set of "yes/no" questions about the phonetic context are evaluated. Each question leads to a different partition of the training data. The question that minimizes the overall conditional entropy of the surface-form realizations is selected for that node. The node-splitting process stops when there are too few training data [Kam 2003].

3.3 Experimental Results and Discussion

Table 5 gives the recognition results with the use of PVDs that are constructed with different values of the VP threshold. The baseline system uses the basic pronunciation lexicon that contains 6,451 words. The size of the PVD increases as the VP threshold decreases. It is obvious that the introduction of pronunciation variants improves recognition performance. The best performance is attained with a VP threshold of 0.05. In this case, the PVD contains 8,568 pronunciations for the 6,451 words, i.e. 1.33 pronunciation variants per word. With a very small value for the VP threshold, e.g. 0.02, the recognition performance is not good because there are too many pronunciation variants being included and some of them do not really represent pronunciation variation.

Table 5. Recognition results of using a PVD with different VP thresholds

	Baseline	VP threshold				
		0.02	**0.05**	0.10	0.15	0.20
Word error rate (%)	25.34	23.91	**23.49**	23.70	23.64	23.58
No. of word entries in the PVD	6,451	20,840	**8,568**	7,356	7,210	7,171

Table 6 shows the recognition results attained by using the DTPM for dynamic search space expansion. It appears that this approach is as effective as the PVD. Unlike the results for the PVD, the performance with a VP threshold of 0.2 is better than that with a threshold of 0.05. This means that the predictions made by the DTPM should be pruned more stringently than the IF confusion matrix. Because of its context-dependent nature, the DTPM has relatively less training data, and the variation probabilities cannot be reliably estimated. It is preferable not to include those unreliably predicted pronunciation variants.

Table 6. Recognition results by dynamic search space expansion

	Baseline	VP threshold	
		0.05	**0.2**
Word error rate (%)	25.34	23.53	**23.27**

By analyzing the recognition results in detail, it is observed that many errors are corrected by allowing the following pronunciation variations:

Initials: [gw]→[g], [n]→[l], [ng]→null

Finals: [ang]→[an], [ng]→[m] (syllabic nasal)

These observations match well with the findings in sociolinguistic studies on Cantonese phonology (Section 2.2).

4. Handling Sound Change by Acoustic Model Refinement

Unlike phone changes, a sound change cannot be described as a simple substitution of one phoneme for another. It is regarded as a partial change from the baseform phoneme to a surface-form phoneme [Liu and Fung 2003]. Our approaches presented below attempt to refine the acoustic models to handle the acoustic variation caused by sound changes. The acoustic models are continuous-density HMMs. The output probability density function (pdf) at each HMM state is a mixture of Gaussian distributions. The use of multiple mixture components is intended to describe complex acoustic variabilities. The acoustic models trained only according to the baseform pronunciations are referred to as baseform models. Each baseform phoneme may have different surface-form realizations. The acoustic models representing these surface-form phonemes are referred to as surface-form models. A baseform model doesnot reflect the acoustic properties of the relevant surface-form phonemes. One way of dealing with this deficiency is through the sharing of Gaussian mixture components among the baseform and surface-form models. In [Saraçlar *et al.* 2000], a state-level pronunciation model (SLPM) was proposed. It allows the HMM states of a baseform model to share the output densities of its surface-form phonemes. A state-to-state alignment was obtained from decision-tree PMs, and the most frequently confused state pairs were involved in parameter sharing. In [Liu and Fung 2004], the method of phonetic mixtures tying was applied to deal with sound changes. A set of so-called extended phone units were derived from acoustic training data to describe the most prominent phonetic confusion. These units were then modeled by mixture tying with the baseform models. In this study, we investigate both the sharing and adaptation of the acoustic model parameters at the mixture level [Kam *et al.* 2003].

4.1 Sharing of Mixture Components

First of all, the states of the baseform and surface-form models are aligned. It is assumed that both models have the same number of states. Then, state j of the baseform model is aligned with state j of the surface-form model. Consider a baseform phoneme B. The output pdf at state j is given as

$$b_j(o_t) = \sum_{m=1}^{M} w_{jm} N(o_t; \mu_{jm}, \Sigma_{jm}), \tag{4}$$

where M is the number of Gaussian mixture components, and w_{jm} is the weight for the mth mixture component. The baseform output pdf can be modified to include the contributions from the surface-form states

$$b_j'(o_t) = VP(B,B)\cdot b_j(o_t) + \sum_{\substack{n=1 \\ S_n \neq B}}^{N} VP(S_n,B)\cdot q_{S_n,j}(o_t), \tag{5}$$

where S_n denotes the *n*th surface-form of B, N is the total number of surface-forms, $VP(S_n,B)$ is the variation probability of S_n with respect to baseform B, and $q_{S_n,j}(o_t)$ denotes the output pdf of state j of the *n*th surface-form model.

The number of mixture components in the resultant baseform model depends on N. More surface-form pronunciations bring in more mixture components to the modified baseform state. As the number of mixture components is changed, re-estimation of mixture weights is required.

4.2 Adaptation of Mixture Components

Although sharing mixture components yields an acoustically richer model, it also greatly increases the model size for which more memory space and higher computation complexities are required. Moreover, if the baseform and surface-form mixture components are very similar, including them all in the modified baseform is unnecessarily superfluous.

We propose to refine the baseform acoustic models through parameters adaptation. The total number of model parameters remains unchanged. Like in the approach of mixture sharing, the states of the baseform and surface-form models are aligned. The surface-forms are generated from the IF confusion matrix. Consider the aligned states of the baseform phoneme B and one of its surface-forms S. Let $m_B(i)$ and $m_S(j)$ denote the *i*th mixture component in the baseform state and the *j*th mixture component in the surface-form state, respectively, where $i, j = 1,2,\cdots,M$. The distances between all pairs ($m_B(i)$, $m_S(j)$) are computed. Then each surface-form component is paired up with the nearest baseform component. That is, for each $m_S(j)$, we find

$$\hat{i} = \arg\min_{m_B(i)} d(m_B(i), m_S(j)). \tag{6}$$

The "distance" between two Gaussian distributions is calculated using the Kullback-Leibler divergence (KLD) [Myrvoll and Soong 2003]. Given two multivariate Gaussian distributions f and g, the symmetric KLD has the following closed form

$$d(f,g) = \frac{1}{2} trace\{(\Sigma_f^{-1} + \Sigma_g^{-1})(\mu_f - \mu_g)(\mu_f - \mu_g)^T + \Sigma_f \Sigma_g^{-1} + \Sigma_g \Sigma_f^{-1} - 2\mathrm{I}\}, \tag{7}$$

where μ and Σ denote the mean vectors and the covariance matrices of the two distributions, respectively, and I is the identity matrix.

As a result, for this pair of baseform and surface-form states, each Gaussian component $m_B(i)$ is associated with k surface-form components, as illustrated in Figure 3. The centroid of these k components is computed. If the baseform B has n surface forms, there will be n such centroids. These surface-form centroids and the corresponding baseform component are weighted with the VP, and together produce a new centroid that is taken as the adapted baseform component. In this way, the adapted model is expected to shift towards the surface-form phonemes. The extent of such a shift depends on the VP. The mean and covariance of the centroid of k weighted Gaussian components can be found by minimizing the following weighted divergence

$$\{\mu_c', \Sigma_c'\} = \arg\min_{\mu_c, \Sigma_c} \sum_{n=1}^{k} a_n d(f_c, f_n), \tag{8}$$

where f_n denotes the nth component and a_n is the respective weighting coefficient. Assuming diagonal covariances, the weighted centroid is given as [Myrvoll and Soong 2003]

$$\begin{aligned} \mu_c'(i) &= \frac{\Sigma_{n=1}^{k} a_n (\Sigma_c^{-1}(i) + \Sigma_n^{-1}(i)) \mu_n(i)}{\Sigma_{n=1}^{k} a_n (\Sigma_c^{-1}(i) + \Sigma_n^{-1}(i))} \\ \Sigma_c'(i) &= \sqrt{\frac{\Sigma_{n=1}^{k} a_n [\Sigma_n(i) + (\mu_c(i) - \mu_n(i))^2]}{\Sigma_{n=1}^{k} a_n \Sigma_n^{-1}(i)}} \end{aligned}. \tag{9}$$

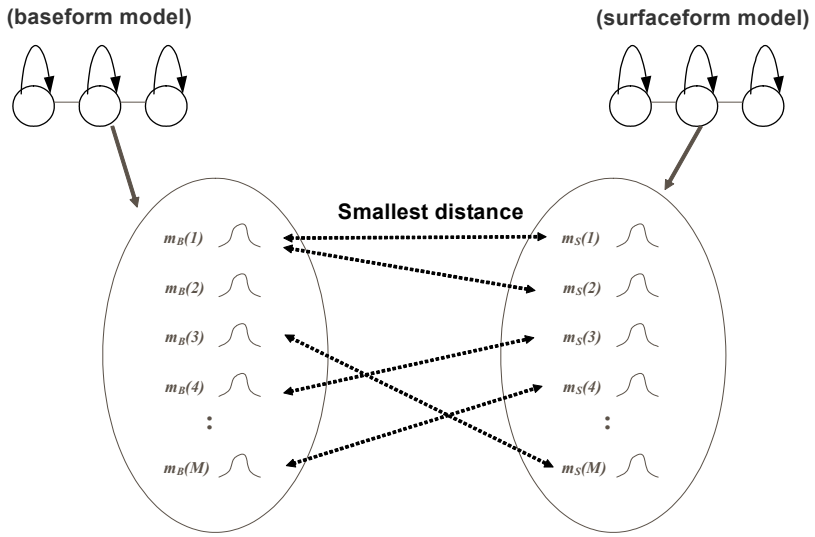


Figure 3. Mapping between baseform and surfaceform mixture components

4.3 Experimental Results and Discussion

Table 7 gives the recognition results attained with the two methods of acoustic model refinement. The VP threshold for surface-form prediction is set at 0.05. Apparently, both approaches improve recognition performance. The sharing of mixture components seems to be more effective than adaptation. However, this is at the cost of a substantial increase in model complexity. The baseline acoustic models have a total of 32,144 Gaussian components. The adaptation approach retains the same number of Gaussian components. The models obtained with the sharing approach have 37,505 components, 17% more than the baseline. If we use an equal number of components in the baseline acoustic models, the baseline word error rate will be reduced to 24.34%, and the benefit of sharing mixture components is only marginal.

Table 7. Recognition results with different methods of acoustic model refinement

	Baseline	Sharing	Adaptation
Word error rate (%)	25.34	23.96	24.70

With the adaptation approach, the baseform pdf is shifted towards the corresponding surface forms. If a surface-form pdf is far away from the baseform one, the extent of the modification will be substantial and, consequently, the modified pdf may fail to model the original baseform. On the other hand, the sharing approach has the problem of undesirably including redundant components in the baseform models. Thus we combine these two approaches. The idea is to perform adaptation using the surface-form components that are close to the baseform, and at the same time, to use those relatively distant components for sharing.

The values of the KLD between the baseform pdf and the nearest surface-form pdf have been analyzed. As illustrative examples, the histograms of the KLD at different states between [aak] (baseform) and [aa] (surface form), and between [aak] and [aat], are shown as in Figure 4. There are two main types of KLD distributions: 1) concentration around small values (e.g., states 1 and 2 of the pair "[aak]→[aa]"), and 2) a wide range of values (e.g., states 3 to 5 of the pair "[aak]→[aa]"). A small KLD means that the mixture components of the baseform and surface forms are similar. In this case, the baseform components adapt to the surface form. In the case of a widely distributed KLD, the surface-form components should not be used to adapt the baseform components, but rather should be kept along with the modified baseform model in order to explicitly characterize irregular pronunciations. In this way, a combined approach to baseform model refinement is formulated.

Despite the good intentions, the combined use of sharing and adaptation doesnot lead to favorable experimental results. With a total of 34,042 mixture components in the refined acoustic models, the word error rate is 24.57%. The baseline performance is 24.93% with the same model complexity.

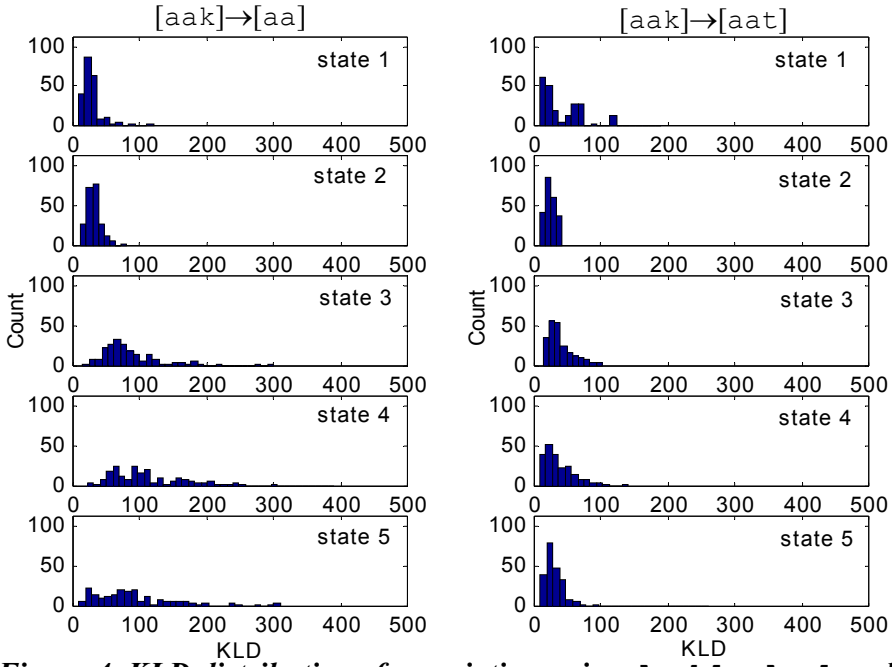


Figure 4. KLD distributions for variation pairs **[aak]→[aa]** ***and*** **[aak]→ [aat]**

5. Conclusions

In this study, we have classified pronunciation variations into phone changes and sound changes. However, these are not well defined classifications, especially for the sound changes. There is not a clear boundary that separates a phoneme substitution (phone change) from a phoneme modification (sound change). This may partially explain why the proposed techniques of handling sound change are not as effective as the methods for handling phone change.

The use of a PVD is intuitive and straightforward in implementation. It can reduce the word error rate noticeably. When constructing a PVD, the value of the VP threshold needs to be carefully determined. While a tight threshold obviously doesnot show any effect, a lax control of the PVD size leads to not only a long recognition time but also performance degradation. The method of dynamic search space expansion during decoding can bring about the same degree of performance improvement as the PVD. However, the training of context-dependent pronunciation prediction models requires a large amount of data.

The methods of acoustic model refinement donot improve recognition performance as much as we expected. Similar effect can be achieved by using more mixture components. Indeed, more mixture components can describe more complex acoustic variations, which include the variations caused by alternative pronunciations. The sharing of mixture

components is equivalent to having more mixture components right at the beginning of acoustic models training. Adaptation of mixture components is not as effective as increasing the number of mixture components.

For any of the above methods to be effective, the accurate and efficient acquisition of pronunciation variation information is most critical. Manual labeling is impractical. Automatic detection of pronunciation variations is still an open problem.

Acknowledgement

This research is partially supported by a Research Grant from the Hong Kong Research Grants Council (Ref: CUHK4206/01E).

References

Aubert, X., and C. Dugast, "Improved acoustic-phonetic modeling in Philips' dictation system by handling liaisons and multiple pronunciations," In *Proceedings of 1995 European Conference on Speech Communication and Technology*, pp.767 – 770.

Bauer, R.S., and P.K. Benedict, *Trends in Linguistics, Studies and Monographs 102, Modern Cantonese Phonology*, Mouton de Gruyter, Berlin, New York, 1997.

Bourgerie, D.S., *A Quantitative Study of Sociolinguistic Variation in Cantonese*, PhD Thesis, The Ohio State University, 1990.

Byrne, W., V. Venkataramani, T. Kamm, T.F. Zheng, Z. Song, P. Fung, Y. Liu and U. Ruhi, "Automatic generation of pronunciation lexicons for Mandarin spontaneous speech," In *Proceedings of the 2001 International Conference on Acoustics, Speech and Signal Processing*, l.1, pp.569 – 572.

Chao, Y.R., *A Grammar of Spoken Chinese*, University of California Press, 1965.

Choi, W.N., *An Efficient Decoding Method for Continuous Speech Recognition Based on a Tree-Structured Lexicon*, MPhil Thesis, The Chinese University of Hong Kong, 2001.

CUPDICT: Cantonese Pronunciation Dictionary (Electronic Version), Department of Electronic Engineering, The Chinese University of Hong Kong, http://dsp.ee.cuhk.edu.hk/speech/, 2003.

Fosler-Lussier, E., "Multi-level decision trees for static and dynamic pronunciation models," In *Proceedings of 1999 European Conference on Speech Communication and Technology*, pp.463 – 466.

Grimes, B.F. *et al.*, *Ethnologue, Languages of the World*, SIL International, 2000.

Hashimoto, O.-K. Y., *Studies in Yue Dialects 1: Phonology of Cantonese*, Cambridge University Press, 1972.

Ho, M.T., *(n-) and (l-) in Hong Kong Cantonese: A Sociolinguistic Case Study*, MA Thesis, University of Essex, 1994.

Huang, X., A. Acero, and H.W. Hon, *Spoken Language Processing: A Guide to Theory, Algorithm and System Development*, Prentice Hall PTR., 2001.

Kam, P., *Pronunciation Modeling for Cantonese Speech Recognition*, MPhil Thesis, The Chinese University of Hong Kong, 2003.

Kam, P., and T. Lee, "Modeling pronunciation variation for Cantonese speech recognition," In *Proceedings of ISCA ITR-Workshop on Pronunciation Modeling and Lexicon Adaptation* 2002, pp.12-17.

Kam, P., T. Lee and F. Soong, "Modeling Cantonese pronunciation variation by acoustic model refinement," In *Proceedings of 2003 European Conference on Speech Communication and Technology*, pp.1477 – 1480.

Kessens, J.M., M. Wester and H. Strik, "Improving the performance of a Dutch CSR by modeling within-word and cross-word pronunciation variation," *Speech Communication*, 29, pp.193 – 207, 1999.

Kessens, J.M., C. Cucchiarini and H. Strik, "A data driven method for modeling pronunciation variation," *Speech Communication*, 40, pp.517 – 534, 2003.

Lee, T., W.K. Lo, P.C. Ching and H. Meng, "Spoken language resources for Cantonese speech processing," *Speech Communication*, 36, No.3-4, pp.327-342, 2002

Linguistic Society of Hong Kong (LSHK), *Hong Kong Jyut Ping Characters Table (粵語拼音字表)*. Linguistic Society of Hong Kong Press (香港語言學會出版), 1997.

Liu, M., B. Xu, T. Huang, Y. Deng and C. Li, "Mandarin accent adaptation based on context-independent/context-dependent pronunciation modeling," In *Proceedings of the 2000 International Conference on Acoustics, Speech and Signal Processing*, 2, pp.1025-1028.

Liu, Y., *Pronunciation Modeling for Spontaneous Mandarin Speech Recognition*, PhD Thesis, The Hong Kong University of Science and Technology, 2002.

Liu, Y. and P. Fung, "Modeling partial pronunciation variations for spontaneous Mandarin speech recognition," *Computer Speech and Language*, 17, 2003, pp.357 – 379.

Liu, Y. and P. Fung, "State-dependent phonetic tied mixtures with pronunication modeling for spontaneous speech recognition," *IEEE Trans. Speech and Audio Processing*, 12(4), 2004, pp.351 – 364.

Myrvoll, T.A. and F. Soong, "Optimal clustering of multivariate normal distributions using divergence and its application to HMM adaptation", In *Proceedings of the 2003 International Conference on Acoustics, Speech and Signal Processing*, 1, pp.552 - 555.

Riley, M., W. Byrne, M. Finke, S. Khudanpur, A. Ljolje, J. McDonough, H. Nock, M. Saraçlar, C. Wooters and G. Zavaliagkos, "Stochastic pronunciation modeling from hand-labelled phonetic corpora," *Speech Communication*, 29, 1999, pp.209 – 224.

Saraçlar, M. and S. Khudanpur, "Pronunciation ambiguity vs. pronunciation variability in speech recognition," In *Proceedings of the 2000 International Conference on Acoustics, Speech and Signal Processing*, 3, pp.1679-1682.

Saraçlar, M., H. Nock and S. Khudanpur, "Pronunciation modeling by sharing Gaussian densities across phonetic models," *Computer Speech and Language*, 14, 2000, pp.137 – 160.

Strik, H. and C. Cucchiarini, "Modeling pronunciation variation for ASR: a survey of the literature," *Speech Communication*, 29, 1999, pp.255 – 246.

Venkataramani, V. and W. Byrne, "MLLR adaptation techniques for pronunciation modeling," In *Proceedings of IEEE Workshop on Automatic Speech Recognition and Understanding* 2001, CD-ROM.

Wester, M., "Pronunciation modeling for ASR – knowledge-based and data-derived methods," *Computer Speech and Language*, 17, 2003, pp.69 – 85.

Wong, Y.W., *Large Vocabulary Continuous Speech Recognition for Cantonese*, MPhil Thesis, The Chinese University of Hong Kong, 2000.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 1, March 2006, pp. 37-56

A Maximum Entropy Approach for Semantic Language Modeling

Chuang-Hua Chueh*, Hsin-Min Wang+ and Jen-Tzung Chien*

Abstract

The conventional *n*-gram language model exploits only the immediate context of historical words without exploring long-distance semantic information. In this paper, we present a new information source extracted from latent semantic analysis (LSA) and adopt the maximum entropy (ME) principle to integrate it into an *n*-gram language model. With the ME approach, each information source serves as a set of constraints, which should be satisfied to estimate a hybrid statistical language model with maximum randomness. For comparative study, we also carry out knowledge integration via linear interpolation (LI). In the experiments on the TDT2 Chinese corpus, we find that the ME language model that combines the features of trigram and semantic information achieves a 17.9% perplexity reduction compared to the conventional trigram language model, and it outperforms the LI language model. Furthermore, in evaluation on a Mandarin speech recognition task, the ME and LI language models reduce the character error rate by 16.9% and 8.5%, respectively, over the bigram language model.

