Speaker Cluster based GMM Tokenization for Speaker Recognition

Bin Ma, Donglai Zhu, Rong Tong and Haizhou Li

Institute for Infocomm Research
21 Heng Mui Keng Terrace, Singapore 119613
{mabin, dzhu, tongrong, hli}@i2r.a-star.edu.sg

ABSTRACT

We present a speaker recognition system with multiple GMM tokenizers as the front-end, and vector space modeling as the back-end classifier. GMM tokenizer captures the acoustic and phonetic characteristics of a speaker from the speech without the need of phonetic transcription. To enhance the speaker characteristics coverage and provide more discriminative information, a speaker clustering algorithm is proposed to build multiple GMM tokenizers that are arranged in parallel. For an input utterance, each of the tokenizers outputs a token sequence, which is then represented by a vector of n-gram probabilities. Multiple vectors are concatenated to form a composite vector. Finally the Support Vector Machine (SVM) is used as the back-end classifier of the composite vectors. We use the 2002 NIST Speaker Recognition Evaluation (SRE) corpus for training GMM tokenizers and background modeling, and evaluate on the 2001 NIST SRE corpus.
Index Terms: speaker recognition, speaker clustering, GMM tokenization

1. INTRODUCTION

Text-independent speaker recognition has made much progress in the past decade by using the conventional spectral/prosodic features, such as Gaussian Mixture Modeling (GMM) on amplitude spectrum based features [1], Support Vector Machine (SVM) on Shifted Delta Cepstral (SDC) [2], and Prosodic Dynamics Modeling [3]. In recent years, some tokenization methods with higher level information have been attracted great interests. These tokenization methods convert the speech into different sizes of tokens, such as words, phones and GMM tokens. For example, lexical features based on word n-grams has been studied in [4] for speaker recognition; Parallel Phone Recognition followed by Language Modeling (PPRLM) [5] has been extensively adopted in language and speaker recognition; Gaussian Mixture Model Tokenization [6] has been used with the tokens at the frame level for language identification.

Compared with phone level tokenization, GMM tokenizer captures another aspect of acoustic and phonetic characteristics of a speaker. Since it is constructed at the frame level, GMM tokenizer can have more tokens than phone tokenizer from limited speech data of one speaker in speaker recognition task. In this paper, we will study several aspects of GMM tokenizer method, including its token resolution, speaker characteristics coverage, front-end construction, backend classifier choice, and comparison with phone level tokenization.

Inspired by the finding in P-PRLM in language recognition where multiple single-language phone recognizers in the front-end can enhance the language coverage and improve the language recognition accuracy over single phone recognizer, we would like to explore multiple GMM tokenizers to improve speaker characteristics coverage and to provide more discriminative information for speaker recognition. In this paper, we propose using speaker cluster based GMM tokenizers to serve as the front-end of speaker recognition system.

GMM tokenizer converts the speech into a sequence of GMM token symbols which are the indexes of the Gaussian components scoring highest at every frame in the GMM computation. With these token symbols, we have two choices to construct the back-end speaker classifier. One is to use *n*-gram scoring with n-gram language modeling [7]. It estimates n-gram language model of GMM tokens from the training speech of a speaker and apply n-grams likelihood scoring on the evaluation speech data.

Another choice of the back-end classifier is to use the vector space modeling (VSM) method. The amount of speech data for each speaker in speaker recognition is quite limited and then the estimated n-gram language model might not be robust due to the data sparsity. To alleviate this problem, we can use SVM in the vector space [8] as the back-end classifier. The vector is created from the sequence of token symbols, and each dimension of the vector represents the statistics of the n-gram unit. We will make a comparison study with these two back-end classifiers.

The tokenization with phone recognizers and the tokenization with GMM tokens characterize speaker at phonetic and spectral levels. We will also carry out the experiments by using the phone recognizers of seven languages as the front-end. We will explore two backend setups, the n-gram scoring method and the VSM method.