Keywords: Language Modeling, Latent Semantic Analysis, Maximum Entropy, Speech Recognition

1. Introduction

Language modeling plays an important role in automatic speech recognition (ASR). Given a speech signal O, the most likely word sequence $\hat{W}$ is obtained by maximizing *a posteriori* probability $p(W|O)$, or, equivalently, the product of acoustic likelihood $p(O|W)$ and prior probability of word sequence $p(W)$:

$$\hat{W} = \arg\max_{W} p(W|O) = \arg\max_{W} p(O|W)\,p(W)\,. \quad (1)$$

* Department of Computer Science and Information Engineering, National Cheng Kung University, Tainan, Taiwan, R. O. C
E-mail: chchueh@chien.csie.ncku.edu.tw
+ Institute of Information Science, Academia Sinica, Taipei, Taiwan, R. O. C

[Received June 16, 2005; Revised January 2, 2006; Accepted January 3, 2006]

This prior probability corresponds to the language model that is useful in characterizing regularities in natural language. Also, this language model has been widely employed in optical character recognition, machine translation, document classification, information retrieval [Ponte and Croft 1998], and many other applications. In the literature, there were several approaches have been taken to extract different linguistic regularities in natural language. The structural language model [Chelba and Jelinek 2000] extracted the relevant syntactic regularities based on predefined grammar rules. Also, the large-span language model [Bellegarda 2000] was feasible for exploring the document-level semantic regularities. Nevertheless, the conventional *n*-gram model was effective at capturing local lexical regularities. In this paper, we focus on developing a novel latent semantic *n*-gram language model for continuous Mandarin speech recognition.

When considering an *n*-gram model, the probability of a word sequence W is written as a product of probabilities of individual words conditioned on their preceding *n*-1 words

$$p(W) = p(w_1, w_2, \cdots, w_T) \cong \prod_{i=1}^{T} p(w_i \left| w_{i-n+1}, ..., w_{i-1}) = \prod_{i=1}^{T} p(w_i \right| w_{i-n+1}^{i-1}), \tag{2}$$

where w_{i-n+1}^{i-1} represents historical words for word w_i, and the *n*-gram parameter $p(w_i \left| w_{i-n+1}^{i-1}\right.)$ is usually obtained via the maximum likelihood estimation:

$$p(w_i \left| w_{i-n+1}^{i-1}\right.) = \frac{c(w_{i-n+1}^{i})}{c(w_{i-n+1}^{i-1})}. \tag{3}$$

Here, $c(w_{i-n+1}^{i})$ is the number of occurrences of word sequence w_{i-n+1}^{i} in the training data. Since the *n*-gram language model is limited by the span of window size *n*, it is difficult to characterize long-distance semantic information in *n*-gram probabilities. To deal with the issue of insufficient long-distance word dependencies, several methods have been developed by incorporating semantic or syntactic regularities in order to achieve long-distance language modeling.

One simple combination approach is performed using the *linear interpolation* of different information sources. With this approach, each information source is characterized by a separate model. Various information sources are combined using weighted averaging, which minimizes overall perplexity without considering the strengths and weaknesses of the sources in particular contexts. In other words, the weights were optimized globally instead of locally. The hybrid model obtained in this way cannot guarantee the optimal use of different information sources [Rosenfeld 1996]. Another important approach is based on Jaynes' maximum entropy (ME) principle [Jaynes 1957]. This approach includes a procedure for setting up probability distributions on the basis of partial knowledge. Different from linear interpolation, this approach determines probability models with the largest randomness and

simultaneously captures all information provided by various knowledge sources. The ME framework was first applied to language modeling in [Della Pietra *et al.* 1992]. In the following, we survey several language model algorithms where the idea of information combination is adopted.

In [Kuhn and de Mori 1992], the cache language model was proposed to merge domain information by boosting the probabilities of words in the previously-observed history. In [Zhou and Lua 1999], *n*-gram models were integrated with the mutual information (MI) of trigger words. The MI-Trigram model achieved a significant reduction in perplexity. In [Rosenfeld 1996], the information source provided by trigger pairs was incorporated into an *n*-gram model under the ME framework. Long-distance information was successfully applied in language modeling. This new model achieved a 27% reduction in perplexity and a 10% reduction in the word error rate. Although trigger pairs are feasible for characterizing long-distance word associations, this approach only considers the frequently co-occurring word pairs in the training data. Some important semantic information with low frequency of occurrence is lost. To compensate for this weakness, the information of entire historical contexts should be discovered. Since the words used in different topics are inherently different in probability distribution, topic-dependent language models have been developed accordingly. In [Clarkson and Robinson 1997], the topic language model was built based on a mixture model framework, where topic labels were assigned. Wu and Khudanpur [2002] proposed an ME model by integrating *n*-gram, syntactic and topic information. Topic information was extracted from unsupervised clustering in the original document space. A word error rate reduction of 3.3% was obtained using the combined language model. In [Florian and Yarowsky 1999], a delicate tree framework was developed to represent the topic structure in text articles. Different levels of information were integrated by performing linear interpolation hierarchically. In this paper, we propose a new semantic information source using latent semantic analysis (LSA) [Deerwester *et al*. 1990; Berry *et al*. 1995], which is used for reducing the disambiguity caused by polysemy and synonymy [Deerwester *et al*. 1990]. Also, the relations of semantic topics and target words are incorporated with *n*-gram models under the ME framework. We illustrate the performance of the new ME model by investigating perplexity in language modeling and the character-error rate in continuous Mandarin speech recognition. The paper is organized as follows. In the next section, we introduce an overview of the ME principle and its relations to other methods. In Section 3, the integration of semantic information and n-gram model via linear interpolation and maximum entropy is presented. Section 4 describes the experimental results. The evaluation of perplexity and character-error rate versus different factors is conducted. The final conclusions drawn from this study are discussed in Section 5.

2. Maximum Entropy Principle

2.1 ME Language Modeling

The underlying idea of the ME principle [Jaynes 1957] is to subtly model what we know, and assume nothing about what we do not know. Accordingly, we choose a model that satisfies all the information we have and that makes the model distribution as uniform as possible. Using the ME model, we can combine different knowledge sources for language modeling [Berger *et al*. 1996]. Each knowledge source provides a set of constraints, which must be satisfied to find a unique ME solution. These constraints are typically expressed as marginal distributions. Given features $f_1,\cdots,f_N$, which specify the properties extracted from observed data, the expectation of f_i with respect to empirical distribution $\tilde{p}(h,w)$ of history *h* and word *w* is calculated by

$$\tilde{p}(f_i) = \sum_{h,w} \tilde{p}(h,w) f_i(h,w), \tag{4}$$

where $f_i(\cdot)$ is a binary-valued feature function. Also, using conditional probabilities in language modeling, we yield the expectation with respect to the target conditional distribution $p(w|h)$ by

$$p(f_i) = \sum_{h,w} \tilde{p}(h) p(w|h) f_i(h,w). \tag{5}$$

Because the target distribution is required to contain all the information provided by these features, we specify these constraints

$$p(f_i) = \tilde{p}(f_i), \quad \text{for } i = 1,\cdots,N \ . \tag{6}$$

Under these constraints, we maximize the conditional entropy or uniformity of distribution $p(w|h)$. Lagrange optimization is adopted to solve this constrained optimization problem. For each feature f_i, we introduce a Lagrange multiplier λ_i. The Lagrangian function $\Lambda(p,\lambda)$ is extended by

$$\Lambda(p,\lambda) = H(p) + \sum_{i=1}^{N} \lambda_i \left[p(f_i) - \tilde{p}(f_i) \right], \tag{7}$$

with conditional entropy defined by

$$H(p) = -\sum_{h,w} \tilde{p}(h) p(w|h) \log p(w|h). \tag{8}$$

Finally, the target distribution $p(w|h)$ is estimated as a log-linear model distribution

$$p(w|h) = \frac{1}{Z_\lambda(h)} \exp\left(\sum_{i=1}^{N} \lambda_i f_i(h,w) \right), \tag{9}$$

where $Z_\lambda(h)$ is a normalization term in the form of

$$Z_\lambda(h) = \sum_w \exp\left(\sum_{i=1}^{N} \lambda_i f_i(h,w) \right), \tag{10}$$

determined by the constraint $\sum_w p(w|h) = 1$. The General Iterative Scaling (GIS) algorithm or Improved Iterative Scaling (IIS) algorithm [Darroch and Ratcliff 1972; Berger *et al*. 1996; Della Pietra *et al*. 1997] can be used to find the Lagrange parameters λ. The IIS algorithm is briefly described as follows.

Input: Feature functions $f_1, f_2, \cdots, f_N$ and empirical distribution $\tilde{p}(h,w)$

Output: Optimal Lagrange multiplier $\hat{\lambda}_i$

1. Start with $\lambda_i = 0$ for all $i = 1, 2, \cdots, N$.
2. For each $i = 1, 2, \cdots, N$:
   a. Let $\Delta\lambda_i$ be the solution to

$$\sum_{h,w} \tilde{p}(h) p(w|h) f_i(h,w) \exp(\Delta\lambda_i F(h,w)) = \tilde{p}(f_i),$$

   where $F(h,w) = \sum_{i=1}^{N} f_i(h,w)$.
   b. Update the value of λ_i according to $\lambda_i = \lambda_i + \Delta\lambda_i$.
3. Go to step 2 if any λ_i has not converged.

With the parameters $\{\hat{\lambda}_i\}$, we can calculate the ME language model by using Eqs. (9) and (10).

2.2 Relation between ML and ME Modeling

It is interesting to note the relation between maximum likelihood (ML) and ME language models. The purpose of ML estimation is to find a generative model with the maximum likelihood of training data. Generally, the log-likelihood function is adopted in the form of

$$L(p) = \log \prod_{h,w} p(w \mid h)^{\tilde{p}(h,w)} = \sum_{h,w} \tilde{p}(h,w) \log p(w \mid h). \tag{11}$$

Under the same assumption that the target distribution $p(w|h)$ is log-linear, as shown in Eqs. (9) and (10), the log-likelihood function is extended to

$$L(p_\lambda) = \sum_{h,w} \tilde{p}(h,w) \log \frac{\exp\left(\sum_{i=1}^{N} \lambda_i f_i(h,w)\right)}{\sum_{w'} \exp\left(\sum_{i=1}^{N} \lambda_i f_i(h,w')\right)}. \tag{12}$$

By taking the derivative of the log-likelihood function with respect to λ_i and setting it at zero, we can obtain the same constraints in Eq. (6) by using the following derivations:

$$\begin{aligned}
&\sum_{h,w} \tilde{p}(h,w) f_i(h,w) - \sum_{h,w} \tilde{p}(h,w) \sum_{w''} \frac{\exp\left(\sum_{i=1}^{N} \lambda_i f_i(h,w'')\right)}{\sum_{w'} \exp\left(\sum_{i=1}^{N} \lambda_i f_i(h,w')\right)} f_i(h,w'') = 0, \\
&\Rightarrow \sum_{h,w} \tilde{p}(h,w) f_i(h,w) - \sum_{h,w} \tilde{p}(h,w) \sum_{w''} p(w''|h) f_i(h,w'') = 0, \\
&\Rightarrow \sum_{h,w} \tilde{p}(h,w) f_i(h,w) - \sum_{h} \sum_{w''} \tilde{p}(h) p(w''|h) f_i(h,w'') = 0, \\
&\Rightarrow \tilde{p}(f_i) = p(f_i).
\end{aligned} \tag{13}$$

In other words, the ME model is equivalent to an ML model with a log-linear model. In Table 1, we compare various properties using ML and ME criteria. Under the assumption of log-linear distribution, the optimal parameter λ_{ML} is estimated according to the ML criterion. The corresponding ML model $p_{\lambda_{\mathrm{ML}}}$ is obtained through an unconstrained optimization procedure. On the other hand, ME performs the constrained optimization. The ME constraint allows us to determine the combined model $p_{\lambda_{\mathrm{ML}}}$ with the highest entropy. Interestingly, these two estimation methods achieve the same result.

Table 1. Relation between ML and ME language models

Objective function	$L(p_\lambda)$	$H(p)$
Criterion	Maximum Likelihood	Maximum Entropy
Type of search	Unconstrained optimization	Constrained optimization
Search space	$\lambda \in$ real values	p satisfied with constraints
Solution	λ_{ML}	p_{ME}
$p_{\lambda_{\mathrm{ML}}} = p_{\mathrm{ME}}$		

2.3 Minimum Discrimination Information and Latent ME

The ME principle is a special case of minimum discrimination information (MDI) that has been successfully applied to language model adaptation [Federico 1999]. Let $p_b(h,w)$ be the background model trained from a large corpus of general domain, and $p_a(h,w)$ represents the adapted model estimated from an adaptation corpus of new domain. In the MDI adaptation, the language model is adapted by minimizing the distance between the background model and the adapted model. The non-symmetric Kullback-Leibler distance (KLD)

$$D(p_a(h,w), p_b(h,w)) = \sum_w p_a(h,w) \log \frac{p_a(h,w)}{p_b(h,w)} \tag{14}$$

is used for distance measuring. Obviously, when the background model is a uniform distribution, the MDI adaptation is equivalent to the ME estimation. More recently, the ME principle was extended to latent ME (LME) mixture modeling, where the latent variables representing underlying topics were merged [Wang *et al*. 2004]. To find the LME solution, the modified GIS algorithm, called expectation maximization iterative scaling (EM-IS), was used. The authors also applied the LME principle to incorporate probabilistic latent semantic analysis [Hofmann 1999] into *n*-gram modeling by serving the semantic information as the latent variables [Wang *et al*. 2003]. In this study, we use the semantic information as *explicit features* for ME language modeling. Latent semantic analysis (LSA) is adopted to build semantic topics.

3. Integration of Semantic Information and N-Gram Models

Modeling long-distance information is crucial for language modeling. In [Chien and Chen 2004; Chien *et al.* 2004], we successfully incorporated long-distance association patterns and latent semantic knowledge in language models. In [Wu and Khudanpur 2002], the integration of statistical *n*-gram and topic unigram using the ME approach was presented. Clustering of document vectors in the original document space was performed to extract topic information. However, the original document space was generally sparse and filled with noises caused by polysemy and synonymy [Deerwester *et al*. 1990]. To explore robust and representative topic characteristics, here we introduce a new knowledge source to extract long-distance semantic information for *n*-gram modeling. Our idea is to adopt the LSA approach and extract semantic topic information from the reduced LSA space. The proposed procedure of ME semantic topic modeling is illustrated in Figure 1. Because the occurrence of a word is highly related to the topic of current discourse, we apply LSA to build representative semantic topics. The subspace of semantic topics is constructed via *k*-means clustering of document vectors generated from the LSA model. Furthermore, we combine semantic topics and conventional *n*-grams under the ME framework [Chueh *et al.* 2004].

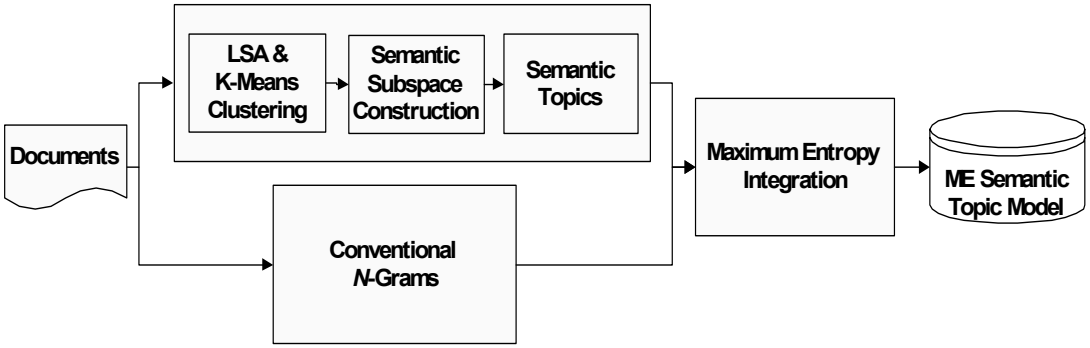


Figure 1. Implementation procedure for ME semantic topic modeling

3.1 Construction of Semantic Topics

Latent semantic analysis (LSA) is popular in the areas of information retrieval [Berry *et al*. 1995] and semantic inference [Bellegarda 2000]. Using LSA, we can extract latent structures embedded in words across documents. LSA is feasible for exploiting these structures. The first stage of LSA is to construct an $M \times D$ word-by-document matrix $\mathbf{A}$. Here, M and D represent the vocabulary size and the number of documents in the training corpus, respectively. The expression for the (i, j) entry of matrix $\mathbf{A}$ is [Bellegarda 2000]

$$a_{i,j} = (1-\varepsilon_i)\frac{c_{i,j}}{n_j}, \tag{15}$$

where $c_{i,j}$ is the number of times word w_i appears in document d_j, n_j is the total number of words in d_j, and ε_i is the normalized entropy of w_i, computed by

$$\varepsilon_i = -\frac{1}{\log D}\sum_{j=1}^{D}\frac{c_{i,j}}{t_i}\log\frac{c_{i,j}}{t_i}, \tag{16}$$

where t_i is the total number of times term w_i appears in the training corpus. In the second stage, we project words and documents into a lower dimensional space by performing singular value decomposition (SVD) for matrix $\mathbf{A}$

$$\mathbf{A} = \mathbf{U}\Sigma\mathbf{V}^T \approx \mathbf{U}_R\Sigma_R\mathbf{V}_R^T = \mathbf{A}_R, \tag{17}$$

where Σ_R is a reduced $R \times R$ diagonal matrix with singular values, $\mathbf{U}_R$ is an $M \times R$ matrix whose columns are the first R eigenvectors derived from word-by-word correlation matrix $\mathbf{A}\mathbf{A}^T$, and $\mathbf{V}_R$ is a $D \times R$ matrix whose columns are the first R eigenvectors derived from the document-by-document correlation matrix $\mathbf{A}^T\mathbf{A}$. The matrices $\mathbf{U}$, Σ, and $\mathbf{V}$ are original full matrices for $\mathbf{U}_R$, Σ_R, and $\mathbf{V}_R$, respectively. The reduced dimension

has the property $R < \min(M, D)$. After the projection, each column of $\Sigma_R \mathbf{V}_R^T$ characterizes the location of a particular document in the reduced *R*-dimensional semantic space. Also, we can perform document clustering [Bellegarda 2000; Bellegarda *et al*. 1996] in the common semantic space. Each cluster consists of related documents in the semantic space. In general, each cluster in the semantic space reflects a particular semantic topic, which is helpful for integration in language modeling. During document clustering, the similarity of documents and topics in the common semantic space is determined by a cosine measure

$$\text{sim}(\mathbf{d}_j, \mathbf{t}_k) = \cos(\mathbf{U}_R^T \mathbf{d}_j, \mathbf{U}_R^T \mathbf{t}_k) = \frac{\mathbf{d}_j^T \mathbf{U}_R \mathbf{U}_R^T \mathbf{t}_k}{\| \mathbf{U}_R^T \mathbf{d}_j \| \mathbf{U}_R^T \mathbf{t}_k \|}, \tag{18}$$

where $\mathbf{d}_j$, $\mathbf{t}_k$ are the vectors constructed by document *j* and document cluster *k*, respectively. $\mathbf{U}_R^T \mathbf{d}_j$ and $\mathbf{U}_R^T \mathbf{t}_k$ are the projected vectors in the semantic space. By assigning topics to different documents, we can estimate the topic-dependent unigram $p(w_i | \mathbf{t}_k)$ and incorporate this information into the *n*-gram model. In what follows, we present two approaches for integrating the LSA information into the semantic language model, namely the linear interpolation approach and the maximum entropy approach.

3.2 Integration via Linear Interpolation

Linear interpolation (LI) [Rosenfeld 1996] is a simple approach to combining information sources from *n*-grams and semantic topics. To find the LI *n*-gram model, we first construct a pseudo document-vector from a particular historical context h. Using the projected document vector, we apply the nearest neighbor rule to detect the closest semantic topic $\mathbf{t}_k$ corresponding to history h. Given *n*-gram model $p_{\text{n}}(w|h)$ and topic-dependent unigram model $p(w|\mathbf{t}_k)$, the hybrid LI language model is computed by

$$p_{\mathbf{LI}}(w|h) = k_{\mathbf{n}} p_{\mathbf{n}}(w|h) + k_{\mathbf{t}} p(w|\mathbf{t}_k), \tag{19}$$

where the interpolation coefficients have the properties $0 < k_{\text{n}}, k_{\text{t}} \le 1$ and $k_{\text{n}} + k_{\text{t}} = 1$. Without the loss of generalization, an *n*-gram model and a topic-dependent model are integrated using fixed weights. Also, the expectation-maximization (EM) algorithm [Dempster *et al*. 1977] can be applied to dynamically determine the value of these weights by minimizing the overall perplexity.

3.3 Integration via Maximum Entropy

More importantly, we present a new ME language model combining information sources of *n*-grams and semantic topics. *N*-grams and semantic topics serve as constraints for the ME estimation. As shown in Table 2, two information sources partition the event space so as to

obtain feature functions. Here, the trigram model is considered. Let w_i denote the current word to be predicted by its historical words. The columns and rows represent different constraints that are due to trigrams and semantic topics, respectively. The event space is partitioned into events $E_{\mathbf{n}}$ and $E_{\mathbf{t}}$ for different cases of *n*-grams and semantic topics, respectively. It comes out of the probability of the joint event $p(E_{\mathbf{n}}, E_{\mathbf{t}})$ to be estimated.

Table 2. Event space partitioned according to trigrams and semantic topics

$w = w_i$	h ends in w_1 (E_{n1})	h ends in w_1, w_2 (E_{n2})	h ends in w_2, w_3 (E_{n3})	…
$h \in \mathbf{t}_1$ (E_{t1})	$p(E_{\mathrm{n1}}, E_{\mathrm{t1}})$	$p(E_{\mathrm{n2}}, E_{\mathrm{t1}})$	$p(E_{\mathrm{n3}}, E_{\mathrm{t1}})$	…
$h \in \mathbf{t}_2$ (E_{t2})	$p(E_{\mathrm{n1}}, E_{\mathrm{t2}})$	$p(E_{\mathrm{n2}}, E_{\mathrm{t2}})$	$p(E_{\mathrm{n3}}, E_{\mathrm{t2}})$	…
⋮	⋮	⋮	⋮	…

Accordingly, the feature function for each column or *n*-gram event is given by

$$f_i^{\mathrm{n}}(h, w) = \begin{cases} 1 & \text{if } h \text{ ends in } w_{i-1}, w_{i-2} \text{ and } w = w_i \\ 0 & \text{otherwise} \end{cases}. \tag{20}$$

In addition, the feature function for each row or semantic topic event has the form

$$f_i^{\mathrm{t}}(h, w) = \begin{cases} 1 & \text{if } h \in \mathbf{t}_k \text{ and } w = w_i \\ 0 & \text{otherwise} \end{cases}. \tag{21}$$

We can build constraints corresponding to the trigrams and semantic topics as follows:

Trigram:

$$\sum_{h,w} \tilde{p}(h) p(w|h) f_i^{\mathrm{n}}(h, w) = \sum_{h,w} \tilde{p}(h, w) f_i^{\mathrm{n}}(h, w) = \tilde{p}(w_{i-2}, w_{i-1}, w_i). \tag{22}$$

Semantic topics:

$$\sum_{h,w} \tilde{p}(h) p(w|h) f_i^{\mathrm{t}}(h, w) = \sum_{h,w} \tilde{p}(h, w) f_i^{\mathrm{t}}(h, w) = \tilde{p}(h \in \mathbf{t}_k, w_i). \tag{23}$$

Under these constraints, we apply the IIS procedure described in Section 2.1 to estimate feature parameters λ_i^{n} and λ_i^{t}, used for combining information sources from trigrams and semantic topics, respectively. Finally, the solution provided by the ME semantic language modeling $p_{\mathrm{ME}}(w|h)$ is computed by substituting λ_i^{n} and λ_i^{t} into Eqs. (9) and (10). We will compare the performance of LI language model $p_{\mathrm{LI}}(w|h)$ and ME language model $p_{\mathrm{ME}}(w|h)$ in the following experiments.

4. Experimental Results

In this study, we evaluate the proposed ME language model by measuring the model perplexity and the character-error rate in continuous speech recognition. The conventional *n*-gram language model is used as the baseline, while the ME language model proposed by Wu and Khudanpur [2002] is also employed for comparison. In addition, we also compare the maximum-entropy-based (ME) hybrid language model with the linear-interpolation-based (LI) hybrid language model. In the experiments, the training corpus for language modeling was composed of 5,500 Chinese articles (1,746,978 words in total) of the TDT2 Corpus, which were collected from the XinHua News Agency [Cieri *et al.* 1999] from January to June in 1998. The TDT2 corpus contained the recordings of broadcasted news audio developed for the tasks of cross-lingual cross-media Topic Detection and Tracking (TDT) and speech recognition. The audio files were recorded in single channel at 16 KHz in 16-bit linear SPHERE files. We used a dictionary of 32,909 words provided by Academic Sinica, Taiwan. 18,539 words in this dictionary occurred at least once in the training corpus. When carrying out the LSA procedure, we built a $32,909 \times 5,500$ word by document matrix $\mathbf{A}$ from the training data. We used MATLAB to implement SVD and *k*-means operations and, accordingly, performed document clustering and determined semantic topic vectors. The topic-dependent unigram was interpolated with the general unigram for model smoothing. The dimensionality of the LSA model was reduced to $R = 100$. We performed the IIS algorithm with 30 iterations. All language models were smoothed using Jelinek-Mercer smoothing [Jelinek and Mercer 1980], which is calculated based on the interpolation of estimated distribution and lower order *n*-grams.

4.1 Convergence of the IIS Algorithm

First of all, we examine the convergence property of the IIS algorithm. Figure 2 shows the log-likelihood of the training data using the ME language model versus different IIS iterations. In this evaluation, the number of semantic topics was set at 30. The ME model that combines the features of trigram and semantic topic information was considered. Typically, the log-likelihood increases consistently with the IIS iterations. The IIS procedure for the ME integration converged after five or six iterations.

4.2 Evaluation of Perplexity

One popular evaluation metric for language models for speech recognition is the *perplexity* of test data. Perplexity can be interpreted as the average number of branches in the text. The higher the perplexity, the more branches the speech recognition system should consider. Generally speaking, a language model with lower perplexity implies less confusion in recognition and achieves higher speech-recognition accuracy. To evaluate the perplexity, we

selected an additional 734 Chinese documents from the XinHua News Agency, which consisted of 244,573 words, as the test data. First, we evaluated the effect of the length of history h for topic identification. The perplexities of LI and ME models are shown in Figures 3 and 4, respectively. Here, C represents the number of document clusters or semantic topics. In the LI implementation, for each length of history h, the interpolation weight with the lowest perplexity was empirically selected. It is obvious that the proposed ME language model outperforms Wu's ME language model [Wu and Khudanpur 2002] and the ME language model outperforms the LI language model. Furthermore, a larger C produces lower perplexity and the case that considering 50 historical words obtains the lowest perplexity. Accordingly, we fixed the length of h at 50 in the subsequent experiments. Table 3 details the perplexities for bigram and semantic language models based on LI and ME. We found that the perplexity was reduced from 451.4 (for the baseline bigram) to 444.7 by using Wu's method and to 441 by using the proposed method when the combination was based on linear interpolation (LI) and the topic number was 30. With the maximum entropy (ME) estimation, the perplexity was further reduced to 399 and 393.7 by using Wu's method and the proposed method, respectively. No matter whether Wu's method or the proposed method was used, the ME language model consistently outperformed the LI language model with different numbers of semantic topics. We also evaluated these models based on the trigram features. The results are summarized in Table 4. We can see that, by integrating latent semantic information into the trigram model, the perplexity is reduced from 376.6 (for the baseline trigram) to 345.3 by using the LI model and to 309.3 by using the ME model, for the case of C=100. The experimental results again demonstrate that the performance improves with the number of semantic topics and that the proposed method consistently outperforms Wu's method, though the improvement is not very significant.

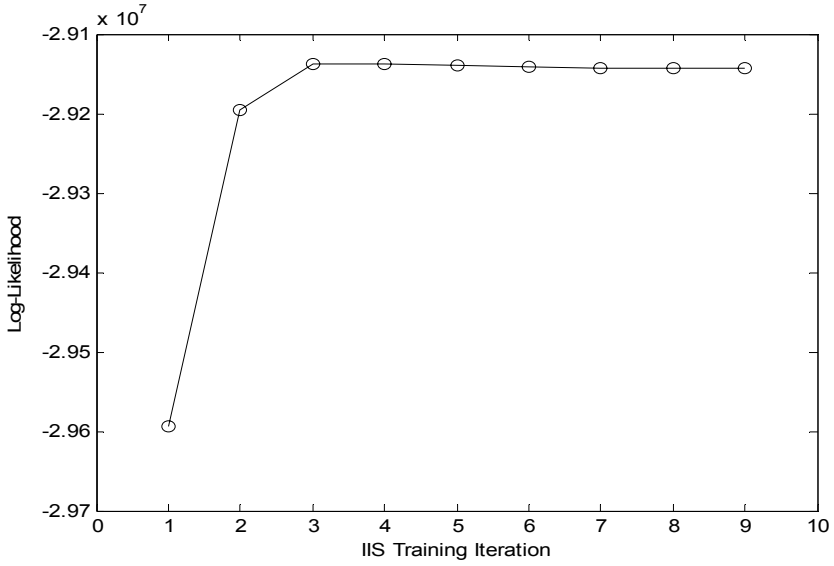


Figure 2. Log-Likelihood of training data versus the number of IIS iterations

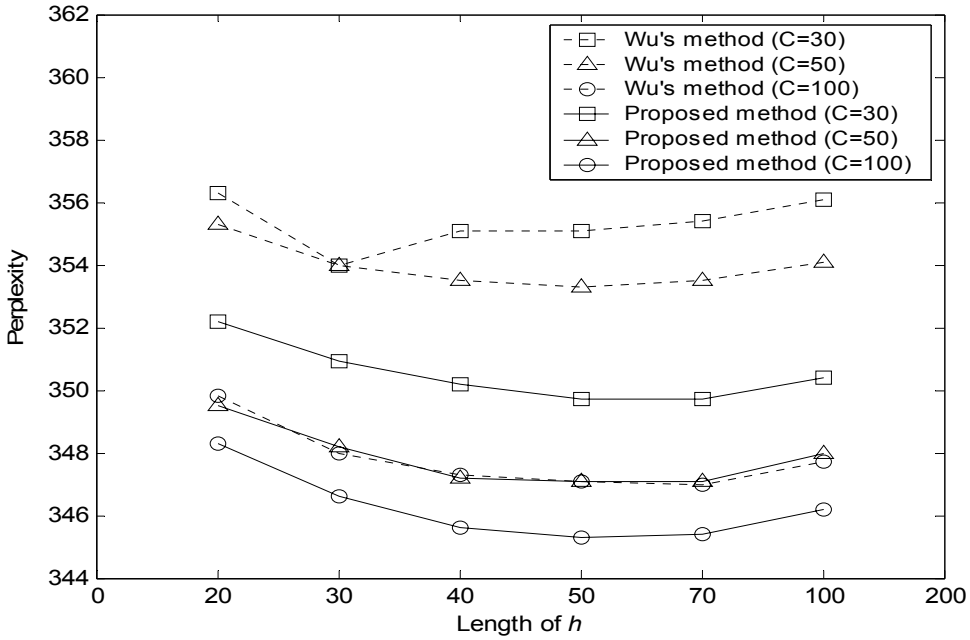


Figure 3. Perplexity of the LI model versus the length of history

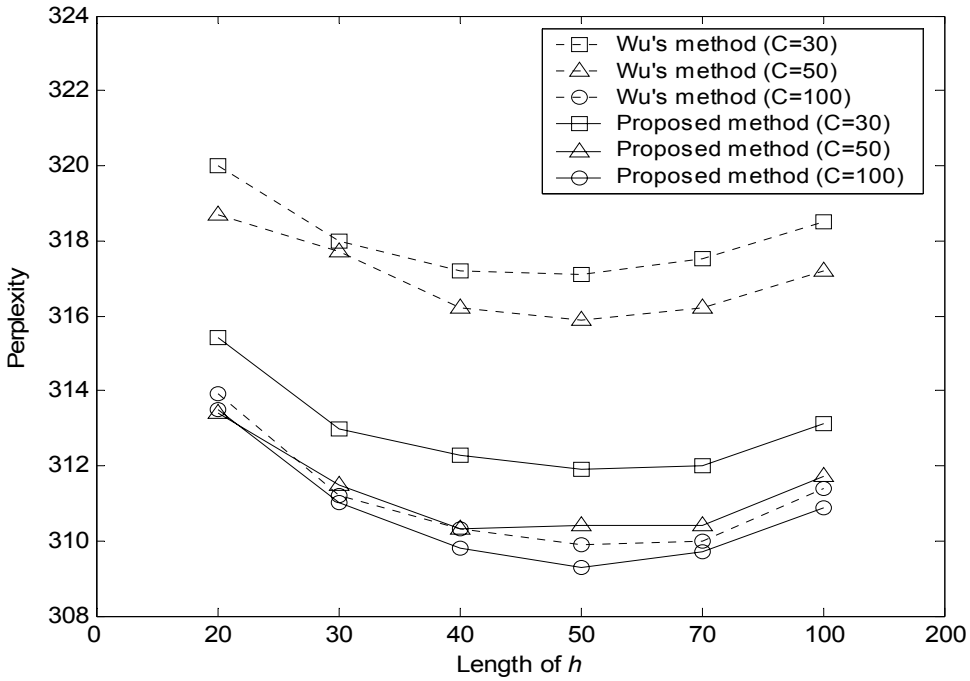


Figure 4. Perplexity of the ME model versus the length of history

Table 3. Comparison of perplexity for bigram, LI and ME semantic language models

	Bigram	Wu's method		Proposed method	
		LI	ME	LI	ME
C=30	451.4	444.7	399	441	393.7
C=50		442.9	402	438	394.8
C=100		437	397.2	435.7	401.2

Table 4. Comparison of perplexity for trigram, LI and ME semantic language models

	Trigram	Wu's method		Proposed method	
		LI	ME	LI	ME
C=30	376.6	355.1	317.1	349.7	311.9
C=50		353.3	315.9	347.1	310.4
C=100		347.1	309.9	345.3	309.3

4.3 Evaluation of Speech Recognition

In addition to perplexity, we evaluated the proposed language models for a continuous Mandarin speech recognition task. Character-error rates are reported for comparison. The initial speaker-independent, hidden Markov models (HMM's) were trained by the benchmark Mandarin speech corpus TCC300 [Chien and Huang 2003], which was recorded in office environments using close-talking microphones. We followed the construction of context-dependent sub-syllable HMM's for Mandarin speech presented in [Chien and Huang 2003]. Each Mandarin syllable was modeled by right context-dependent states where each state had, at most, 32 mixture components. Each feature vector consisted of twelve Mel-frequency cepstral coefficients, one log energy, and their first derivatives. The maximum *a posteriori* (MAP) adaptation [Gauvian and Lee 1994] was performed on the initial HMM's using 83 training sentences (about 10 minutes long), from Voice of America (VOA) news, in the TDT2 corpus for corrective training. The additional 49 sentences selected from VOA news were used for speech recognition evaluation. This test set contained 1,852 syllables, with a total length of 6.6 minutes. To reduce the complexity of the tree copy search in decoding a test sentence, we assumed each test sentence corresponded to a single topic, which was assigned according to the nearest neighbor rule. Due to the above complexity, in this study we only implemented the language model by combining bigram and semantic information in our recognizer. Figure 5 displays the character-error rate versus the number of topics. We can see that the character-error rate decreases in the beginning and then increases as the number of topics increases. Basically, more topics provide higher resolution for representing the

information source. However, the model with higher resolution requires larger training data for parameter estimation. Otherwise, the overtraining problem occurs and the performance degrades accordingly. The character-error rates used in Wu's method and the proposed method are summarized in Table 5. In the case of C=50, the proposed LI model can achieve an error-rate reduction of 8.5% compared to the bigram model, while the proposed ME model attains a 16.9% error-rate reduction. The proposed method in general achieves lower error rates compared to Wu's method.

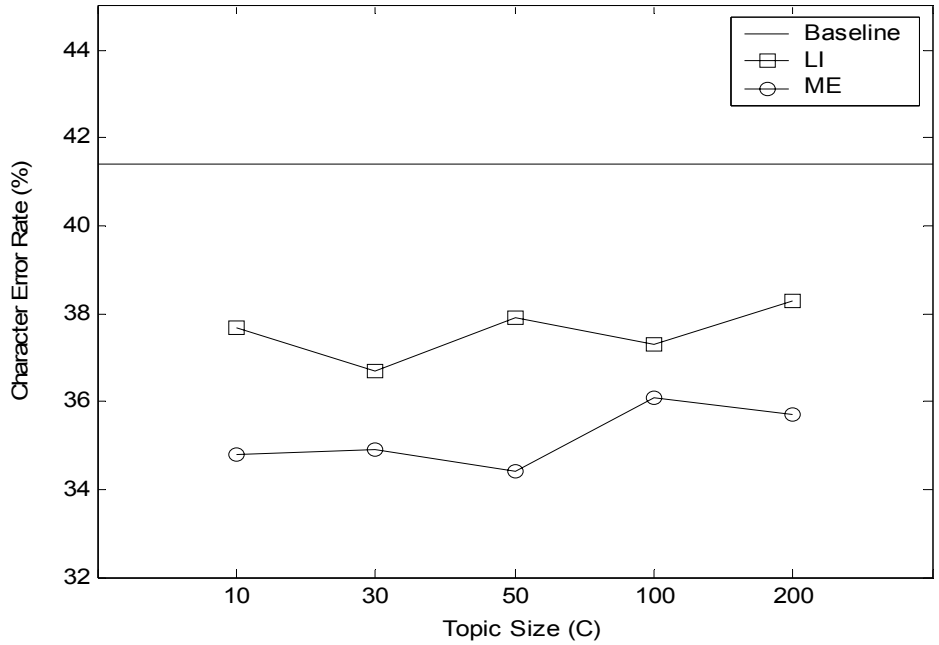


Figure 5. Character error rate (%) versus the number of topics

Table 5. Comparison of character error rate (%) for bigram, LI and ME semantic language models

	Bigram	Wu's method		Proposed method	
		LI	ME	LI	ME
C=30	41.4	38.9	36.4	36.7	34.9
C=50		38.1	36.8	37.9	34.4
C=100		38.3	36.5	37.3	36.1

To evaluate the statistical significance of performance difference between the proposed method and Wu's method, we applied the *matched-pairs* test [Gillick and Cox 1989] to test the hypothesis that the number of recognition errors that occur when using the proposed method is close to that with Wu's method. In the evaluation, we calculated the difference

between character errors induced by Wu's method E_a and the proposed method E_t for each utterance. If the mean of variable $z = E_t - E_a$ was zero, we accepted the conclusion that these two methods are not statistically different. To carry out the test, we calculated the sample mean $\bar{\mu}_z$ and sample variance $\bar{\sigma}_z$ from N utterances and determined the test statistic $\omega = \bar{\mu}_z / (\bar{\sigma}_z / \sqrt{N})$. Then, we computed the probability $P = 2\Pr(z \geq |\omega|)$ and compared P with a chosen significance level α. When $P < \alpha$, this hypothesis was rejected or, equivalently, the improvement obtained with the proposed method was statistically significant. In the evaluation, we applied the respective best case of Wu's method and the proposed method (i.e., ME language modeling, and C=30 for Wu's method but C=50 for the proposed method) in the test and obtained a P value of 0.0214. Thus, at the $\alpha = 0.05$ level of significance, the proposed method is better than Wu's method. That is, the proposed LSA based topic extraction is desirable for discovering semantic information for language modeling.

5. Conclusions

We have presented a new language modeling approach to overcome the drawback of lacking long-distance dependencies in a conventional *n*-gram model that is due to the assumption of the Markov chain. We introduced a new long-distance semantic information source, called the semantic topic, for knowledge integration. Instead of extracting the topic information from the original document space, we proposed extracting semantic topics from the LSA space. In the constructed LSA space with reduced dimensionality, the latent relation between words and documents was explored. The *k*-means clustering technique was applied for document clustering. The estimated clusters were representative of semantic topics embedded in general text documents. Accordingly, the topic-dependent unigrams were estimated and combined with the conventional *n*-grams. When performing knowledge integration, both linear interpolation and maximum entropy approaches were carried out for comparison. Generally speaking, linear interpolation was simpler for implementation. LI combined two information sources through a weighting factor, which was estimated by minimizing the overall perplexity. This weight was optimized globally such that we could not localize the use of weights for different sources. To achieve an optimal combination, the ME principle was applied. Each information source served as a set of constrains to be satisfied for model combination. The IIS algorithm was adopted for constrained optimization. From the experimental results of Chinese document modeling and Mandarin speech recognition, we found that ME semantic language modeling achieved a desirable performance in terms of model perplexity and character-error rates. The combined model, through linear interpolation, achieved about an 8.3% perplexity reduction over the trigram model. The proposed semantic language model did compensate the insufficiency of long-distance information in a conventional *n*-gram model. Furthermore, the

ME semantic language model reduced perplexity by 17.9%. The ME approach did provide a delicate mechanism for model combination. Also, in the evaluation of speech recognition, the ME semantic language model obtained a 16.9% character-error rate reduction over the bigram model. The ME model was better than the LI model for speech recognition. In the future, we will validate the coincidence between the semantic topics discovered by the proposed method and the semantic topics labeled manually. We will also extend the evaluation of speech recognition using higher-order *n*-gram models over a larger collection of speech data.

REFERENCES

Bellegarda, J., "Exploiting latent semantic information in statistical language modeling," *Proceedings of the IEEE*, 88(8), 2000, pp. 1279-1296.

Bellegarda, J., J. Butzberger, Y. Chow, N. Coccaro, and D. Naik, "A novel word clustering algorithm based on latent semantic analysis," *IEEE Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1*,* 1996, pp. 172-175.

Berger, A., S. Della Pietra, and V. Della Pietra, "A maximum entropy approach to natural language processing," *Computational Linguistics*, 22(1), 1996, pp. 39-71.

Berry, M., S. Dumais, and G. O'Brien, "Using linear algebra for intelligent information retrieval," *SIAM Review*, 37(4), 1995, pp. 573-595.

Chelba, C. and F. Jelinek, "Structured language modeling," *Computer Speech and Language*, 14(4), 2000, pp. 283-332.

Chien, J.-T., and C.-H. Huang, "Bayesian learning of speech duration model," *IEEE Transactions on Speech and Audio Processing*, 11(6), 2003, pp. 558-567.

Chien, J.-T., and H.-Y. Chen, "Mining of association patterns for language modeling," *Proc. International Conference on Spoken Language Processing (ICSLP)*, 2, 2004, pp. 1369-1372.

Chien, J.-T., M.-S. Wu, and H.-J. Peng, "Latent semantic language modeling and smoothing," *International Journal of Computational Linguistics and Chinese Language Processing*, 9(2), 2004, pp. 29-44.

Chueh, C.-H., J.-T. Chien, and H. Wang, "A maximum entropy approach for integrating semantic information in statistical language models," *Proc. International Symposium on Chinese Spoken Language Processing (ISCSLP)*, 2004, pp. 309-312.

Cieri, C., D. Graff, M. Liberman, N. Martey, and S. Strassel, "The TDT-2 text and speech corpus," *Proc. of the DARPA Broadcast News Workshop*, 28Feb-3Mar 1999.

Clarkson, P., and A. Robinson, "Language model adaptation using mixtures and an exponential decay cache," *IEEE Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 2*,* 1997, pp. 799- 802.

Darroch, J., and D. Ratcliff, "Generalized iterative scaling for log-linear models," *The Annals of Mathematical Statistics*, 43, 1972, pp. 1470-1480.

Deerwester, S., S. Dumais, G. Furnas, T. Landauer, and R. Harshman, "Indexing by latent semantic analysis," *Journal of the American Society of Information Science*, 41, 1990, pp. 391-407.

Della Pietra, S., V. Della Pietra, and J. Lafferty, "Inducing features of random field," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(4), 1997, pp. 380-393.

Della Pietra, S., V. Della Pietra, R. Mercer, and S. Roukos, "Adaptive language modeling using minimum discriminant estimation," *IEEE Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1, 1992, pp. 633-636.

Dempster, A., N. Laird, and D.Rubin, "Maximum likelihood from incomplete data via the EM algorithm," *Journal of the Royal Statistical Society*, 39(1), 1977, pp. 1-38.