This paper is organized as follows. In section 2, speaker recognition with speaker cluster based GMM tokenization and vector space modeling will be described in details. In section 3, speaker verification experiments will be established by using 2002 NIST SRE corpus for GMM tokenizers and background modeling, and using 2001 NIST SRE corpus as the evaluation data. Finally a discussion will be given in Section 4.

2. GMM TOKENIZATION WITH SVM FOR SPEAKER RECOGNITION

2.1 Vector Space Modeling

Vector space modeling (VSM) has become a standard tool in Information Retrieval systems since its introduction decades

ago [9]. It uses a vector to represent a text document. One of the advantages of this method is that it allows the discriminative training of classifier over the high dimensional document vectors. We can derive the distance between documents easily as long as the vector attributes are well defined characteristics of the documents. Each coordinate in the vector reflects the presence of the corresponding attribute.

In speaker recognition, we can regard a speech segment from a speaker as a spoken document, and regard the statistics of n-gram of the tokens, such as words, phones or GMM tokens, as the components in the spoken document vector. For a speech segment of Ω tokens $T = \{t_1, ..., t_\pi, ..., t_\Omega\}$, where $t_\pi \in \{w_1, w_2, ...w_J\}$ is one of the J tokens, we can establish a high-dimensional feature vector where all of its elements are expressed as the n-gram probability attributes

$$p(w_n \mid w_1, ...w_{n-1}) = p(t_\pi = w_n \mid t_{\pi-1} = w_1, ..., t_{\pi-n+1} = w_{n-1}) . \quad (1)$$

Its dimensionality is equal to the total number of n-gram patterns to highlight the overall behavior of a speaker as:

$$\bar{\lambda} = \left(p(w_1), ..., p(w_2 \mid w_1), ..., p(w_3 \mid w_1, w_2), ... \right) \quad (2)$$

The vector space modeling approach evaluates the goodness of fit, or score function, using vector-based distance, such as an inner product:

$$P\left(T \mid \lambda_{spk}\right) \propto \bar{\lambda}^T \cdot \omega_{spk} \quad (3)$$

where ω_{spk} is a speaker-dependent weight vector of equal dimension to $\bar{\lambda}$, with each component representing the contribution of its individual n-gram probability to the overall speaker score.

2.2 Speaker Cluster based GMM Tokenization

Among different token symbols (words, phones and GMM tokens), GMM tokenization provides an unsupervised training method to construct the tokenizer. The model training does not require the phonetic transcription of speech data. Since GMM tokenization works at frame level, it can provide more tokens than those tokenizers at word and phone levels, and then might alleviate the data sparsity problem in speaker recognition where there is only limited speech data available for the training and testing from one speaker.

P-PRLM [5] has proved to outperform single phone recognizer by using multiple single language phone recognizers in language identification. A set of parallel GMM tokenizers, each of which is trained for one of 12 languages [6], have also been used to enhance the language coverage and then to improve the language identification accuracy where n-gram scoring is used in the back-end classifier.

In speaker recognition, we assume that speakers can be grouped according to their spectral characteristics, for example, by speaker gender. By clustering all the speakers in the training set into several speaker clusters, we can partition the training space in a flexible data-driven manner. Each partition of speech data can then be used to train a GMM tokenizer. After the multiple GMM tokenizers are constructed, a speech segment passes through all these tokenizers to be converted into multiple feature vectors as shown in (1) and (2). These feature vectors will be concatenated into one single composite feature vector to represent the speech segment. The detailed implementation of the GMM tokenization is shown as follows:

Speaker Clustering for GMM Tokenizer Modeling

- Train a Universal Background Model (UBM) with *M* mixture of Gaussian components by using the speech data from *N* speakers;
- Convert the speech data of each speaker into a spoken document vector as shown in (1) and (2) by using the above obtained UBM;
- Cluster these *N* speakers into *C* speaker clusters with k-means algorithm by using the spoken document vectors, and split the whole training data set into *C* partitions accordingly.
- Train a cluster-dependent GMM by using the speech data from each of *C* speaker clusters. Acoustic adaptation algorithm can be used to obtain the GMMs based on the UBM.