Federico, M., "Efficient language model adaptation through MDI estimation," *Proceedings of European Conference on Speech Communication and Technology (EUROSPEECH)*, 1999, pp. 1583-1586.

Florian, R., and D. Yarowsky, "Dynamic nonlocal language modeling via hierarchical topic-based adaptation," *Proc. 37th Annual Meeting of ACL*, 1999, pp. 167-174.

Gauvain, J.-L., and C.-H. Lee, "Maximum *a posteriori* estimation for multivariate Gaussian mixture observation of Markov chain," *IEEE Transactions on Speech and Audio Processing*, 2(4), 1994, pp. 291-298.

Gillick, L., and S. J. Cox, "Some statistical issues in the comparison of speech recognition algorithms," *IEEE Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1989, pp. 532-535.

Hofmann, T., "Probabilistic latent semantic indexing," *Proc. 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 1999, pp. 50-57.

Jaynes, E., "Information theory and statistical mechanics," *Physics Reviews*, 106(4), 1957, pp. 620-630.

Jelinek, F., and R. L. Mercer, "Interpolated estimation of Markov source parameters from sparse data," *Proc. Workshop on Pattern Recognition in Practice*, 1980, pp. 381-402.

Khudanpur, S., and J. Wu, "Maximum entropy techniques for exploiting syntactic, semantic and collocational dependencies in language modeling," *Computer Speech and Language*, 14, 2000, pp. 355-372.

Kuhn, R., and R. de Mori, "A cache based natural language model for speech recognition," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 12(6), 1992, pp. 570-583.

Ponte, J., and W. Croft, "A language modeling approach for information retrieval," *Proc. ACM SIGIR on Research and Development in Information Retrieval*, 1998, pp. 275-281.

Rosenfeld, R., "A maximum entropy approach to adaptive statistical language modeling," *Computer Speech and Language*, 10, 1996, pp. 187-228.

Wang, S., D. Schuurmans, F. Peng, and Y. Zhao, "Learning mixture models with the regularized latent maximum entropy principle," *IEEE Transactions on Neural Networks*, 15(4), 2004, pp. 903-916.

Wang, S., D. Schuurmans, F. Peng, and Y. Zhao, "Semantic *n*-gram language modeling with the latent maximum entropy principle," *IEEE Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1, 2003, pp. 376-379.

Wu, J., and S. Khudanpur, "Building a topic-dependent maximum entropy model for very large corpora," *IEEE Proceedings of International Conference on Acoustic, Speech and Signal Processing (ICASSP)*, 1, 2002, pp. 777-780.

Zhou, G. D., and K. T. Lua, "Interpolation of *n*-gram and mutual-information based trigger pair language models for Mandarin speech recognition," *Computer Speech and Language*, 13, 1999, pp. 125-141.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 1, March 2006, pp. 57-72

Robust Target Speaker Tracking in Broadcast TV Streams

Junmei Bai*, Hongchen Jiang*, Shilei Zhang*,

Shuwu Zhang* and Bo Xu*

Abstract

This paper addresses the problem of audio change detection and speaker tracking in broadcast TV streams. A two-pass audio change detection algorithm, which includes detection of the potential change boundaries and refinement, is proposed. Speaker tracking is performed based on the results of speaker change detection. In speaker tracking, Wiener filtering, endpoint detection of pitch, and segmental cepstral feature normalization are applied to obtain a more reliable result. The algorithm has low complexity. Our experiments show that the algorithm achieves very satisfactory results.

Keywords: Speaker Tracking, Audio Segmentation, Entropy, GMM

1. Introduction

Broadcast TV programs are rich multimedia information resources. They contain large amounts of AV (audio & video) contents including speech, music, images, motion, text, and so on. Finding ways to extract and manage these various kinds of AV content information is becoming extremely important and necessary for application-oriented multimedia content mining and management. The analysis and classification of audio data are important tasks in many applications, such as speaker tracking, speech recognition, and content-based indexing. Among of them, target speaker tracking in TV streams is an important research topic for TV scene analysis. In contrast with general speaker recognition, speaker detection in audio streams usually requires segments of relatively homogenous speech and speaker tracking in this task should also determine the target speakers' locations, in other word, the starting and ending times. In such applications, effective methods for segmenting continuous audio streams

* The High-Tech. Innovation Center, Institute of Automation, Chinese Academy of Sciences, Beijing, China.
E-mail: {jmbai, hcjiang, slzhang, swzhang, xubo}@hitic.ia.ac.cn

[Received June 15, 2005; Revised January 23, 2006; Accepted January 27, 2006]

into homogeneous segments are required.

The problem of acoustic segmentation and classification has become crucial for the application of automatic speech recognition to audio stream processing. The automatic segmentation of long audio streams and the clustering of audio segments according to different acoustic characteristics have received much attention recently [Lu and Zhang 2002; Chen and Gopalakrishnan 1998; Delacourt and Wellekens 2000; Wilcox *et al*. 1994; Pietquin *et al*. 2002]. To detect target speakers in an audio stream, it is best to segment the audio stream into homogeneous regions according to changes in speaker identity, environmental conditions and channel conditions. In fact, there are no explicit cues of changes among these audio signals, and the same speaker may appear multiple times in audio streams. Thus, it is not easy to segment an audio stream correctly. Various segmentation algorithms proposed in the literature [Lu and Zhang 2002; Chen and Gopalakrishnan 1998; Delacourt and Wellekens 2000; Ajmera *et al*. 2003; Cettolo and Federico 2000] can be categorized as follows [Chen *et al*. 1998]:

1) Decoder-guided segmentation algorithms: The input audio stream is first decoded by an automation speech recognition (ASR) systems, and then the desired segments are produced by cutting the input at the silence locations generated by the decoder. Other information from the decoder, such as gender information, can also be utilized in segmentation.
2) Model-based segmentation algorithms: Different models, e.g., Gaussian mixture models, are build for a fixed set of acoustic classes, such as telephone speech, pure music, etc, from a training corpus. In these schemes, a sliding window approach and multivariate Gaussian models are generally used. Decisions about the maximum likelihood boundary are made.
3) Metric-based segmentation algotithms: The audio stream is segmented at places where maxima of the distances between neighboring windows appear, and distance measures, such as the KL distance and the generalized likelihood ratio (GLR) distance [Fisher *et al*. 2003], are utilized.

These methods are not very successful at detecting acoustic changes that occur in data [Chen *et al*.1998]. Decoder-guided segmentation only places boundaries at silence locations, which in general have no direct connection with acoustic changes in the data. Model-based segmentation usually can not be generalized to unseen acoustic conditions. Meanwhile, both model-based and metric-based segmentation rely on a threshold which sometimes lacks stability and robustness. In addition, model-based segmentation does not generalize to unseen acoustic conditions.

As for target speaker detection, which is similar to general speaker verification, the traditional methods focus on likelihood ratio detection and template matching. Among these approaches, Gaussian Mixture Models (GMMs) have been the most successful so far

[Reynolds *et al*. 2000]. Reynolds also extended of these methods by adapting the speaker model from a universal background model (UBM). The speaker detector we adopted in our experiments is based on adapted GMMs. In the target speaker detecting system, we also used the segmental cepstral mean and variance normalization (SCMVN) to normalize the cepstral coefficients to get robust segmental parameter statistics that are suitable for various kinds of environmental conditions.

2. Overview

The task of automatic speaker tracking involves finding target speakers in test audio streams. Given an audio stream, all the segments containing a target speaker's voice must be located with the starting and ending times. The general approach to speaker tracking consists of three steps: audio segmentation, audio classification, and speaker verification. A complete block diagram of the proposed speaker tracking system is shown in Figure 1. The diagram shows how the components of the system fit together.

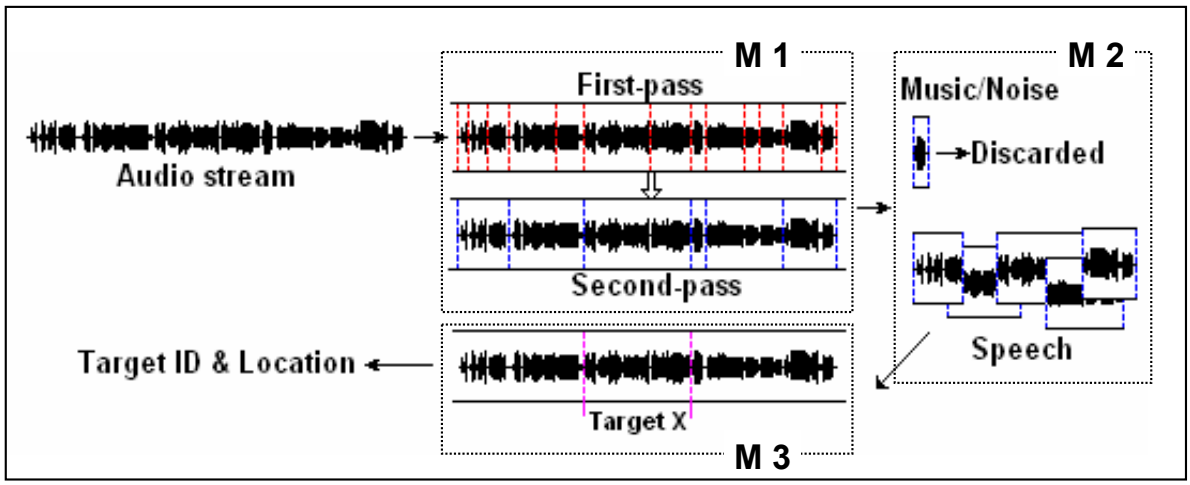


Figure 1. Block diagram of the speaker tracking system components

M1—Segmentation Module, **M2** —Classification Module, **M3**—Speaker verification

The three steps are defined as three modules in Figure 1, denoted as M1, M2, and M3. First, audio streams are segmented in M1 by means of two-pass audio segmentation. Then, in M2, these audio segments are classified into different classes, such as speech, music, noise and so on. Last, the speech segments are tested in M3 to verify if target speakers appear in the audio streams. Sometimes, M2 is not necessary when the speaker verification module can distinguish target speakers with other audio signals with acceptable precision. The individual blocks will be described in detail in following sections.

3. Two-Pass Audio Segmentation

The goal of automatic segmentation of audio signals is to detect changes in speaker identity, environmental conditions, and channel conditions. The problem is to find acoustic change detection points in an audio stream. A two-pass segmentation process for audio streams is presented in this paper. First, audio segmentation based on entropy is used to detect potential audio change points. Then, speaker change boundary refinement based on Bayesian decisions is applied.

3.1 First-Pass Segmentation Based on Entropy

In the first pass, we use entropy measures to determine the turn candidates. Entropy is a measure of the uncertainty or disorder in a given distribution [Papoulis 1991]. There are many methods for calculating entropy. Ajmera calculates entropy based on posterior probabilities and sets it as one of the features for discriminating speech and music [Ajmera *et al.* 2003]. It is a model-based classification scheme that makes decisions based on the scores of audio signals to two models: a speech model and a music model. Generally, the speech model is estimated from lots of speech spoken by different speakers, and it acts as a universal model. Thus, it is not suitable for distinguishing different speakers, particularly unknown speakers.

The entropy method used in this work is also an extension of the model-based segmentation scheme. Generally, model-based methods apply a maximum likelihood of the Gaussian process with a penalty weight to detect turns in audio streams. By appropriately defining this penalty, one can generate decisions based on the Akaike Information Criterion (AIC), the Bayesian Information Criterion (BIC), the Consistent AIC (CAIC), the Minimum Description Length (MDL) principle, and the Minimum Message Length (MML) principle. It has been found that BIC, MDL, and CAIC give the best results and that with proper tuning, all three produce comparable results [Cettolo *et al*. 2000].

In this paper, entropy is calculated based on statistical parameters of audio features. The decision rule is not based on scores but on the shape of the entropy contour. In order to clearly show the performance of our method, it is compared with BIC in this paper. The of entropy-based audio segmentation scheme is described in detail in the following:

Entropy of a Gaussian Random Variable [You *et al*. 2004]**:**

Assume a random variable X of dimension K. The entropy of the random variable (RV) is computed by first estimating its probability distribution function (pdf). We can compute the pdf either from the RV's histogram or from a parameterized distribution. The latter is used to reduce the amount of computation. Assume that the pdf follows a K-dimensional Gaussian density:

$$P(X) = |2\pi\Sigma|^{-\frac{1}{2}} e^{-\frac{1}{2}(X-\mu)^T\Sigma^{-1}(X-\mu)}, \tag{1-a}$$

where μ is the mean vector and Σ is the covariance matrix. The entropy of X is

$$E(X) = -\int P(X)LogP(X)dX \quad . \tag{1-b}$$

Eq. (1-b) can been replaced by [You *et al.* 2004]:

$$E(X) \approx KLog\sqrt{2\pi} + \log\Sigma \, . \tag{1-c}$$

The entropy curve of a speech signal in a sliding window is calculated as follows:

Define $\mathbf{Y} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots \mathbf{y}_N\}$ as the cepstral sequence of an audio stream in a sliding window of N frames. At a given frame index j $(1 < j < N)$, the sliding window is partitioned into two sub-windows. Denote them as $\mathbf{Y}_{j(l)} = \{\mathbf{y}_1, \mathbf{y}_2, \cdots, \mathbf{y}_j\}$ and $\mathbf{Y}_{j(r)} = \{\mathbf{y}_{j+1}, \mathbf{y}_{j+2}, \cdots, \mathbf{y}_N\}$, respectively. The lengths of the two windows are $N_{j(l)}$ and $N_{j(r)}$ ($N_{j(l)} + N_{j(r)} = N$) respectively. Assume that each window is generally modeled with a multivariate Gaussian density, such as $\mathbf{N}(\boldsymbol{\mu}_{j(l)}, \boldsymbol{\Sigma}_{j(l)})$ and $\mathbf{N}(\boldsymbol{\mu}_{j(r)}, \boldsymbol{\Sigma}_{j(r)})$, respectively. The sum of the entropy of each side of the window is computed as follows:

$$E_{j(l)} = \sum_{i=1}^{N_{j(l)}} (K\log\sqrt{2\pi} + \Sigma_{j(l)}^{(i)}) = \sum_{i=1}^{N_{j(l)}} K\log\sqrt{2\pi} + N_{j(l)}\Sigma_{j(l)}) \, , \tag{1-d}$$

$$E_{j(r)} = \sum_{i=1}^{N_{j(r)}} (K\log\sqrt{2\pi} + \Sigma_{j(r)}^{(i)}) = \sum_{i=1}^{N_{j(r)}} K\log\sqrt{2\pi} + N_{j(r)}\Sigma_{j(r)} \, . \tag{1-e}$$

Then, the segmentation entropy at j can be computed as follows

$$\begin{aligned} E(j) &= \sum_{i=1}^{N_{j(l)}} K\log\sqrt{2\pi} + N_{j(l)} \times \log|\mathbf{\Sigma}_{j(l)}| + \sum_{i=1}^{N_{j(r)}} K\log\sqrt{2\pi} + N_{j(r)} \times \log|\mathbf{\Sigma}_{j(r)}|, \\ E(j) &= NK\log\sqrt{2\pi} + N_{j(l)} \times \log|\mathbf{\Sigma}_{j(l)}| + N_{j(r)} \times \log|\mathbf{\Sigma}_{j(r)}| \, . \end{aligned} \tag{1-f}$$

$NK\log\sqrt{2\pi}$ is a constant. It is ineffective for determining the entropy curve and can been omitted. Thus, the segmentation entropy at j can be simplified as follows:

$$E(j) = N_{j(l)} \times \log|\mathbf{\Sigma}_{j(l)}| + N_{j(r)} \times \log|\mathbf{\Sigma}_{j(r)}| \, . \tag{1}$$

Decision making is performed by analyzing the entropy curve in each window as described below.

H1: There is a potential change point in the sliding window.

The sequence entropy value shows a step-down change until it reaches a minimal value at time t. Then, it increases gradually. t can be considered as a change point. Here, $t = \arg\min_j E(j)$.

H0: There is no any change point in the sliding window.

The segmentation entropies vary randomly.

We can make the following observations:

a) The minimal entropy varies for different window sizes and different audio conditions. However, if the entropy decreases gradually till it reaches a minimal polar, then it increases gradually, there is a changing point at the polar.

b) Since there are fewer data in the region close to the original point on the left, the segmentation entropies in this region are unable to describe the entropy curve accurately. The same is true, on the right. Thus, these two regions are ignored in the final analysis. As shown in Figure 2, θ_t is defined as the number of the points ignored on each side.

c) The basic processing unit or the sliding window length is 3s; however, the overlapping length between two neighboring windows is not fixed. If there is not change point in the prior window, the overlapping length is 1.5s; otherwise, the overlapping length is relative to the location of the last change point in the prior window.

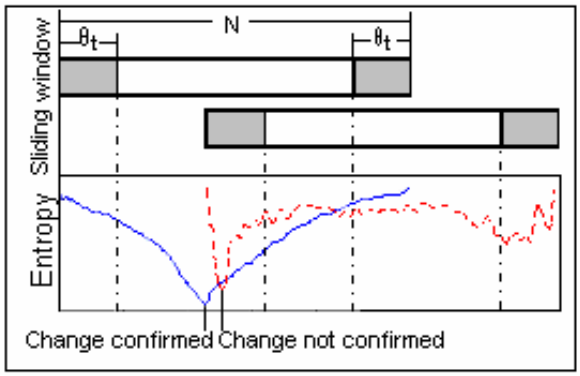


Figure 2. Samples of entropy contour

3.2 Second-Pass Speaker Change Boundary Refinement

Often there are false positives in potential speaker change points obtained with the algorithms described above. To remove false positives, a refinement algorithm is applied. The algorithm is based on the dissimilarity between two adjacent sub-segments. In this step, two distance measures, the Bayesian decision and KL distance, are applied to validate or discard candidates from the first pass. Suppose the feature vector extracted from each sub-segment is Gaussian, and assume that the feature probability distribution functions are n-variable normal populations, such as $\mathbf{N}(\boldsymbol{\mu}_1, \boldsymbol{\Sigma}_1)$ and $\mathbf{N}(\boldsymbol{\mu}_2, \boldsymbol{\Sigma}_2)$. The Bayesian decision distance between two speech segments can be defined as [Lu *et al*. 2002]

$$D_{BD} = \frac{1}{2} tr[(\Sigma_1 - \Sigma_2)(\Sigma_1^{-1} - \Sigma_2^{-1})]. \quad (2)$$

Provided that the speech of each segment can been modeled with a multivariate Gaussian density, the Kullback-Leibler (KL) distance between two speech slices is defined by [Homayoon *et al*. 1998]

$$D_{KL} = \frac{\sum_{i=1}^{M}(w_1^i d_1^i + w_2^i d_2^i)}{\sum_{i=1}^{M}(w_1^i + w_2^i)}, \tag{3}$$

$$d_{ij} = \frac{\sum_1^i}{\sum_2^j} + \frac{\sum_2^j}{\sum_1^i} + \frac{\mu_1^i - \mu_2^j}{\sum_1^i} + \frac{\mu_1^i - \mu_2^j}{\sum_2^j}, \tag{3-a}$$

$$d_1^i = \min_j(d_{ij}), \tag{3-b}$$

$$d_2^j = \min_i(d_{ij}). \tag{3-c}$$

$w_t\{w_t^i \mid i = 1,2,...,M\}$ is the mixture weight of the model of the *t*th segment.

In general, if two speech segments are spoken by the same speaker, the distance between them will be small; otherwise, the distance will be large. Thus, we apply a simple criterion: if the distance between two speech segments is larger than a given threshold, then these two segments can be considered as to be spoken by different speakers. The thresholds adopted in this study were set experientially. Figure 3 shows an example of two-pass audio segmentation of 26-second long audio stream. The audio stream includes two speakers and 3 speaker change boundaries, which are 7s, 15s and 22s respectively. It can be seen that the number of the potential boundaries is greater than that of real boundaries. The Bayesian decision is performed on these potential speaker change points to remove the false ones. In Figure 3, D_{bd},

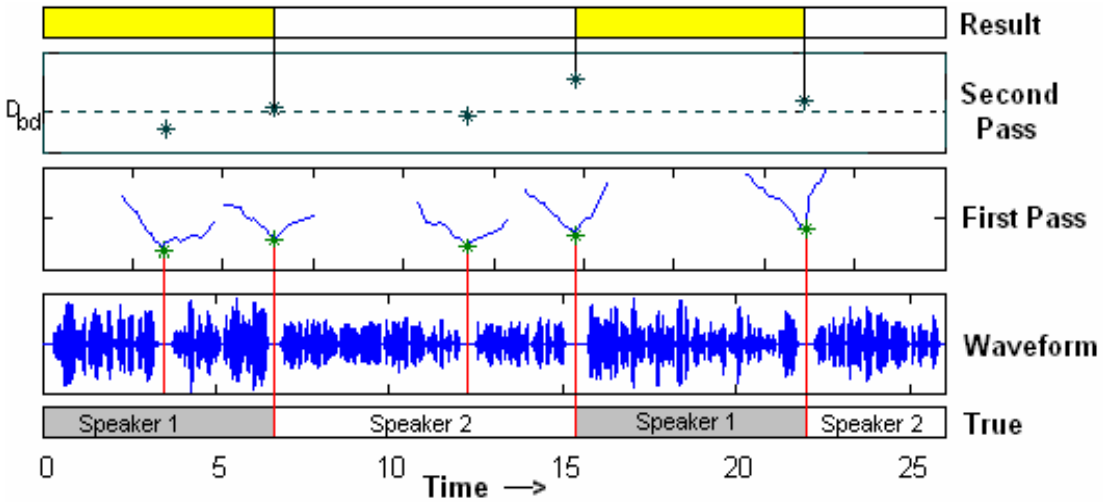


Figure 3. Two pass segmentation procedure: the entropy contour and the Bayesian decision

is an experiential threshold for the Bayesian decision.

3.3 Audio Segments Classification

The aim of audio classification is to distinguish speech and other audio signals. Currently, the state-of-the-art method of classification is based on GMM. Four models were applied in our experiments, a speech model, an unvoiced model, a music model, and a noise model, to classify the audio segments. Among them, only speech slices were used to detect target speakers in subsequent processing.

4. Target Speaker Tracking System

To a certain extent, speaker detection is similar to automatic speaker verification (ASV), which is used to verify the identity claimed by a speaker. The general approach to speaker detection mainly consists of four parts: speech signal pre-processing, speaker feature extraction, speaker modeling, and recognition.

4.1 Speech Slice Pre-Processing

In automatic speaker detection systems, the mismatch between training and recognition, generated by additive or convoluting noises, often severely degrades the recognition accuracy. In addition, the non-speech signals, mainly silence and noise, contain little information of speakers. They are the same for each person and contain no distinguishing features, only ones that are confusing for speaker detection. They can degrade the discrimination ability for different speakers. Thus it is necessary to reduce the noise and discard the irrelevant information before performing speaker features extraction. In our experiments, we applied Wiener filtering and pitch-based endpoint detection in speech slice pre-processing.

Though pitch is a robust feature to noise, it is difficult to measure pitch accurately and reliably for several reasons. Since the key is to detect the active endpoints by means of pitch,

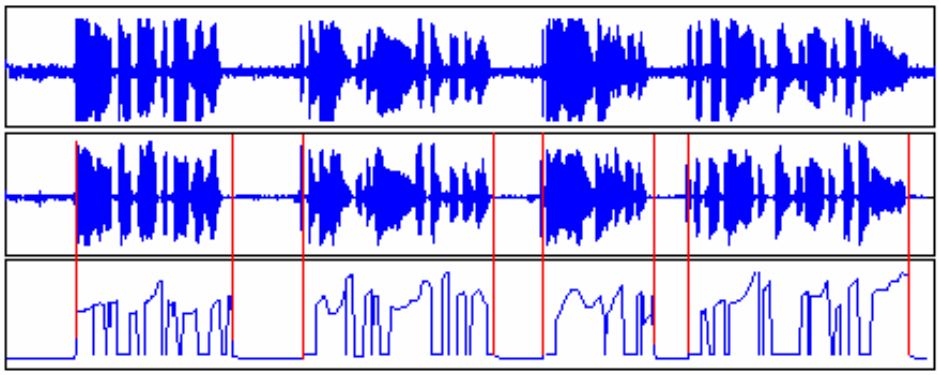


Figure 4. Pre-processing by Wiener filter and endpoint detection on pitch

it is not appropriate to put much emphasis on the precision values of pitch. Moreover, we use wiener filter to alleviate the noise, which makes the pitch detection more precise. In Figure 4, we can see that the pitch, which is mostly susceptible to noise, is near the endpoint. We set the active endpoint at the place where the pitch is less than zero. Although the pitch may not be precise, it is valid for endpoint detection. If the interval between two adjacent unvoiced frames is too short, say, less than 10 frames, then these unvoiced frames will be reserved.

4.2 Speaker Feature Extraction and Normalization

Although there is no exclusive feature for distinguishing different speakers' voices, the speech spectrum has been shown to be very effective for speaker recognition. This is because the spectrum reflects a person's vocal tract structure, the predominant physiological factor that distinguishes one person's voice from others. The Mel-frequency cepstral coefficient (MFCC) vectors have been used extensively for speaker recognition. However, the MFCC features can be severely affected by noise. Thus, some methods should be used to compensate for the corrupted speech.

The widely used method for Cepstral feature normalization is Cepstral mean subtraction (CMS). CMS is performed over an entire file, and it can reduce the stationary convolution noise caused by the channel. However, CMS can also reduce some slow dynamic features of speakers. In this study, the segmental cepstral mean and variance normalization (SCMVN) were used. SCMVN is calculated as follows:

$$\hat{x}_{t+(L-1)/2}(i) = \frac{x_{t+(L-1)/2}(i) - \mu_t(i)}{\sigma_t(i)} \quad , \tag{4}$$

where, X_t is the feature vector at time t, and L is the length of the sliding window; t, which is the first frame in the current window, gives the current place of the window in the speech; $\mu_t(i)$ and $\sigma_t(i)$ are the means and variances of the feature vector in this window. It should be noted that the length of the window, L, is fixed since the normalization of all feature should be uniform. In addition, a proper value of L should be adopted. The estimations of $\mu_t(i)$ and $\sigma_t(i)$ may be imprecise if L is too short. And if it is too long, the calculation will be more complex.

SCMVN has two possible effects: Firstly, it can reduce the action of addition noises in feature variance. Generally, addition noises result in decreased variance. Secondly, the features are mapped to a normal distribution over a sliding window, which is helpful for modeling the speakers' GMM later in speaker recognition.

4.3 Speaker Tracking

The basic speaker detector is a likelihood ratio detector with target and alternative probability distributions. For text independent speaker verification GMMs (Gaussian Mixture Models) have been most successful so far [Reynolds *et al*. 2000]. The test ratio may be expanded by using the Bayesian rule:

$$T(x) = \frac{f(\lambda_i \mid x)}{f(\lambda_{UBM} \mid x)} = \frac{g(\lambda_{UBM}) f(x \mid \lambda_i)}{g(\lambda_i) f(x \mid \lambda_{UBM})}, \tag{5}$$

where $g(\lambda)$ is the prior density. In fact, the prior density is assumed to be equal for the UBM and the target model. The set of feature vectors is often very large, hence, the value of $f(..)$ is often very small. Therefore, it is common to compute the logarithm of the test ratio instead. The log-test ratio is given by

$$\theta_i(x) = \log f(x \mid \lambda_i) - \log f(x \mid \lambda_{UBM}). \tag{6}$$

Thus, the most suitable speaker models can be found based on the largest likelihood ratio. If the largest likelihood ratio is larger than a threshold, the identity of the current speaker can be determined; otherwise, the current segment is considered for a new speaker. In this way, we can determine the identity of the current speaker. Suppose that so far, *K* speakers are registered in the speaker model database; the concrete expression for identifying the speaker of the current segment is as follows:

$$ID = \begin{cases} \arg\max_i \theta_i & if \ \max \theta_i \geq \theta_0 \\ Non & if \ \max \theta_i \leq \theta_0 \end{cases}, \tag{7}$$

where $1 \leq i \leq K$ and *Non* represents a new speaker. The threshold θ_0 can be either speaker dependent or speaker independent. The purpose of speaker dependent thresholds is to reduce the negative effects of speaker dependent variability on performance. Another solution is to apply a reversible transform to score values so that the result is equivalent to using speaker dependent thresholds. For practical reasons, the transform is based on impostor scores rather than on true speaker scores. One such method, currently known as znorm [Reynolds 1995], transforms the impostor score distribution to zero mean and unit variance, while a Gaussian distribution is assumed. For an observation x and a claimed identity λ_i, the normalized log-test is given by

$$\theta_i^{Znorm}(x) = \frac{\theta_i(x) - \mu_i}{\sigma_i}, \tag{8}$$

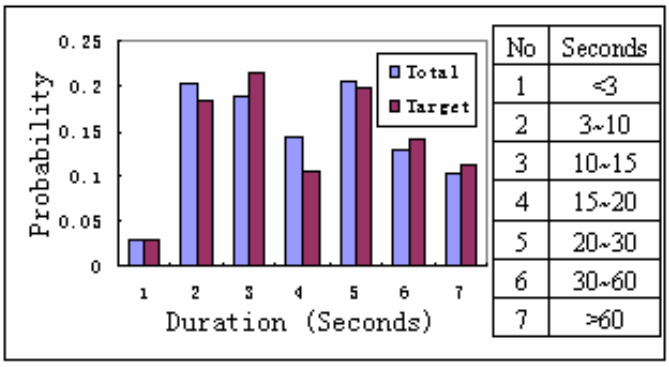
where μ_i and σ_i are the moment estimates of the impostor score distribution for a speaker λ_i.

5. Experiments

5.1 Database

The proposed audio segmentation and speaker tracking algorithms were evaluated using an audio database, recorded directly from the CCTV news channel. The database is composed of about 10 hours of audio streams, which are from different TV programs, such as news, interviews, music, and movies. In the test database, at least one target speaker appeared in each file.

Figure 5 reports the length statistics for the segments in the test set. A segment was defined as a contiguous portion of an audio signal, homogeneous in terms of acoustic source and channel. The duration of two adjacent turns in the test data varied from 2 seconds to 5 minutes. In Figure 5, the x-axis is the time duration, and the number reprensents the duration. On the right side of Figure 5, the first row corresponds to the second row. For example, 1="<3s" and 2="3s~10s". This shows that about 2% of the audio segments were less than 3 seconds long. We tested the performance with windows of 2 seconds and 3 seconds. It was observed that the performance decreased dramatically when the two-second window is used. Thus, we selected 3 seconds as the unit window size. That is to say, for those speaker segments which were less than 3 seconds long, the segmentation results were not reliable.



No	Seconds
1	<3
2	3~10
3	10~15
4	15~20
5	20~30
6	30~60
7	>60

Figure 5. Histogram for the audio segment durations of all audio streams and Target speakers

5.2 Experimental Setup

The input audio stream was first down-sampled into a uniform format: 8KHZ, 16bits, and mono-channel, regardless of the input format. In first pass segmentation, the speech stream was then pre-emphasized and divided into sub-segments using 3-second window with some overlapping. That is, the basic processing unit was 3 seconds; however, the temporal resolution of segmentation was not fixed. If there was no change point in the prior window,

the overlapping length was 1.5 second, or the overlapping length was relative to the location of the last change point in the prior window.

In target speaker detection, the most important features extracted from the frame were MFCC and pitch. MFCC and the delta parameters were employed to characterize target speakers. The 16-dimensional MFCC vector and 1-dimensional energy were extracted from the speech signal every 12 ms with a 24 ms window. The delta parameters were then computed and appended to the previous vectors, thus producing a 34-dimensional feature vector.

There were a total of 40 target speakers, who consisted of reporters, commentators, comperes, and interviewees. The target models were adapted from UBM parameters, using two minutes of training data. The target speaker detector was a likelihood ratio detector for adaptation GMMs. Our UBM was a 1024 mixture GMM, trained using about 6 hours of broadcast data from 60 speakers with equal number of males and females. Target models were derived by means of Bayesian adaptation from the UBM parameters using two minutes of training data. Only the mean vectors were adapted, as this had been observed to provide better performance. The amount of adaptation of each mixture mean was data dependent.

The baseline system only used CMS to alleviate noises; then, Wiener filtering, endpoint detection via the pitch, and SCMVN were applied, respectively.

5.3 Experimental Results

The criteria of performance for audio segmentation and speaker detection are shown below:

For audio segmentation, the false alarm rate (FAR) and missed detection rate (MDR) were calculated as follows [Lu *et al*. 2002]:

$$FAR = \frac{number\ \ of\ \ false\ \ \mathrm{det}ection}{number\ \ of\ \ false\ \ \mathrm{det}ection + number\ \ of\ \ true\ \ change} \times 100\% \ ,$$

$$MDR = \frac{number\ \ of\ \ miss\ \ \mathrm{det}ection}{number\ \ of\ \ true\ \ change} \times 100\% \ .$$

For target speaker detection: the false alarm rate (FA), false reject rate (FR), and Equal Error Rate (ERR) were calculated as follows:

$$FA = \frac{number\ \ of\ \ false\ \ accepted\ \ as\ \ t\mathrm{arg}ets}{number\ \ of\ \ segments\ -\ number\ \ of\ \ t\mathrm{arg}et\ \ segments} \times 100\% \ ,$$

$$FR = \frac{number\ \ of\ \ False\ \ rejected}{number\ \ of\ \ true\ \ t\mathrm{arg}et\ \ segments} \times 100\% \ .$$

When $FA = FR$, $ERR = FA = FR$. ERR is a common criterion for judging the

performance of speaker verification systems.

5.3.1 Results of Audio Segmentation

The statistics results of audio segmentation are shown in Table 1. In first pass segmentation, the entropy-based method was better than BIC, particularly in *MDR*. However, *FAR* was still a little high with both methods. This was mostly due to the following reasons. First, *FAR* in long segments is great. As shown in Figure 5, about 10% of the segments were longer than 60 seconds. These long segments resulted in 5%-10% *FAR*. Second, the noise information increased *FAR*. In fact, some of the false detections in long segments affected the speaker-tracking performance a little, for about 20 seconds of speech is enough for speaker recognition. What's more, about 25% *FAR* appeared in speech signals. Thus, a speaker change boundary refinement algorithm was applied to remove false positives. As shown in Table 1, second pass refinement decreased *FAR* from 30.4% to 14.4% and from 31.2% to 14.9% based on the entropy results and on BIC results, respectively, In *MDR,* there was about a 0.6% increase based on the entropy results and a 1.8% increase based on the BIC results. As for the second pass refinement schemes, Bayesian decision was little better than the KL distance.

Table 1. The results of two-pass audio segmentation

		FAR	**MDR**			**FAR**	**MDR**
First Pass	Entropy	30.4%	6.5%	**Second pass**	BD	**14.4%**	**7.1%**
					KL	16.0%	7.3%
	BIC	31.2%	13.1%	**Second pass**	BD	14.9%	14.5%
					KL	15.2%	15.0%

5.3.2 Results of Target Speaker Detection

There are many factors that affect the performance of speaker detection. Among them, the target speech duration is a very important factors especially for the false reject (*FR*) rate in target speaker detection. Generally speaking, the shorter the speech is, the higher *FR* and *FA* will be. As shown in Figure 6, the FR rate decreased greatly with increasing time when the speech durations were less than 20 seconds long. And it changed little when the speech durations were longer than 20 seconds. Noise is another interference factor in target speaker detection. The performance in target speaker detection with different strategies is shown in Table 2. The EER and the relative improvement compared with the baseline are illustrated in Table 2. Compared with the conventional CMS, SCMVN was better at compensating for the corruption caused by noise. Its effect was clear in target speaker detection. Wiener filtering and endpoint detection based on pitch are only used in speaker detection because the error in

noise estimating in Wiener filtering increases when the noise environment changes, so it cannot work well with long speech durations. In this case, Wiener filtering is not helpful but costly in terms of time. And silence signals are useful for audio segmenting, so they are not discarded. However, their effects in speaker detection were clear in our experiments. The integrated system with SCMVN, Wiener filtering, and endpoint detection showed the best performance.

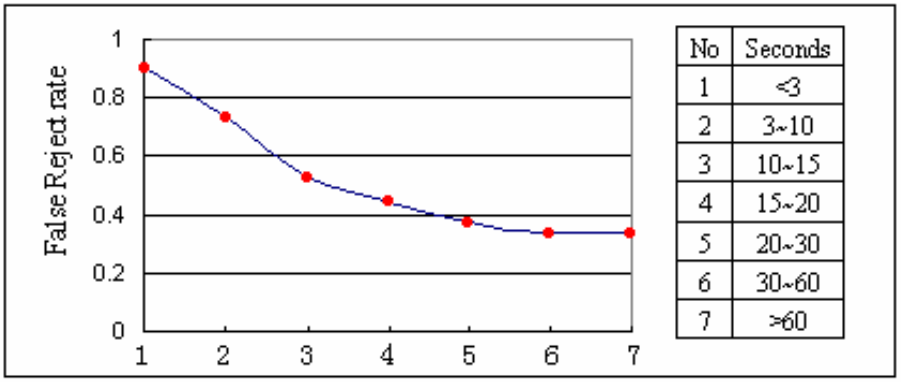


Figure 6. The FR of speaker detection at different speech durations

Table 2. The ERR of target speaker detection

Case	ERR	ERR Relative Reduction
Baseline	25.2%	0
WF + ED	23.3%	9.1%
SCMVN + WF + ED	22.8%	9.5%

6. Conclusion

In this paper, we have presented a novel approach to unsupervised audio segmentation and a speaker tracking system. A two-pass audio change detection algorithm has been proposed, which includes potential audio change detection and speaker boundary refinement. The results of two-pass audio segmentation are classified as speech or music according their characteristics. Speaker tracking is based on the results of audio classification. In speaker tracking, Wiener filtering, endpoint detction based on pitch, and the segmental cepstral mean and variance normalization are applied to get more reliable results. The algorithm achieves satisfactory accuracy.

There is still room for improvement of the proposed approach. In the experiments, we found that if two speakers were speaking synchronously, it was not easy to detect the change boundary. It was also found that the same speaker in various environments sometimes was detected as different speakers or rejected. This indicates that our compensation for the

mismatch effect of the environment or channel is still insufficient. In our future research, we will focus on these issues.

Acknowledgement

Support provided by the National Natural Science Foundation of China (NSFC) under grant no. 60475014 and the National Hi-tech Research Plan under grant no. 2005AA114130 is gratefully acknowledged.

Reference

Ajmera, J., I. McCowan, and H. Bourlard, "Speech/music segmentation using entropy and dynamism features in a HMM classification framework," *Speech Communication*, 40(3), 2003, pp.351-363.

Bai, J., S. Zhang, R. Zheng, S. Zhang, and B. Xu, "Audio Segmentation and Speaker Detection in Broadcast TV Stream," In *Proc. of 10th International Conference on SPEECH and COMPUTER*, 2005, Patras, Greece, pp.547-550.

Beigi, H. S. M., S. H. Maes, and J. S. Sorensen, "A Distance Measure Between collections of Distributions and Its Application to Speaker Recognition," In *Proc. of Int. Conf. On Acoustic, Speech, and Signal Processing*, *1998*, Seattle, Washington, USA, pp. 753-756.

Campbell, J.P., "Speaker Recognition: a Tutorial." *Proceedings of The IEEE*, 85(9), 1997, pp. 1437-1462.

Cettolo, M., and M. Federico., "Model Selection Criteria for Acoustic Segmentation," In *Proc. of the ISCA ITRW ASR2000 Automatic Speech Recognition*, 2000, Paris, France, pp. 221-227.

Chen, S., and P.S. Gopalakrishnan, "Speaker, Environment, and Channel Change Detection and Clustering via the Bayesian Information Criterion," In *Proc. of the DARPA Broadcast News Transcription and Understanding Workshop*. 1998. Virginia, USA, pp.127-132.

Delacourt, P., and C.J. Wellekens, "DISTBIC: A speaker-based segmentation for audio data indexing," *Speech Communication*, 32 (1-2), 2000, pp.111-126.

Dunn, R.B., D. A. Reynolds, and T. F. Quatieri, "Approaches to speaker detection and tracking in conversational speech," *Digital Signal Processing,* 10 (1-3), 2000, pp.93–112.

Fisher, E., J. Tabrikian, and S. Dubnov, "Generalized likelihood ratio test for voiced/unvoiced decision using the harmonic plus noise model," In *Proc. Of Int. Conf. On Acoustic, Speech, and Signal Processing*, 2003, Hong Kong, pp. 440-443.

Jin, H., F. Kubala, and R. Scwartz, "Automatic Speaker Clustering," In *Proc. of the DARPA Speech Recognition Workshop*, 1997, pp. 108-111.

Lu, L., and H.J. Zhang. "Speaker change detection and tracking in real-time news broadcasting analysis, " In *Proc. of the 10th ACM International Conference on Multimedia*, 2002, Juan-les-Pins, France, pp. 602-610.

Mori, K., and S. Nakagawa. "Speaker Change Detection and Speaker Clustering Using VQ Distortion for Broadcast News," In *Proc. of Int. Conf. On Acoustic, Speech, and Signal Processing*, 2001, Salt-Lake City, USA, pp.413-416.

Papoulis, *A Probability, Random Variables, and Stochastic Processes*. 3rd ed. McGraw-Hill, 1991.

Pietquin, O., L. Couvreur, and P. Couvreur, "Applied Clustering for Automatic Speaker-based segmentation of Audio Material," *Belgian Journal of Operations Research, Statistics and Computer Science*, 41(1-2), 2002, pp. 69-81.

Reynolds, D. A., T.F. Quatieri, and R.B. Dunn, "Speaker Verification. Using Adapted Gaussian Mixture Models," *Digital Signal Processing*, 10(1-3), 2000, pp. 19-41.

Reynolds, D.A., "Speaker Identification and Verification using Gaussian Mixture Speaker Models," *Speech Communication,* 17(1-2), 1995, pp. 91-108.

Sigler, M.A., U. Jain, B. Raj, and M. Stern. "Automatic Segmentation Classification and Clustering of Broadcast News Audio," In *Proc. of the DARPA Speech Recognition Workshop*, 1997, pp. 97-99.

Tsekeridou, S., and Ioannis Pitas, "Audio-Visual Content Analysis for Content-Based Video Indexing," In *Proc. of 1999 IEEE Int. Conf. on Multimedia Computing and Systems*, 1999, Florence, Italy, pp. 667--672.

Wilcox, L., F. Chen, D. Kumber, and V. Balasubramanian, "Segmentation of Speech Using Speaker Identification," In *Proc. of Int. Conf. On Acoustic, Speech, and Signal Processing*, 1994, Adelaide, Australia, pp.161-164.

Wu, J., J. Droppo, L. Deng, and A. Acero, "A noise-robust ASR Front-end Using Wiener filter Constructed from MMSE Estimation of Clean Speech and Noise," In *Proc. of IEEE Automatic Speech Recognition and Understanding Workshop*, St. Thomas, U.S, 2003, pp.321-326.

You, H., Q. Zhu, and A. Alwan, "Entropy-based variable frame rate analysis of speech signals and its application to ASR," In *Proc. of Int. Conf. On Acoustic, Speech, and Signal Processing*, 2004, Montreal, Canada, pp. 529-552.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 1, March 2006, pp. 73-86

A Fast Framework for the Constrained Mean Trajectory Segment Model by Avoidance of Redundant Computation on Segment[1]

Yun Tang*, Wenju Liu*, Yiyan Zhang+ and Bo Xu*

Abstract

The segment model (SM) is a family of methods that use the segmental distribution rather than frame-based density (e.g. HMM) to represent the underlying characteristics of the observation sequence. It has been proved to be more precise than HMM. However, their high level of complexity prevents these models from being used in practical systems. In this paper, we propose a framework that can reduce the computational complexity of the Constrained Mean Trajectory Segment Model (CMTSM), one type of SM, by fixing the number of regions in a segment so as to share the intermediate computation results. Our work is twofold. First, we compare the complexity of SM with that of HMM and point out the source of the complexity in SM. Secondly, a fast CMTSM framework is proposed, and two examples are used to illustrate this framework. The fast CMTSM achieves a 95.0% string accurate rate in the speaker-independent test on our mandarin digit string data corpus, which is much higher than the performance obtained with HMM-based system. At the mean time, we successfully keep the computation complexity of SM at the same level as that of HMM.

Keywords: Speech Recognition, Segment Model, Mandarin Digit String Recognition

1. Introduction

The Hidden Markov Model (HMM) [Rabiner *et al.* 1993] has been used successfully for

[1] This work was supported in part by the China National Nature Science Foundation (No. 60172055) and the Beijing Nature Science Foundation (No.4042025).