Spoken Document Vector from Multi-GMM Tokenization

- For each of speech segments, *C* parallel recognitions are made by using the *C* cluster-dependent GMM models and *C* GMM token sequences are obtained;
- These *C* GMM token sequences are converted into *C* feature vectors as shown in (1) and (2). In this paper, we only use unigram and bigram patterns to represent the speaker characteristics;
- Concatenate the *C* feature vectors into one single composite feature vector to represent the speech segment.

With the multiple GMM token sequences from *C* GMM tokenizers, there are two choices to make the classification decision. One is to use the n-gram scoring to combine the scores from n-gram likelihood of GMM tokens [7]. Another choice is to use the vector space modeling approach [8] with the n-gram probability as the value of each dimension in the feature vector and with SVM as the classifier.

2.3 N-gram Scoring

For a single GMM tokenizer case, the n-gram likelihood scores from the hypothesized speaker model and the Universal Background Phone Model (UBMP) [7] can be combined to form the recognition score η_i by using the following log-likelihood ratio formula:

$$\eta_i = \frac{\sum_n (k(n)[S_i(n) - B(n)])}{\sum_n k(n)} \quad (4)$$

where *n* is the index of n-gram GMM tokens, $S_i(n)$ represents the log-likelihood score from the *i*-th hypothesized speaker model, $B(n)$ is the log-likelihood score from the UBPM, and $k(n)$ is the weighting function based on the number of occurrences of a particular n-gram GMM token.

The score fusion from multiple GMM tokenizers can be made by summing all the *C* scores from *C* GMM tokenizers as the following formula:

$$\gamma_i = \sum_{c=1}^{C} \alpha_c \eta_{c,i} \quad (5)$$

where α_c are the speaker cluster dependent weights.

2.4 Support Vector Machine (SVM)

The spoken document vector is high dimensional in nature where high order n-gram patterns are included. SVM is optimized on a structural risk minimization principle [10] and is a classifier of natural choice here because the feature vectors are sparse and do not follow a specific distribution. Because of its distribution-free property, SVM is suitable for designing vector-based classifiers.

Figure 1 shows the framework how a single high dimensional feature vector is constructed for a speech segment. Instead of using the score fusion from multiple SVMs as in [8], we create one feature vector from each of the GMM token sequences and concatenate these feature vectors into a single composite feature vector. A single SVM classifier is used to output the score in (3) for the speaker recognition.

By using a single composite feature vector, we can avoid the trouble of summing the scores of multiple SVMs while the score range from the SVMs might vary largely due to the different range of distances between the feature vectors and the separating hyperplanes. A single composite feature vector can help to solve this problem by using a unique decision hyperplane.

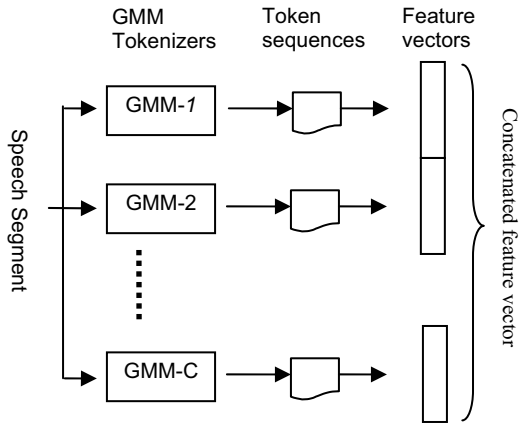

Figure 1: Speaker cluster based multiple GMM tokenizers

Log-likelihood ratio weighting scheme at each dimension of the feature vector as in [8] is adopted for term weighting of the SVM kernel construction. It makes a log of likelihood ratio normalization based on the n-gram likelihood values of background training data.