* Institute of Automation, Chinese Academy of Sciences, P.O.Box 2728, nlpr, Beijing 100080
Tel: +086 01062659279 Fax: +086 01062551993
E-mail: tangyun@nlpr.ia.ac.cn

+ Institute of Scientific and Technical Information of China

[Received Feb. 24, 2005; Revised Oct. 8, 2005; Accepted Oct. 12, 2005]

acoustic modeling in many speech recognition systems. Given the state sequence, feature vectors are assumed to be conditionally independent, and the task of extracting the trajectory can be elegantly achieved by applying the Viterbi algorithm frame by frame. However, the above assumption is far from realistic, which limits the HMM's ability to capture the relations within a segment. Another weakness of HMM is that it is not accurate enough to represent a non-stationary observation sequence by means of a piecewise constant state [Deng *et al.* 1994; Hon *et al.* 1999]. In order to handle these problems, a lot of methods have been proposed, including SM [Ostendorf *et al.* 1996], which is a family of methods among them.

SM is totally different from HMM in terms of its segmental decoding method and potential for accomplishing some tasks effectively that are naturally difficult for an HMM based system, since it integrates more segmental information into the decoding process, produces the n-best list during the decoding process etc. However, the good acoustic modeling of SM is at the cost of high computation, which is much higher than that of HMM. It prevents SM from being applied in practical systems. The high complexity of SM is mainly due to the segment evaluation process. Segment evaluation cannot be decomposed and the intermediate computation information is not shareable between different segments even when two segments only differ by one frame. Previous work accelerated SM using efficient segment pruning algorithms. V. Digalakis *et al*. [1992] proposed a pruning method to speed up SM. They estimate the score of a segment from part of the segment. Then those hypotheses with low likelihood are pruned before the whole segment is evaluated. The amount of reduction in computation depends on the discrimination ability of the feature vector. S. Lee *et al.* [1998] and J. Glass [2003] proposed a landmark-based algorithm that reduces the search space by detecting the potential boundaries of phonemes with the aid of special features or HMM decoders, so that the number of the possible hypothesized segments in the search space can be reduced greatly. However, since the detection of boundaries is unreliable and not accurate enough, the efficiency of this algorithm is discounted. The most important point is that the speed of SM based on the above methods is still far slower than that of HMM, since the computations performed by these algorithms are based on segments, while in the case of HMM, they are based on frames. In this paper, we propose a framework to reduce the complexity of the Constrained Mean Trajectory Segment Model (CMTSM) [Ostendorf *et al.* 1996], one family of SM. In this new framework, CMTSM can divide segment computations into frame computations, which are shared between different segments; thus, the redundant computations of segments can be avoided. Guided by this framework, we have measured the complexity of Stochastic Segment Model (SSM) [Ostendorf *et al.* 1989] based on the number of Gaussian mixture models evaluated during recognition, and found that the complexity is not proportional to the product of the model's number and the maximum allowable duration, but is only related to the number of models, or more exactly, to the number of regions in the system.

The complexity of SSM is on the same level as that of HMM. The speed of the Parametric Trajectory Model (PTM) [Gish *et al*. 1992; Deng *et al.* 1994], another type of CMTSM, can also be greatly enhanced with some minor modifications of the original algorithm, based on our framework.

The rest of this paper is organized as follows. SM is introduced in the next section by comparing HMM with SM in terms of modeling and decoding. Then, in Section 3, we present the fast framework for CMTSM and two examples, the fast SSM and the fixed PTM, illustrate it. Section 4 presents experimental results obtained with the fast framework. Finally, conclusions are drawn in Section 5.

2. Segment Model and Decoding

2.1 Introduction to the Segment Model

In HMM, the model unit is the state, and the relations among feature vectors are represented by the relations among the states mapping to these features. In SM, the model unit is based on segments, such as phonemes, syllables, and words. Hence, the relations between feature vectors in the same segment are modeled directly. The probability density of a variable length feature sequence $x_1^l = \{x_1, x_2, ...x_l\}$ measured by SM can be represented as follows:

$$p(x_1^l \mid \alpha) = f(x_1^l \mid \alpha) g(x_1^l \mid \alpha), \tag{1}$$

where α is the label of the acoustic model, $f(x_1^l \mid \alpha)$ is the output density of SM, and $g(x_1^l \mid \alpha)$ is a segment level score, such as the duration score.

2.2 Decoding Comparison between HMM and SM

The goal of a speech recognizer is to find the most likely word sequence given sentence x_1^T. Let α_1^N be the label sequence of acoustic models representing words intended by the speaker, who produces x_1^T above. That is,

$$\hat{\alpha}_1^N = \arg\max_{N,\alpha_1^N} p(\alpha_1^N \mid x_1^T) = \arg\max_{N,\alpha_1^N} p(\alpha_1^N) p(x_1^T \mid \alpha_1^N), \tag{2}$$

where $p(\alpha_1^N)$ is the probability measured by the language model and $p(x_1^T \mid \alpha_1^N)$ is the density measured by acoustic models. More exactly, $p(x_1^T \mid \alpha_1^N)$ is the product of acoustic models' densities in different segments of x_1^T:

$$p(x_1^T \mid \alpha_1^N) = \sum_{S \in \Lambda_{T,N}} \prod_{i=1}^{N} p(x_{S(i-1)+1}^{S(i)} \mid \alpha_i) \approx \max_{S} \prod_{i=1}^{N} p(x_{S(i-1)+1}^{S(i)} \mid \alpha_i), \tag{3}$$

$S(i-1) < S(i)$, $S(0) = 0$ and $S(N) = T$,

where $\Lambda_{T,N}$ is the segmentation boundary set dividing a T-length sequence into N parts and $S(i)$ is the boundary point of segment i.

2.2.1 Decoding in HMM

The above decoding process is accomplished by the Viterbi algorithm in HMM. We will take a left-to-right HMM without state skipping as an example to illustrate decoding in HMM:

$$J_m^*(\alpha,i) = \ln p(x_m \mid \alpha,i) + \max_{i-1\le j\le i}(J_{m-1}^*(\alpha,j)).\ \ i \le m \le T,\ 1 \le \alpha \le |\Omega|,\ 2 \le i \le L_\alpha, \tag{4}$$

where $J_m^*(\alpha,i)$ is the maximum accumulated score for the state sequence from the 1-th frame to the m-th frame, given state i and model label α for frame x_m; $p(x_m \mid \alpha,i)$ is the state score of frame x_m; $|\Omega|$ is the number of models; L_α is the number of states for α.

The above formula can be applied to all internal states of each model (i.e., $i \ge 2$). At the boundary of the model, i.e., $i = 1$, the formula is in the following form:

$$J_m^*(\alpha,1) = \ln p(x_m \mid \alpha,1) + \max_{1\le\beta\le|\Omega|}[J_{m-1}^*(\beta,L_\beta) + \ln(p(\alpha)), J_{m-1}^*(\alpha,1)]. \tag{5}$$

The final solution for the best path is

$$J^* = \max_{1\le\alpha\le|\Omega|}[J_T^*(\alpha,L_\alpha)], \tag{6}$$

and the best path can be obtained by backtracking the best final score.

The cost of the Viterbi algorithm is essentially the cost of computing the state scores. According to (4) and (5), the amount of computation required for the state scores is proportional to the number of states in each model and the observation sequence length. If the pruning is not considered, the approximate time complexity for the Viterbi algorithm is $O(T\cdot|\Omega|\cdot\bar{L}\cdot C_S)$, where C_S is the time cost of computing $p(x_m \mid \alpha,i)$ and $\bar{L}$ is the average number of states in each model.

2.2.2 Decoding in SM

SMs have to explore all possible segment boundaries due to the segmental decoding, whereas the problem of obtaining exact acoustic model boundaries can be avoided with HMM, since the frame that the exit state maps to is the boundary of the model. Though the decoding procedure can be performed by means of dynamic programming, the complexity of SM is still much higher than that of HMM. The decoding formula for SM is

$$J_m^* = \max_{\tau,\alpha}\{J_\tau^* + \ln[p(x_\tau^m \mid \alpha)](m-\tau) + \ln[P(\alpha)] + C\}, \tag{7}$$

where J_m^* is the accumulated score of the best model sequence ending at time point m and C is the insert factor for each segment. The best segment sequence can be obtained by back-tracking from the best final score J_T^*.

Given the beginning (or end) point of models, the decoder has to hypothesize segments with different durations, from the minimum to the maximum length, to determine the other boundary point of the segment that may spring from this point. Assuming that the maximum allowed duration is $L_{\max}$, we find that the time complexity of SM is $O(C_{Seg} \cdot T \cdot |\Omega| \cdot L_{\max})$, where C_{Seg} is the time cost of a segment and is comparable with or even more complex than $C_S \cdot \bar{L}$ in HMM. Hence, SM is more costly than HMM.

3. Fast Framework for CMTSM

As discussed in Section 2.2.2, the high complexity of SM is due to two factors. First, SM explores more hypothesized models than HMM does in each frame; second, in each frame, SM needs to measure the densities of segments that pass this point, whereas HMM only needs to evaluate the densities of states mapping to this point. The second factor is more important for current SM systems, since density evaluation represents the lion's share in the whole computation. Figure 1 shows the percentage of the time spent on density evaluation against the total time needed for the digit string recognition task with HMM and SSM. The model unit for SSM and HMM is the context independent whole-word. The computation involved in density evaluation is extremely time-consuming in the case of the conventional SSM and 97.6% of the time is spent obtaining segment scores, whereas the corresponding percentage in the case of HMM is only 51.4%. The time cost ratio for density evaluation in SM is much higher than that in HMM. The key advantage of our fast framework is that it changes the computation in SM from segment-based style to frame-based style and the frame-based results can be shared by different segments. Such transformation can be achieved in one family of SM, i.e., CMTSM. In the fast SM, which we will describe below, the time cost ratio for density evaluation is lowered to 64.2%, close to that of HMM. The details of experimental setup and total time used for decoding will be given in Section 4 (Table 5).

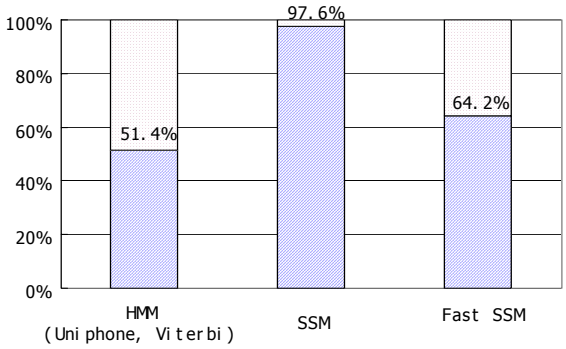


Figure 1. Percentage of the time for density evaluation in the decoding

Those SMs, including SSM and PTM, whose segmental distributions are modeled by means of region distributions while frame-based features are assumed to be conditionally independent given the region sequence, are called CMTSM. The so-called region here is similar to the conception of the state in HMM, which is the basic unit used to measure the probability distribution of a frame. The value of $f(x_1^l \mid \alpha)$ in (1) is the product of a series of frame-based region scores [Ostendorf *et al.* 1996]:

$$\ln f(x_1^l \mid \alpha) = \sum_{i=1}^{l} \ln p(x_i \mid \alpha, r_i, l), \tag{8}$$

where $p(x_i \mid \alpha, r_i, l)$ is the score of region r_i in frame x_i for model α, given duration l. The mapping of a feature vector to a region is only related to the segment duration and its position in the segment. So the measurement of the frame score for a specific region is unrelated to other frames or other regions.

The assumption, frame-based features being assumed to be conditionally independent given the region sequence in CMTSM, guarantees to change the density evaluation from segment-based style to frame-based style, and the segment score can be obtained by recombining the region scores in an efficient way. However, these frame-based results can not be shared among different segments, since region models are conditional on the segment duration, as Equation (8) shows. We relax the modeling condition by assuming that the region model is independent of the segment duration. In order to achieve this, we use linear time resampling to map the variable length segment x_1^l to a fixed length feature sequence y_1^L, so all the segment models have the same duration. In other words, the duration can be ignored in region models. In this way, the region scores can be shared by segments with different durations. The resampling function is [Ostendorf *et al.* 1989]

$$y_i = x_{\left\lfloor \frac{i}{L} \cdot l \right\rfloor}, \ 0 \le i < L, \tag{9}$$

where $\lfloor z \rfloor$ is the largest integer $n \le z$. Equation (8) can be simplified as

$$\ln f(x_1^l \mid \alpha) = \sum_{i=1}^{L} \ln p(y_i \mid \alpha, r_i). \tag{10}$$

In short, to speed up CMTSM, we first resample a variable length segment to obtain a fixed length sequence and then measure region models using the fixed length segment model. In our implementation, a memory table is used to store the region scores in different frames. The computation at each feature frame consists of two parts: the computations for all the region models mapping to that frame, and addition operations needed to obtain the scores of segments over that frame; whereas the conventional SMs have to completely measure all the segments that pass that frame. This is the framework we propose to reduce the complexity of

CMTSM. In the following, two examples will be given to illustrate the framework.

3.1 Complexity of SSM

SSM represents a variable length observation sequence by means of a fixed length region sequence. A resampling function is used to map the variable length segment x_1^l to the fixed length model region sequence y_1^L. Two kinds of resampling can be adopted to map a variable length sequences to a fixed *L*-length sequence. One is space-based resampling, and the other is linear time resampling [Ostendorf *et al.* 1989]. Space resampling chooses *L* sampling points, which are equidistant (Euclidean distance) along the segment trajectory, by means of interpolation. The linear time resampling is similar to (9). The two resampling functions have similar performances as reported by M. Ostendorf. Given model α, the log conditional probability of a segment x_1^l is

$$\log[P(x_1^l \mid a)] = \sum_{i=1}^{L} \log[p(y_i \mid a, r_i)] + \lambda \log[P(l \mid \alpha)], \tag{11}$$

where $P(l \mid \alpha)$ is the duration distribution of the segment, given α.

According to (11), C_{Seg} is proportional to the number of regions in the model and can be represented as $C_R \cdot \bar{L}$, where C_R is the time cost of region model $p(y_i \mid a, r_i)$ and $\bar{L}$ is the average number of region models. The complexity of SSM is $O(T \cdot |\Omega| \cdot \bar{L} \cdot C_R \cdot L_{\max})$, according to the conclusion drawn in Section 2.2.

Based on the discussion of the fast CMTSM, SSM can be greatly accelerated by choosing the linear time resampling, and the computation of region scores in (11) can be shared by segments with different durations. The total cost of the SSM algorithm is essentially the cost of computing the region scores. Thus, the time complexity, measured based on the number of evaluated region models, is $O(T \cdot |\Omega| \cdot \bar{L} \cdot C_R)$.

3.2 Fast PTM

In PTM, the features in a segment are modeled by means of parameterization through constant, linear, or higher order polynomial regression instead of by using a sequence of regions to represent the curve of the trajectory. Given model α, a speech segment x_1^l can be modeled as

$$x_i = \sum_{p=0}^{P} B_\alpha(p) \left(\frac{i-1}{l-1} \right)^p + E_i(\Sigma_\alpha), \tag{12}$$

where $B_\alpha(p)$ is the polynomial regression coefficient of order P and E_i is a residual error with covariance matrix Σ_α after fitting data using the first term in (12). The frame score with duration l is,

$$p(x_i \mid \alpha, r_i) = \frac{1}{(2\pi)^{d/2} \mid \Sigma_\alpha \mid^{1/2}},$$
$$\exp\{-\frac{1}{2}(x_i - \sum_{p=0}^{P} B_\alpha(p)\left(\frac{i-1}{l-1}\right)^p)' \Sigma_\alpha^{-1}(x_i - \sum_{p=0}^{P} B_\alpha(p)\left(\frac{i-1}{l-1}\right)^p)\}. \quad (13)$$

In the conventional method, the region models are conditional on the segment duration. The durations of segments are different and so are the P-order polynomials in (12). As a result, the frame score $p(x_i \mid \alpha, r_i)$ calculated using (13) can not be shared among different segments, even when two segments only differ from each other by one frame. For example, assume that two segments for the same model both begin at the 1-st frame and that the first one ends at the 10-th frame and the other at the 15-th frame. The polynomial coefficients of these two segments are listed in Table 1.

Table 1. The polynomial coefficients of the segments with different durations

Rate \ No.i	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
i/10	0.1	0.2	0.3	0.4	0.5	0.6	0.7	0.8	0.9	1	—	—	—	—	—
i/15	0.07	0.13	0.20	0.27	0.33	0.40	0.47	0.53	0.60	0.67	0.73	0.80	0.87	0.93	1.00

In fast PTM, we also fix the number of regions in the model and use the linear time resampling to map a variable length segment to the region sequence with a fixed duration, so the region model is independent of the segment duration. In this way, the speed of PTM can be greatly enhanced.

There are two main factors that limit errors introduced by resampling of the original feature on an acceptable scale, and these errors do little harm to the accuracy of the system. The first is the slowly time varying nature of speech signals [Rabiner *et al*. 1993], which can be seen as a quasi-stationary process. The speech feature vector is similar to the nearby feature vectors. Usually, the length of a region sequence in our system is longer than the average length of an observation sequence, so the region model can well approximate the feature that would appear in the corresponding position of a segment. The second factor is that resampled features are used in both the training phase and recognition phase, which guarantees the compatibility of resampled features with models. Figure 2 shows the trajectories of a speech data sequence and two man-made data sequences produced by 5-order polynomial regression. One polynomial fit the original observation sequence, and the other one fit the fixed length observation sequence resampled from the original features. The fixed length was 56. All the trajectories are shown in normalized time axes in Figure 2. It can be seen that the two regression trajectories are almost tiled together and that the linear time resampling does little harm to the model.

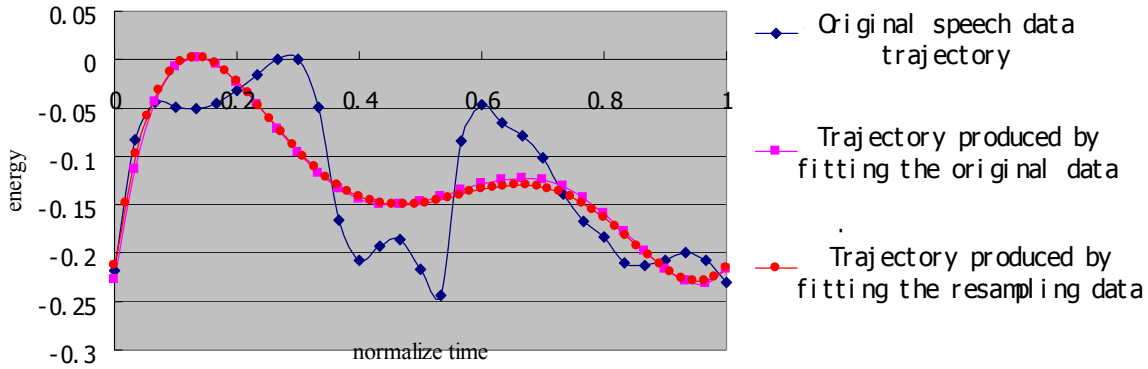


Figure 2. Trajectories for an original data sequence and two man-made data sequences produced by polynomial regression.

4. Experiments and Results

Our methods were verified on a mandarin digit string recognition system. Digit string recognition has achieved a satisfied performance in English [Rabiner *et al.* 1989]. However, due to the serious confusion among mandarin digits, the state-of-the-art of mandarin digital string recognition systems does not match that of the English counterpart. The performance of a recognition system depends not only on the size of the vocabulary but also on the degree of confusability among words in the vocabulary. Mandarin is a monosyllabic and tonal language, in which a syllable is composed of a syllable initial, syllable final, and tone. Insertion or deletion errors mainly exist in non-syllable initial words, e.g., "1," "2," and "5." If a digit's syllable final is similar to that of non-syllable initial words, it is difficult to segment the non-syllable initial words and segmentation errors tend to occur, such as the confusability between "5" and "55." Substitution errors mainly occur among "6," "9," and "yiao" ("yiao" is the variation of "1"), or between "2" and "8" because of the similarity of their syllable finals.

4.1 Experimental Setup

Data Corpus: the mandarin digit string database includes the speech of 55 males, each of which made 80 utterances. The length of each utterance varies from 1 to 7 digits with an average length of 4. The vocabulary is "0" to "9"and "yiao1." Statistical results show that all digits have the same probability of being uttered, and that the connections among digits are considered and balanced. At the same time, the positions (start/middle/end) of the digits in strings are also balanced [Deng *et al.* 2000]. We took the speech of the first 40 speakers (ordered by the name of speakers) as the training set and the data from the remaining 15 speakers as the test set. The frame size of acoustic features was 25.6 ms and the frame shift

was 10ms. For each frame, a 39-dimension vector, composed of 12 MFCC and 1 normalized energy, 13 first order deviations and 13 secondary deviations, was calculated.

Baseline Systems: three systems were studied, HMM, SSM, and PTM. The state in HMM (or region in SSM) was modeled by the Gaussian Mixture Model (GMM). In all the experiments, a diagonal covariance matrix was assumed for each GMM. Table 2 compares the baselines' configures. Sts is the number of states, Res is the number of regions, MCs is the number of mixture components, "ID" means the acoustic unit is modeled by the whole-word (context independent), and "D" means the acoustic unit is tri-word based (context dependent) in Table 2.

Table 2. The settings of the models in the experiments

Model	Sts (Res)	MCs	Type
HMMI	8	16	ID
HMMII	8	16	D
SSMI	25	5	ID
SSMII	40	10	ID
PTM		15	ID

The HMMs in the experiments were structured left to right with 8 states, 6 emitting distributions, and no state skipping, except for the "silence" model, which had 3 states and 1 emitting distribution. HMMI was decoded with the conventional Viterbi algorithm, and HMMII adopted a two-pass search strategy: the first pass was implemented using the forward Viterbi algorithm, and the second pass using the backward A* decoding to integrate the duration distribution [Deng *et al.* 2000]. HMMII was modeled using the tri-word model, while the other systems were modeled by the whole-word model. SSMI and SSMII were two SSM systems. SSMI had Gaussian densities comparable with those of HMMI so that a comparison of the performance between SSM and HMM would be meaningful. SSMII, which had more region models and mixture components than SSMI, achieved the best performance in the digit string recognition task. The baseline PTM was consisted of three sub-segments [Deng *et al.* 1994] and the polynomial regression order was 2.