2.5 Phone Tokenization & GMM Tokenization

Both phone tokenization and GMM tokenization reflect the differences among speakers, in the dynamic realization of acoustic and phonetic characteristics, to pronounce the same sequence of sounds. In this way, they discriminate one speaker from another. Phone tokenization emphasizes on the phonetic-phonotactic information of the speaker while GMM tokenization puts emphasizes on the acoustic-spectral characteristics of the speaker. We examine both of them in this paper.

3. EXPERIMENTS

In the following experiments, we will examine several issues on GMM tokenization method, including the size of GMM Gaussian mixtures, speaker clustering for multiple GMM tokenizers, comparison of n-gram scoring method and vector space modeling method for the classifier design, and the comparison of phone tokenization and GMM tokenization.

3.1 Speech Corpora

We use the 2002 NIST SRE corpus for the training of the background model and speaker cluster based GMM tokenizers. We use the 2001 NIST SRE corpus for the evaluation. In 2001 NIST SRE corpus, the evaluation data is divided into evaluation training data and evaluation test data. The training data consists of 174 speech files that are two minutes long. The test data comprises 2,038 speech files of varying lengths not exceeding sixty seconds.

3.2 Gaussian Mixture Size and Speaker Clustering

The selection of Gaussian mixture size is a compromise between the amount of training data and resolution of GMM modeling ability. By using the 2002 NIST SRE corpus as the background modeling training data, one single GMM tokenizer is built. The performance comparison with the GMM tokenizers of different Gaussian mixture sizes on the 2001 NIST SRE data is shown in the Table 1. We find that 128 mixtures well describe the feature space.

Table 1: Performance comparison of different Gaussian mixture size with a single GMM tokenizer

Mixture Number	32	64	128	256
EER (%)	20.6	19.2	18.8	19.1

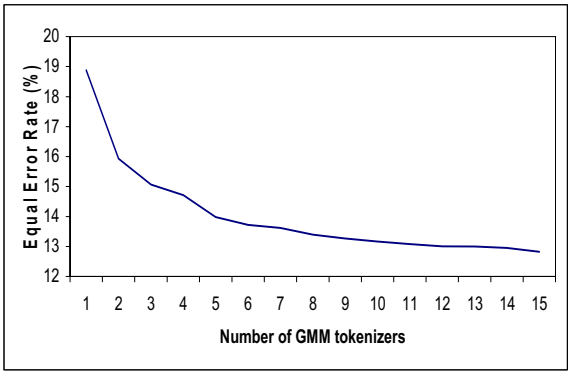

Figure 2: Performance comparison with different number of GMM tokenizers at Gaussian mixture size as 128

Figure 2 shows the performance as a function of the number of GMM tokenizers, each having 128 Gaussian mixtures. The speech data of each speaker in the 2002 NIST SRE is converted into a feature vector with unigram only. By using vector quantization algorithm, the speakers in the 2002 NIST SRE are clustered into certain number of clusters accordingly to the speaker characteristics. The speech data from each of the speaker clusters is used to train a GMM tokenizer. We can see from Figure 2 that, when more GMM tokenizers are available and then more detailed acoustic and phonetic characteristics are provided, the equal error rate (EER%) of speaker recognition is reduced.

3.3 Front-end Tokens and Back-end Classifiers

In this section, we will examine four combinations of front-ends and back-ends by the experiments. The two front-ends are multiple phone tokenizers and multiple GMM tokenizers. For the phone tokenization, we make the parallel phone recognizers (PPR) of seven languages, English, Korean, Mandarin, Japanese, Hindi, Spanish and German [11]. For the GMM tokenization, 15 parallel GMM tokenizers shown in Figure 2 with 128 Gaussian mixture components are used to create the GMM token sequences.

The two back-end classifiers are the n-gram scoring and the vector space modeling with SVM on the concatenated feature vector. Figure 3 shows the DET curves of the four combinations, phone-NGram, phone-VSM, GMM-NGram and GMM-VSM.