4.2 Experimental Results

Table 3 compares the modeling ability of HMM and SSM. It can be seen that SSM achieved better performance than HMM. SSMI performed better than not only HMMI but also HMMII. When the number of regions and mixture components increased, SSMII achieved 95% string accuracy for mandarin digit strings. "S Cor," "W err," " Ins err," "Del err" and "Sub err" are the string correction rate, word error rate, insertion error rate, deletion error rate and the

substitution error rate respectively.

Table 3. Comparison of digit string recognition performance achieved with SSM and HMM

	S Corr.	W err	Ins err	Del err	Sub err
HMMI	87.10%	3.88%	0.64%	2.14%	1.10%
HMMII	91.80%	2.53%	0.19%	0.87%	1.47%
SSMI	92.52%	2.58%	0.23%	0.72%	1.63%
SSMII	95.00%	1.64%	0.35%	0.27%	1.02%

For the purpose of comparison, the number of regions in a sub-region sequence was fixed at 20 and the total number of region models was 60 (20×3) in each fast PTM. The feature frames in a segment were mapping to these 60 region models using the time linear resampling. The other parameters were the same as those for the baseline PTM system. Table 4 presents the recognition results obtained with the fixed PTM and the original PTM. It shows that the performance of the PTM system was slightly downgraded following the modifications but still acceptable (0.6% string accuracy loss).

Table 4. Recognition results obtained with PTM and fixed PTM

Methods	S Corr.	W err	Ins err	Del err	Sub err
PTM	95.10%	1.53%	0.30%	0.24%	0.99%
Fixed PTM	94.50%	1.82%	0.14%	0.68%	1.00%

The efficiency of the different recognition systems, including the conventional SSM, fast SSM, PTM, fixed PTM, and HMM, is compared in Table 5. We used the utterances of one person (80 strings) in the test set. As shown in Table 5, the fast algorithm boosted SMs and reduced the complexity of SM to the same level of that of HMM. The most noticeable achievement was made by the fixed PTM system, which was 90 times faster than the original one.

Table 5. Time comparison of SM, Fast SM, and HMM

	T (s)
HMMI	35
HMMII	87
Conventional SSMI	1816
Fast SSMI	101
Fast SSMII	162
PTM	23854
Fixed PTM	271

5. Conclusions

In this paper, a fast framework has been proposed to boost the speed of CMTSM based on the assumption that the region model of SM is independent from the segment duration, so that intermediate results are shared during the computation of segment scores. Two examples, SSM and PTM, have been used to illustrate this framework. The improved systems are far more effective than the original models. Based on this framework, it is potential to implement SM to LVCSR [Tang *et al*. 2005] in current computation condition and this will be our focus of future work.

Acknowledgements

The authors would like to thank the anonymous reviewers and Mr. Ludwig for their useful comments.

References

Deng, L., M. Aksmanovic, X. Sun, and C. Wu, “Speech Recognition Using Hidden Markov Models with Polynomial Regression Functions as Non-stationary States,” *IEEE Trans. Speech Audio Processing*, 2, 1994, pp. 507-520

Deng, Y., T. Huang, and B. Xu, “Towards high performance continuous mandarin digit string recognition,” In *Proceeding of Int. Conf. on Spoken Language Processing*, 2000, Beijing, China, vol.3, 642-645.

Digalakis, V., M. Ostendorf, and J. Rohlicek, “Fast Algorithms for phone classification and recognition using Segment-based Models,” *IEEE Trans. Speech Audio Processing*, 40(12), 1992, pp 2885-2896.

Gish, H., K.Ng, and J. Rohlicek, “Secondary Processing using Speech Segments for an HMM Word Spotting System,” In *Proceeding of Int. Conf. on Spoken Language Processing*, 1992, Banff, Canada, pp. 17-20.

Glass, J., "A probabilistic framework for segment-based speech recognition," *Computer Speech and Language*, 17, 2003, pp. 137-152.

Hon, H.W., and K. Wang. “Combining Frame and Segment Based Models for Large Vocabulary Continuous Speech Recognition”, In *IEEE Workshop on Automatic Speech Recognition and Understanding*. 1999, Keystone, USA, pp. 221-224.

Lee S., and J. Glass, “Real-Time Probabilistic Segmentation for Segment-Based Speech Recognition,” In *Proceeding of Int. Conf. on Spoken Language Processing*, 1998, Sydney, Australia, pp. 1803-1806.

Ostendorf, M., and S. Roucos, "A Stochastic Segment Model for Phoneme--Based Continuous Speech Recognition," *IEEE Transactions on Acoustics, Speech and Signal Processing*, 4(12), 1989, pp. 1857-1869.

Ostendorf, M., V. Digalakis, and O. Kimball, “From HMM’s to Segment Models: A Unified View of Stochastic Modeling for Speech Recognition.” *IEEE Trans. Speech Audio Processing*, 4(5), 1996, pp. 360-378.

Rabiner, L., and B. H. Juang, *Fundamentals of speech recognition*, Prentice Hall, 1993.

Rabiner, L., J. Wilpon, and F. Soong, “High Performance Connected Digit Recognition Using Hidden Markov Models,” *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 37(8), 1989, pp. 1214-1225.

Tang, Y., H. Zhang, W. Liu, and B. Xu, "Coloring the Speech Utterance to Accelerate the SM based LVCSR Decoding", in *IEEE International Conference on Natural Language Processing and Knowledge Engineering*, 2005, Wuhan, China, pp. 121-126.

Computational Linguistics and Chinese Language Processing
Vol. 11, No. 1, March 2006, pp. 87-100

Voice Activity Detection Based on Auto-Correlation Function Using Wavelet Transform and Teager Energy Operator

Bing-Fei Wu* and Kun-Ching Wang+

Abstract

In this paper, a new robust wavelet-based voice activity detection (VAD) algorithm derived from the discrete wavelet transform (DWT) and Teager energy operation (TEO) processing is presented. We decompose the speech signal into four subbands by using the DWT. By means of the multi-resolution analysis property of the DWT, the voiced, unvoiced, and transient components of speech can be distinctly discriminated. In order to develop a robust feature parameter called the speech activity envelope (SAE), the TEO is then applied to the DWT coefficients of each subband. The periodicity of speech signal is further exploited by using the subband signal auto-correlation function (SSACF) for. Experimental results show that the proposed SAE feature parameter can extract the speech activity under poor SNR conditions and that it is also insensitive to variable-level of noise.

Keywords: Voice Activity Detection, Auto-Correlation, Wavelet, Teager Energy

1. Introduction

Voice activity detection (VAD) refers to the ability to distinguish speech from noise and is an integral part of a variety of speech communication systems, such as speech coding, speech recognition, hand-free telephony, and echo cancellation. In the GSM-based communication system, a VAD scheme is used to lengthen the battery power through discontinuous transmission when speech-pause is detected [Freeman *et al.* 1989]. Moreover, a VAD algorithm can be used under a variable bit rate of the speech coding system in order to control the average bit rate and the overall quality of speech coding [Kondoz *et al*. 1994]. Perviously,

* Department of Electrical and Control Engineering, National Chiao-Tung University, HsinChu, Taiwan
E-mail: bwu@cssp.cn.nctu.edu.tw
+ Information & Communications Research Laboratories, Industrial Technology Research Institute, HsinChu, Taiwan
E-mail: kunching@itri.org.tw

[Received May 13, 2005; Revised Oct. 8, 2005; Accepted Oct. 17, 2005]

Sohn *et al.* [Sohn *et al.* 1998] presented a VAD algorithm that adopts a novel noise spectrum adaptation by applying soft decision techniques. The decision rule is drawn from the generalized likelihood ratio test by assuming that the noise statistics are known a priori. Cho *et al.* [Cho *et al*. 2001] presented an improved version of the algorithm designed by Sohn. Specifically, Cho presented a smoothed likelihood ratio test to reduce the detection errors. Furthermore, Beritelli *et al.* [Beritelli *et al.* 1998] developed a fuzzy VAD using a pattern matching block consisting of a set of six fuzzy rules. Additionally, Nemer *et al.* [Nemer *et al.* 2001] designed a robust algorithm based on higher order statistics (HOS) in the residual domain of the linear prediction coding coefficients (LPC). Meanwhile, the International Telecommunication Union-Telecommunications Sector (ITU-T) designed G. 729B VAD [Benyassine *et al*. 1997], which consists of a set of metrics, including line spectral frequencies (LSF), low band energy, zero-crossing rate (ZCR), and full-band energy. However, the common feature parameters mentioned above are based on averages over windows of fixed length or are derived through analysis based on a uniform time-frequency resolution. For example, it is well known that speech signals contain many transient components and exhibit the non-stationary property. The classical Fourier Transform (FT) works well for wide sense stationary signals but fails in the case of non-stationary signals since it applies only uniform-resolution analysis. Conversely, if the multi-resolution analysis (MRA) property of DWT [Strang *et al.* 1996] is used, the classification of speech into voiced, unvoiced or transient components can be accomplished.

The periodic property is an inherent characteristic of speech signals and is commonly used to characterize speech. In this paper, the periodic properties of subband signals are exploited to accurately extract speech activity. In fact, voiced or vowel speech sounds have a stronger periodic property than unvoiced sounds and noise signals, and this property is concentrated in low frequency bands. Thus, we let the low frequency bands have high resolution in order to enhance the periodic property by decomposing only the low band in each level. Three-level wavelet decomposition is further divided into four non-uniform subbands. Consequently, the well-known "Auto-Correlaction Function (ACF)" is defined in the subband domain to evaluate the periodic intensity of each subband, and is denoted as the "Subband Signal Auto-Correlaction Function (SSACF)". Generally speaking, the existing methods for suppressing noise are almost all based on the frequency domain. However, these methods indeed waste too much computing power in on-line work. Considering computing complexity, the Teager energy operator (TEO), which is a powerful nonlinear operator and has been successfully used in various speech processing applications [Kaiser *et al.* 1990],[Bovik *et al.* 1993],[Jabloun *et al.* 1999] is applied to eliminate noise components from the wavelet coefficients in each subband priori to SSACF measurement. Consequently, to evaluate the periodic intensity of each subband signal, a Mean-Delta method [Ouzounov *et al*. 2004] is

applied in the envelope of each SSACF. First, the Delta SSACF, similar to the delta-cepstrum evaluation, is used to measure the local variation of each SSACF. Next, since the DSSACF is averaged over its length, the value of the Mean DSSACF (MDSSACF) can almost describe the amount of periodicity in each subband. Eventually, by only summing the values of the four MDSSACFs, we can apply a robust feature parameter, called the speech activity envelope (SAE) parameter. Experimental results show that the envelope of the SAE feature parameter can accurately indicate the boundary of speech activity under poor SNR conditions and that it is also insensitive to variable-level noise. In addition, the proposed wavelet-based VAD can be performed on-line.

This paper is organized as follows. Section 2 describes the proposed algorithm based on DWT and TEO. In addition, the proposed robust feature parameter is also discussed. Section 3 evaluates the performance of the proposed algorithm and compares it with that of other wavelet-based VAD algorithms and ITU-T G.729B VAD. Finally, Section 4 presents conclusions.

2. The Proposed Algorithm Based on DWT and TEO

In this section, each part for the proposed VAD algorithm is discussed in turn.

2.1 Discrete Wavelet Transform

The wavelet transform (WT) is based on time-frequency signal analysis. This wavelet analysis adopts a windowing technique with variable-sized regions. It allows the use of long time intervals when we want more precise low-frequency information, and shorter regions where we want high-frequency information. It is well known that speech signals contain many transient components and exhibit the non-stationary property. When we make use of the MRA property of the WT, better time-resolution is needed in the high frequency range to detect the rapid changing transient component of a signal, while better frequency resolution is needed in the low frequency range to track slowly time-varying formants more precisely. Through MRA analysis, the classification of speech into voiced, unvoiced or transient components can be accomplished. An efficient way to implement this DWT using filter banks was developed in 1988 by Mallat [Mallat 1989].

In Mallat's algorithm, the j-level approximations A_j and details D_j of the input signal are determined by using quadrature mirror filters (QMF). Figure 1 shows that the decomposed subband signals A and D are the approximation and detail parts of the input speech signal obtained by using the high-pass filter and low-pass filter, implemented with the Daubechies family wavelet, where the symbol $\downarrow 2$ denotes an operator of downsampling by 2. In fact, a voiced or vowel speech sound has more significant periodicity than an unvoiced sound on noise signal. Thus, the periodicity of a subband signal can be exploited to accurately

extract speech activity. In addition, the periodicity is almostly concentrated in low frequency bands, so we let the low frequency bands have high resolution in order to enhance the periodic property by decomposing only low bands in each level. Figure 2 employed the used structure of three-level wavelet decomposition. By using DWT, we can divide the speech signal into four non-uniform subbands. The wavelet decomposition structure can be used to obtain the most significant periodicity in the subband domain.

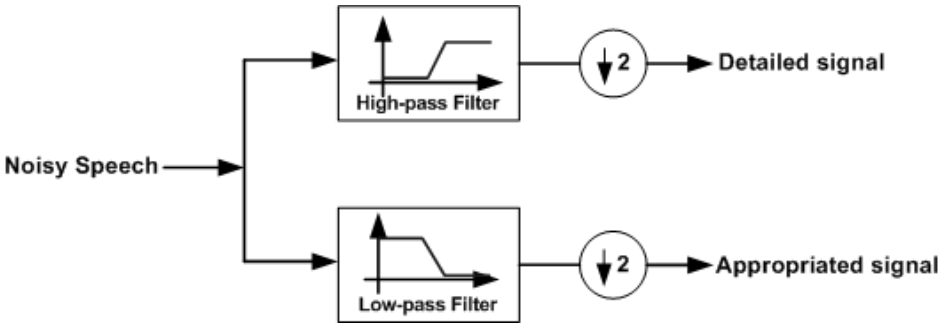


Figure 1. Discrete wavelet transform (DWT) using filter banks

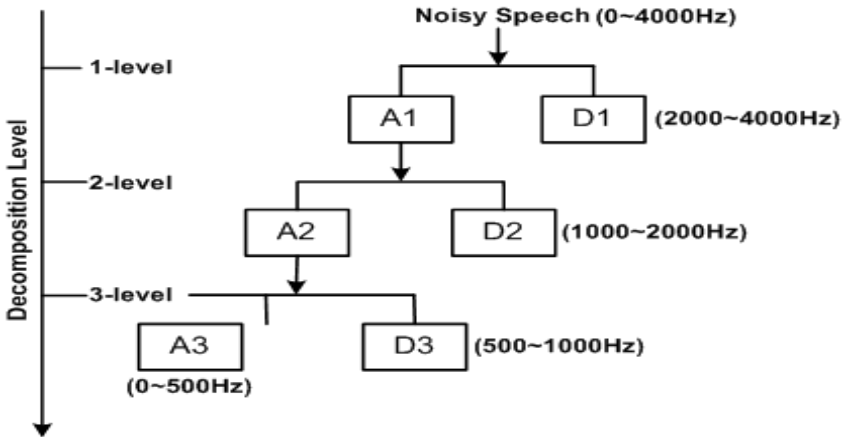


Figure 2. Structure of three-level wavelet decomposition

2.2 Teager Energy Operator

It has been observed that the TEO can enhance the discriminability between speech and noise and further suppress noise components from noisy speech signals [Jabloun *et al*. 1999]. Compared with the traditional noise suppression approach based on the frequency domain, the TEO based noise suppression can be more easily implemented through the time domain.

In continuous-time, the TEO is defined as

$$\psi_c[s(t)] = [\dot{s}(t)]^2 - s(t)\ddot{s}(t),$$

where $s(t)$ is a continuous-time signal and $\dot{s} = ds/dt$. In discrete-time, the TEO can be approximated by

$$\psi_d[s(n)] = s(n)^2 - s(n+1)s(n-1), \quad (1)$$

where $s(n)$ is a discrete-time signal.

Let us consider a speech signal $s(n)$ degraded by uncorrelated additive noise $u(n)$, the resulting signal is shown below:

$$y(n) = s(n) + u(n). \quad (2)$$

The Teager energy of the noisy speech signal $\psi_d[y(n)]$ is given by

$$\psi[y(n)] = \psi_d[s(n)] + \psi_d[u(n)] + 2\tilde{\psi}_d[s(n), u(n)], \quad (3)$$

where $\psi_d[s(n)]$ and $\psi_d[u(n)]$ are the Teager energy of the discrete speech signal and the additive noise, respectively. The subscript d means the "discrete." $\tilde{\psi}_d[s(n), u(n)]$ is the cross-ψ_d energy of $s(n)$ and $v(n)$, such that

$$\tilde{\psi}_d[s(n), u(n)] = s(n)u(n) - 0.5s(n-1)\cdot u(n+1) - 0.5s(n+1)\cdot u(n-1), \quad (4)$$

where the symbol $\cdot$ denotes the inner product. Since $s(n)$ and $u(n)$ are zero mean and independent, the expected value of the cross-ψ_d energy is zero. Thus, Eq.(5) can be derived from Eq.(3) as shown below:

$$E\{\psi[y(n)]\} = E\{\psi[s(n)]\} + E\{\psi[u(n)]\}. \quad (5)$$

Experimental results show that the Teager energy of the speech is much higher than that of the noise. Thus, compared with $E\{\psi_d[y(n)]\}$, $E\{\psi_d[u(n)]\}$ is negligible as shown by

$$E\{\psi_d[y(n)]\} \approx E\{\psi_d[s(n)]\}. \quad (6)$$

2.3 Subband Signal Auto-Correlation Function (SSACF)

The definition of the "Auto-Correlation Function (ACF)" used to measure the self-periodic intensity of subband signal sequences is shown below:

$$R(k) = \sum_{n=0}^{p-k} s(n)s(n+k), \quad k = 0,1,.....p, \quad (7)$$

where p is the length of ACF and k denotes the shift of the sample.

In this subsection, the ACF will be defined in the subband domain and called the "Subband Signal Auto-Correlation Function (SSACF)." It can be derived from the wavelet coefficients on each subband following TEO processing.

Figure 3 displays that the waveform of the normalized SSACFs ($R(0)=1$) of each subband, respectively. It is observed that the SSACF of voiced speech has more obvious peaks than that of unvoiced speech and white noise does. In addition, for unvoiced speech, the ACF has more intense periodicity than white noise does, especially in the $A3$ subband.

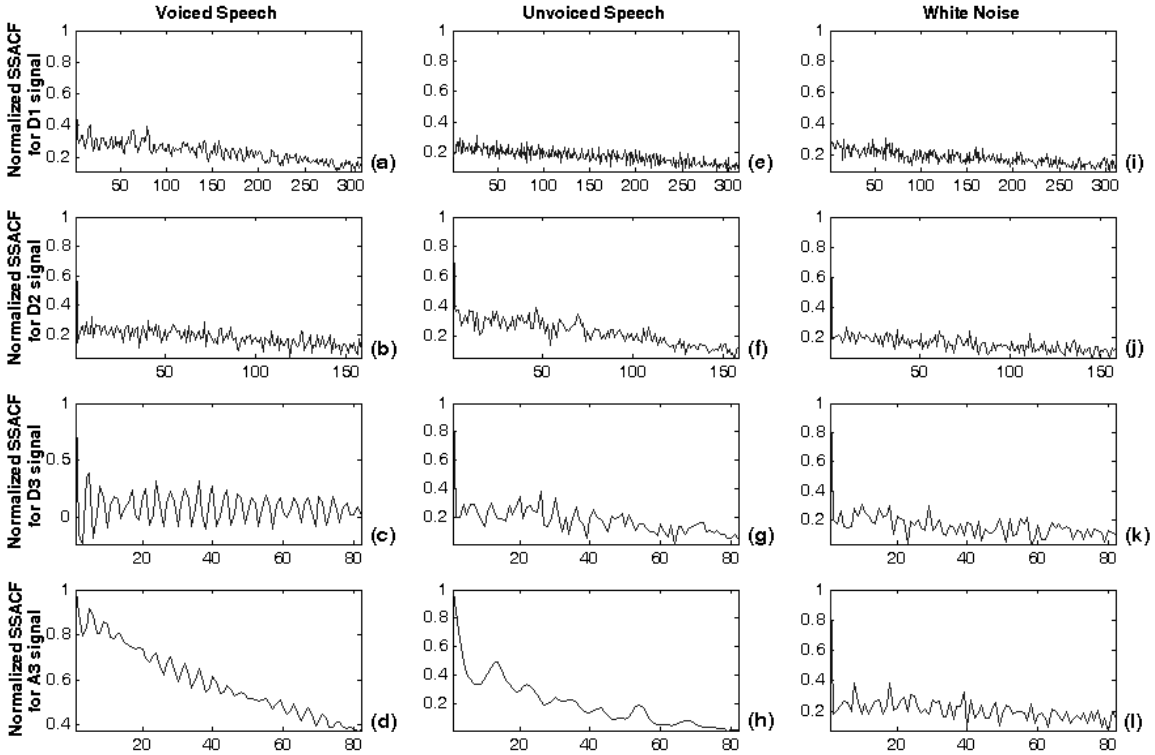


Figure 3. Examples of normalized SSACF for voiced speech, unvoiced speech and white noise

2.4 Mean of the absolute values of the DSSACF (MDSSACF)

To evaluate the periodic intensity of subband signals, a Mean-Delta method is applied here to each SSACF. First, a measure similar to delta cepstrum evaluation is used to estimate the periodic intensity of the SSACF, namely, the "Delta Subband Signal Auto-Correlation Function (DSSACF)," as shown below:

$$\dot{R}_M(k)=\frac{\sum_{m=-M}^{M} mR(k+m)}{\sum_{m=-M}^{M} m^2}, \tag{8}$$

where $\dot{R}_M$ is the DSSACF over an M -sample neighborhood.

For a particular frame, it is computed by using only the frame's SSACF (intra-frame processing), while the delta cepstrum is computed by using cepstrum coefficients from neighboring frames (inter-frame processing). It is observed that the DSSACF value is almost similar to the local variation over the SSACF.

Second, the delta of the SSACF is averaged over an M -sample neighborhood $\bar{R}_M$, where the mean of the absolute values of the DSSACF (MDSSACF) is given by

$$\bar{R}_M = \frac{1}{N_b} \sum_{k=0}^{N_b-1} \left| \dot{R}_M(k) \right|, \tag{9}$$

where N_b indicates the length of the subband signal.

Figure 4 shows that the SAE feature parameter is developed by summing the four MDSSACF values. Each subband can provide information for extracting voice activity precisely. It is found that the SAE feature parameter accurately indicates the boundary of speech activity under -5dB factory noise.

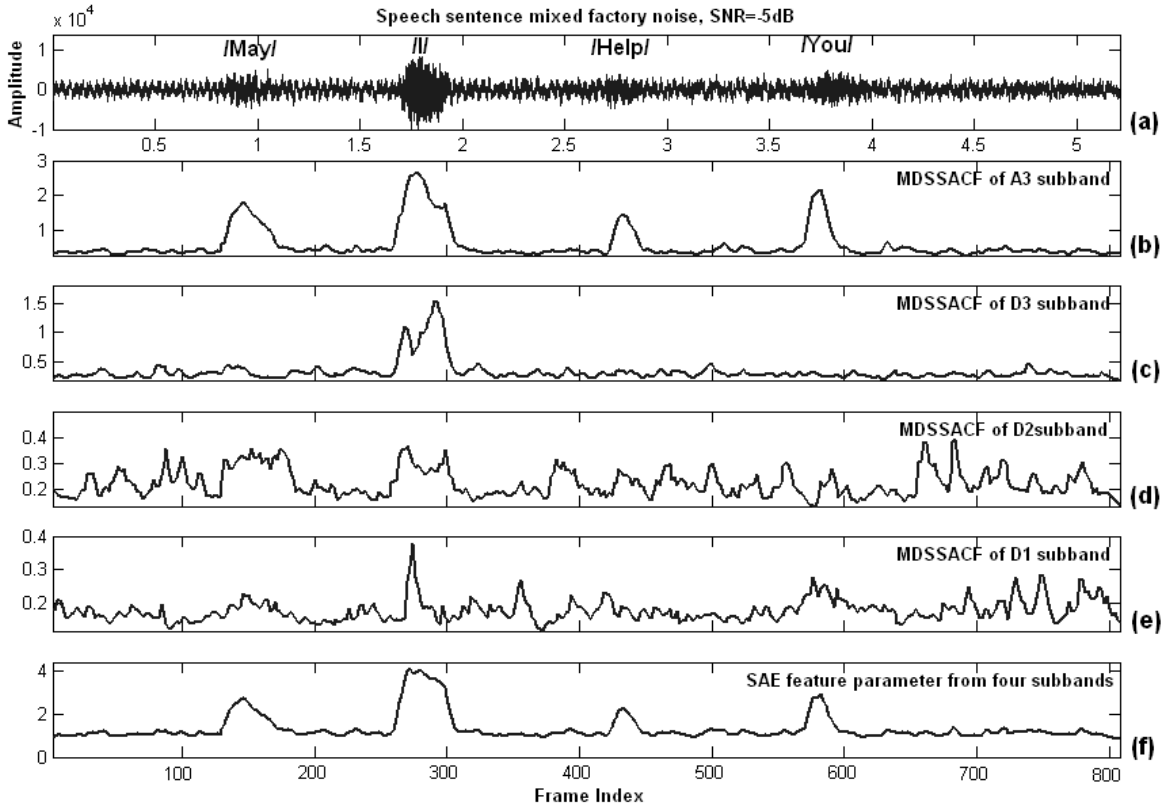


Figure 4. The development of the SAE feature parameter with and without band-decomposition

2.5 Block Diagram of the Proposed Wavelet-Based VAD

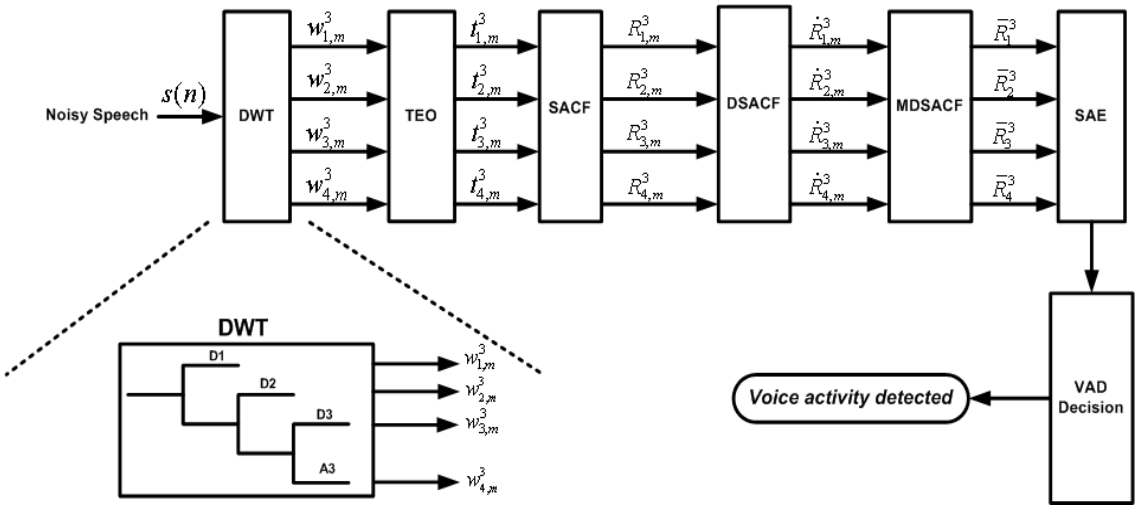


Figure 5. Block diagram of the proposed wavelet-based VAD

A block diagram of the proposed wavelet-based VAD algorithm is displayed in Figure 5. For a given level j, the wavelet transform decomposes the noisy speech signal into $j+1$ subbands corresponding to wavelet coefficients sets, $w_{k,n}^{j}$. In this case, for level $j=3$,

$$w_{k,m}^{3} = DWT\{s(n),3\}, \quad n=1....N,\ k=1....4, \tag{10}$$

where $w_{k,m}^{3}$ denotes the m^{th} coefficient of the k^{th} subband. N denotes the window length. The decomposed length of each subband is $N/2^{k}$. If $k=1$, $w_{1,m}^{3}$ corresponds to the $D1^{th}$ subband signal.

In TEO processing,

$$t_{k,m}^{3} = \psi_{d}[w_{k,m}^{3}],\ \text{k=1...4}. \tag{11}$$

The SSACF is derived from the Teager energy of noisy speech as follows:

$$R_{k,m}^{3} = R[t_{k,m}^{3}], \tag{12}$$

where $R[\cdot]$ denotes the auto-correlation operator.

Next, the DSSACF is given by

$$\dot{R}_{k,m}^{3} = \Delta[R_{k,m}^{3}], \tag{13}$$

where $\Delta[\cdot]$ denotes the Delta operator.

Then, the MDSSACF is obtained by

$$\bar{R}_k^3 = E[\dot{R}_{k,m}^3]. \tag{14}$$

where $E[\cdot]$ indicates the mean operator.

Finally, the SAE feature parameter is obtained by

$$SAE = \sum_{k=1}^{4} \bar{R}_k^3. \tag{15}$$

2.6 A VAD Decision Based on Adaptive Thresholding

In order to accurately determine the boundary of voice activity, the VAD decision is usually made through thresholding. To estimate the time-varying noise characteristics accurately, in this subsection, an adaptive threshold value is derived from the statistics of the SAE feature parameter during a noise-only frame, and the VAD decision process recursively updates the threshold by using the mean and variance of the values of the SAE parameters. We compute the initial noise mean and variance with the first five frames, assuming that the first five frames contain noise only. We then compute the thresholds for the speech and noise as follows [Gerven *et al.* 1997]:

$$T_s = \mu_n + \alpha_s \cdot \sigma_n, \tag{16}$$

$$T_n = \mu_n + \beta_n \cdot \sigma_n, \tag{17}$$

where T_s and T_n indicate the speech threshold and noise threshold, respectively. Similarly, μ_n and σ_n represent the mean and variance of the values of the SAE parameters, respectively.

The VAD decision rule is defined as follows:

$$\begin{aligned} &\text{if } (SAE(t) > T_s) \quad VAD(t)=1 \\ &\text{else if } (SAE(t) < T_n) \quad VAD(t)=0; \\ &\text{else } \quad VAD(t)=VAD(t-1). \end{aligned} \tag{18}$$

If the detection result shows a noise period, the mean and variance of the values of the SAE are updated by as follows:

$$\mu_n(t) = \gamma \cdot \mu_n(t-1) + (1-\gamma) \cdot SAE(t), \tag{19}$$

$$\sigma_n(t) = \sqrt{[SAE_{buffer}{}^2]_{mean} - [\mu_n(t)]^2}, \tag{20}$$

$$[SAE_{buffer}{}^2]_{mean}(t) = \gamma \cdot [SAE_{buffer}{}^2]_{mean}(t-1) + (1-\gamma) \cdot SAE(t)^2 . \quad (21)$$

Here, $[SAE_{buffer}{}^2]_{mean}(t-1)$ is a mean of the buffer of the SAE value during a noise-only frame. We then update the thresholds by using the updated mean and variance of the values of the SAE parameters. Figure 6 displays the VAD decision, based on the adaptive threshold strategy. It is clearly seen that the boundary of voice activity has been accurately extracted. The two thresholds are updated during voice-inactivity but not during voice-activity.

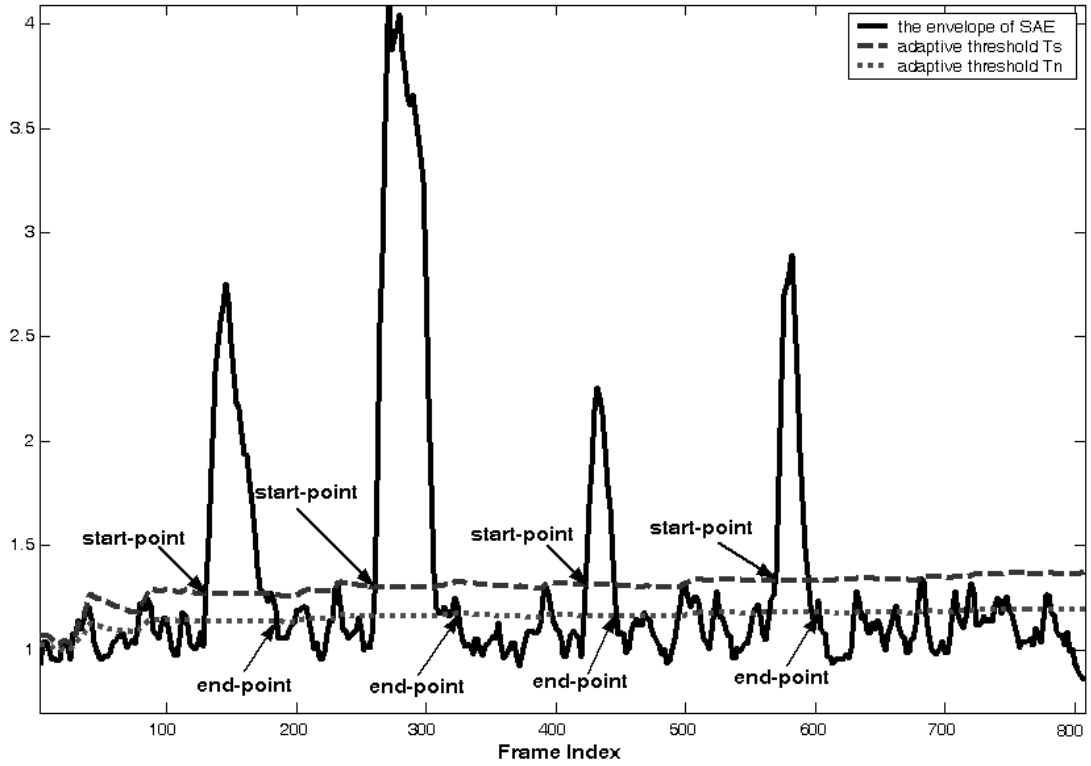


Figure 6. Adaptive thresholding strategy for extracting the boundary of voice activity

3. Simulation Results

The proposed wavelet-based VAD algorithm operates on a frame-by-frame basis (frame size = 256 samples/frame, overlapping size = 64 samples, $M = 8$, $\alpha_s = 5$, $\beta_n = -1$ and $\gamma = 0.95$). The results of speech activity detection were obtained under three kinds of background noise, which included white noise, car noise, and factory noise, taken from the Noisex-92 database [Varga *et al.* 1993]. The speech database contained 60 speech phrases (in

Mandarin and in English) spoken by 32 native speakers (22 males and 10 females), sampled at 8000 Hz and linearly quantized at 16 bits per sample. The two probabilities of correctly detecting speech frames, P_{cs}, and falsely detecting speech frames, P_f, were the ratio of the correct speech decision to the total number of hand-labeled speech frames and the ratio of the false speech decision or false noise decision to the total number of hand-labeled frames used to objectively measure performance of these three VADs.

Table 1 compares the performance of the proposed wavelet-based VAD, the wavelet-based VAD proposed by Chen *et al.* [Chen *et al*. 2002], and the ITU standard G.729B [Benyassine *et al*. 1997] under three types of noise and three specific SNR values: 30,10, and -5dB. From this table, it can be seen that in terms of the average correct and false speech detection probability, the proposed wavelet-based VAD is superior to Chen's VAD algorithm and G.729B VAD over all three SNRs under various types of noise. Table 2 shows the computing time of the three VAD algorithms, where Matlab was used on a Celeron 2.0G CPU PC to process 138 frames of a speech signal. It is found that the computing time consists of the time needed for feature extraction, and the voice activity decision process. The computing time of Chen's VAD was nearly twelve times longer than that of proposed VAD. We attribute the computing time of Chen's VAD to five-level wavelet decomposition. Its feature parameter is based on 17 critical-subbands, using the perceptual wavelet packet transform (PWPT). And after, wavelet reconstruction is required in Chen's approach. In our approach, however, we only divide four subbands using wavelet transform and do not waste extra computing time on wavelet reconstruction.

Table 1. Performance of the proposed wavelet-based approach, Chen's wavelet-based approach [9] and G.729B VAD

Noise Conditions		P_{cs} **(%)**			P_f **(%)**		
Type	**SNR(dB)**	**Proposed VAD**	**Chen's VAD**	**G.729B VAD**	**Proposed VAD**	**Chen's VAD**	**G.729B VAD**
Car Noise	***30***	99.1	97.3	92.1	6.2	6.9	7.3
	10	97.3	96.1	86.5	8.6	9.3	16.3
	-5	92.6	93.5	72.3	10.5	10.9	21.5
Factory Noise	***30***	96.9	97.2	96.9	7.6	10.3	9.1
	10	93.1	94.1	82.3	8.8	13.2	18.9
	-5	87.2	85.6	70.7	10.9	15.4	26.4
White Noise	***30***	99.1	97.2	98.4	1.3	1.9	2.0
	10	98.5	98.1	86.3	1.5	1.8	3.6
	-5	93.2	92.9	60.5	1.6	2.3	3.3
Average		**95.22**	**94.67**	**82.89**	**6.33**	**8**	**12.04**

Table 2. The computing time required by the three VAD algorithms

VAD type	Feature Extraction Processing	Voice Activity Decision
G.729B	0.048 s	0.023 s
Chen'VAD	4.126 s	0.098 s
Proposed VAD	0.23 s	0.12 s

Figure 7 shows the performance of the proposed VAD for an utterance produced continuously under variable-level noise. We decreased and increased the level of background noise and set the SNR value to 0 dB. Compared with the envelope of the VAS parameter, it is observed that the envelope of the SAE parameter was more robust against the variable noise-level and able to extract the exact boundary of the voice activity. This can be mainly attributed to the fact that the value of each MDSSACF depends on the amount of variation of the ACF, not on the energy level of the signal.

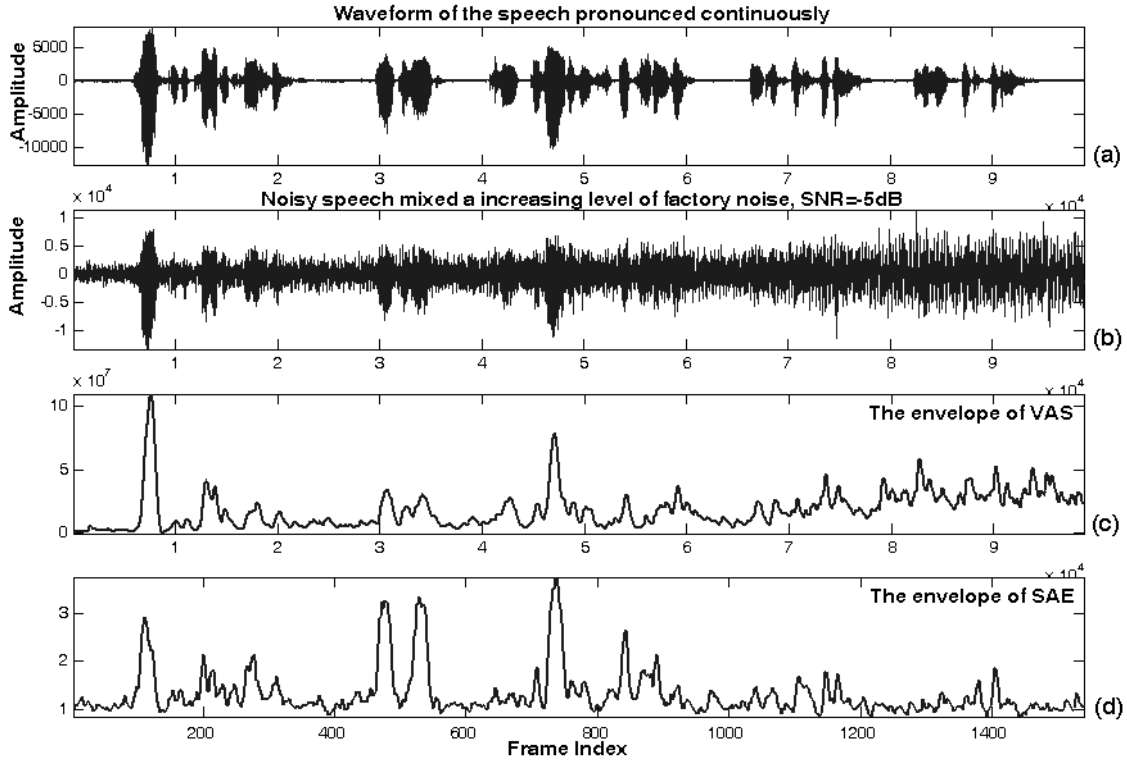


Figure 7. The effects of variable noise-level on the proposed SAE parameter and Chen's VAS parameter for a noisy speech sentence consisting of continuous words

4. Discussion

Compared with Chen's wavelet-based VAD, our experimental results shows that the proposed wavelet-based VAD algorithm is more suitable for on-line work. In terms of complexity, Chen's wavelet-based VAD algorithm [Chen *et al*. 2002] requires five-level wavelet decomposition to decompose the speech signal into 17 critical-subbands by using PWPT. In addition, it uses more extra computing time to complete wavelet reconstruction. In tests with non-stationary noise, it was found that each MDSSACF depends only on the amount of variation of the normalized ACF, not on the energy level of the signal, so the envelope of the proposed SAE feature parameter is insensitive to variable-level noise. Conversely, in Chen's wavelet-based method, the VAS feature parameter closely depends on the subband energy, so the achieved performance is poor under variable-level noise.

Acknowledgments

This work was supported by National Science Council of Taiwan under grant no. NSC 94-2213-E-009-066.

Reference

Benyassine, A., E. Shlomot, H. Y. Su, D. Massaloux, C. Lamblin, and J. P. Petit, "ITU-T Recommendation G.729 Annex B: a silence compression scheme for use with G.729 optimized for V.70 digital simultaneous voice and data applications," *IEEE Communications Magazine*, 35(9), 1997, pp.64-73.

Beritelli, F., S. Casale, and A. Cavallaro, "A robust voice activity detector for wireless communications using soft computing," *IEEE Journal on Selected Areas in Communications*, 16(9), 1998, pp.1818-1829.

Bovik, A. C., P. Maragos, and T. Quatieri, "AM-FM energy detection and separation in noise using multiband energy operators," *IEEE Transactions on Signal Processing*, 41(12), 1993, pp.3245-3265.

Chen, S.H., and J.F. Wang, "A Wavelet-based Voice Activity Detection Algorithm in Noisy Environments," *International Conference on 9th Electronics, Circuits and Systems*, 2002, pp.995-998.

Cho, Y. D., and A. Kondoz, "Analysis and improvement of a statistical model-based voice activity detector," *IEEE Signal Processing Letters*, 8(10), 2001, pp.276-278.

Freeman ,D. K., G. Cosier, C. B. Southcott, and I. Boyd, "The voice activity detector for the pan European digital cellular mobile telephone service," International Conference on *Acoustics, Speech, and Signal Processing*, 1989, pp.369-372.

Gerven , S. V., and F. Xie, "A comparative study of speech detection methods," In *Proceedings of Eurospeech*, 3, 1997, pp.1095-1098.

Jabloun, F., A. E. Cetin, and E. Erzin, "Teager energy based feature parameters for speech recognition in car noise," *IEEE Signal Processing Letters*, 6(10), 1999, pp.259-261.

Kaiser, J. F., "On a simple algorithm to calculate the 'energy' of a signal," *International Conference on Acoustics, Speech, and Signal Processing*, 1990, pp.381-384.

Kondoz, A. M., *Digital Speech Coding for Low Bit Rate Communications Systems*, John Wiley & Sons Ltd., 1994.

Mallat, S., "A theory for multiresolution signal decomposition: the wavelet representation," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 11(7), 1989, pp.674-693.

Nemer, E., R. Goubran and S. Mahmoud, "Robust voice activity detection using higher-order statistics in the LPC residual domain," *IEEE Transactions on Speech and Audio Processing*, 9(3), 2001, pp.217-231.

Ouzounov, A., "A Robust Feature for Speech Detection," *Cybernetics and Information Technologies*, 4(2), 2004, pp.3-14.

Sohn, J., and W. Sung, "A voice activity detector employing soft decision based noise spectrum adaptation," International Conference on *Acoustics, Speech, and Signal Processing*, 1998, pp.365-368.

Strang, G., and T. Nquyen, *Wavelet and Filter Banks*, Wellesley-Cambridge Press, 1996.

Varga, A., and H. J. M. Steeneken, "Assessment for automatic speech recognition: II. NOISEX-92: A database and an experiment to study the effect of additive noise on speech recognition systems," *Speech Communication*, 12, 1993, pp.247-251.