The combination of GMM tokenization and vector space modeling (GMM-VSM) achieves the best results in the 2001 NIST SRE corpus among the four systems.

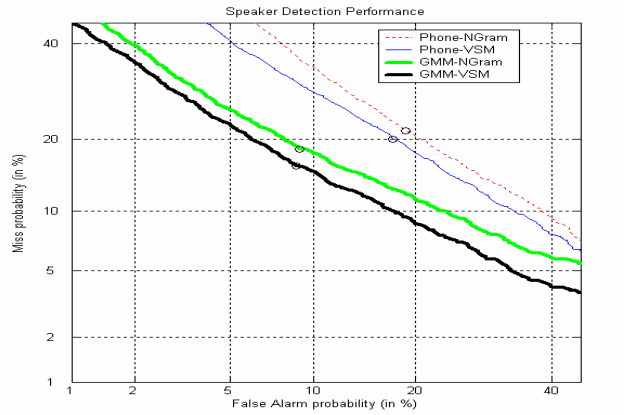

Figure 3: Performance comparison with four combinations of front-ends and back-ends.

4. DISCUSSION

In this paper, we present a speaker recognition system with speaker-clustering based GMM tokenization as the front-end, and with vector space modeling as the back-end classifier. GMM tokenizer does not need the transcribed speech data for model training, and it can provide more tokens at the frame level to alleviate the data sparsity problem in speaker recognition task. By using speaker cluster based GMM tokenization, we can find a flexible way to prepare multiple GMM tokenizers to provide better speaker characteristics coverage that discriminates speakers. By using the vector space modeling method as the back-end classifier, a speech segment is converted into a feature vector with the statistics of GMM token n-grams as the components. We concatenate multiple feature vectors into a single composite feature vector, and use a single SVM classifier to output the score for speaker recognition. This can avoid the trouble of combining the scores of multiple SVMs, which typically have different score ranges.

The tokenization method provides useful information for speaker recognition, but it only reflects one aspect of speaker information. The further benefit can be achieved by combining the tokenization method with other modeling methods on spectral/prosodic features.

5. REFERENCES

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

[2] Campbell, W. M., Campbell, J. P., Reynolds, D. A., Singer, E. and Torres-Carrasquillo, P. A., "Support Vector Machines for Speaker and Language Recognition," *Computer Speech and Language*, 20 (2006), pp. 210-229.

[3] Adami, A. G., Mihaescu, R., Reynolds, D. A. and Godfrey, J. J., "Modeling Prosodic Dynamics for Speaker Recognition," *Proc. ICASSP*, 2003.

[4] Doddington, G., "Speaker Recognition based on Idiolectal Differences between Speakers," *Proc. Eurospeech*, 2001.

[5] Zissman, M. A., "Comparison of Four Approaches to Automatic Language Identification of Telephone Speech," *IEEE Trans. on Speech and Audio Processing*, vol. 4, no. 1, 1996.

[6] Torres-Carrasquillo, P. A., Reynolds, D. A. and Deller, Jr., J. R., "Language Identification using Gaussian Mixture Model Tokenization," *Proc. ICASSP*, 2002.

[7] Andrews, W. D., Kohler, M. A. and Campbell, J. P., "Phonetic Speaker Recognition," *Proc. Eurospeech*, 2001.

[8] Campbell, W. M., Campbell, J. P., Reynolds, D. A., Jones, D. A. and Leek, T. R., "Phonetic Speaker Recognition with Support Vector Machines," *Proc. NIPS*, 2003.

[9] Salton, G, *The SMART Retrievl System*. Prentice-Hall Englewood Cliffs, NJ, 1971.

[10] Vapnik, V, *The Nature of Statistical Learning Theory*. Springer-Verlag, 1995.

[11] Tong, R., Ma, B., Zhu, D., Li, H. and Chng, E. S., "Integrating Acoustic, Prosodic and Phonotactic features for Spoken language identification", *Proc. ICASSP*, 2006.