A Method For On-Line Speaker Indexing Using Generic Reference Models

Soonil Kwon, Shrikanth Narayanan

Department of Electrical Engineering, Speech Analysis and Interpretation Lab,
and Integrated Media Systems Center
University of Southern California, CA, U.S.A.
soonilkw@usc.edu, shri@sipi.usc.edu, http://sail.usc.edu

Abstract

On-line Speaker indexing is useful for multimedia applications such as meeting or teleconference archiving and browsing. It sequentially detects the points where a speaker identity changes in a multi-speaker audio stream, and classifies each speaker segment. The main problem of on-line processing is that we can use only current and previous information in the data stream for any decisioning. To address this difficulty, we apply a pre-determined reference speaker-independent model set. This set can be useful for more accurate speaker modeling and clustering without actual training of target data speaker models. Once a speaker-independent model is selected from the reference set, it is adapted into a speaker-dependent model progressively. Experiments were performed with HUB-4 Broadcast News Evaluation English Test Material(1999) and Speaker Recognition Benchmark NIST Speech(1999). Results showed that our new technique gave 96.5% indexing accuracy on a telephone conversation data source and 84.3% accuracy on a broadcast news source.

1. Introduction

Automatic segmentation and classification of multispeaker audio data have been gaining considerable attention as multimedia communication/information systems are becoming an integral part of our daily life. For example, multimedia meetings and teleconferences are common but important events. However, it is impossible to attend all relevant meetings that are held all over the world although they are important. Multimedia meeting or teleconference browsers can be useful for getting meeting information remotely through the on-line or off-line systems [1] [2].

These applications commonly include a speaker indexing process that tags speaker-specific portions of data to pin point who is talking when [3]. Off-line speaker indexing can be used for record keeping, but it is not appropriate for real-time meeting or teleconferencing systems. In this paper, we propose an on-line method that picks out the speech segments from an audio stream and classifies them by speakers.

On-line speaker indexing method can only be sequentially executed. In other words, assuming streaming audio, we make any decision of indexing with only current and previous speech data. Moreover the indexing problem gets more difficult if there is no prior knowledge about the target speakers in the data including the number of speakers. Since the models of speakers are not available a priori for indexing, we need to build and update them on the fly. This implies a number of challenges. In general, under these circumstances, data are not enough to build a speaker model. Although a model can be roughly built, it is
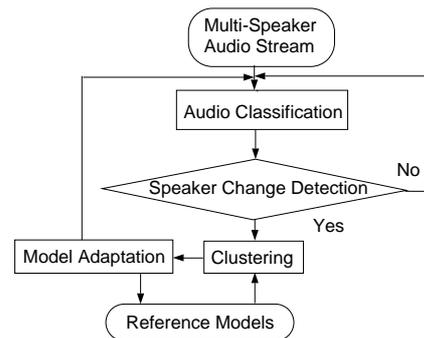

Figure 1: *Block diagram of on-line speaker indexing.*

apt to cause decision errors. We propose a generic reference model set to address the problem. This is built on the hypothesis that independent speech data corpus can help initialize a model set for the unsupervised indexing. The model set is pre-determined by training. Note that the speakers in the training data are independent of the testing data. In other words, the reference model set can be used for initializing/bootstrapping any speaker indexing process. This model set can be referred during speaker clustering with the test data. After clustering, a selected model is continually adapted with the test data that are used for clustering [Fig. 1].

For clustering, the size of analysis frame is fixed. Large frame size helps toward a correct indexing decision, but it is apt to miss speaker changes. To solve this problem, small indexing frame size is used in conjunction with a robust speaker change detection process to improve the precision. We use the generalized likelihood ratio(GLR) Test for speaker change detection [3]. Though the GLR Test can be unstable for small amounts of analysis data, clustering can help compensate for this instability.

We experimented with a part of HUB-4 Broadcast News Evaluation English Test Material(1999) and Speaker Recognition Benchmark NIST Speech(1999). The experimental results show that our on-line speaker indexing can achieve a comparable recognition rate with state of the art off-line system [3].

This paper is organized as follows: section 2 explains our on-line speaker indexing system in detail; section 3 and section 4 describe our experiment data, method, and results respectively; our conclusion and future plan are described in section 5.

2. On-Line speaker indexing

Several efforts have been reported on speaker indexing. One of these is about self-organized modeling by Nishida and Ariki

for on-line speaker indexing [6]. This means that speaker indexing and model construction can be performed sequentially without storing all the testing data in advance. But the problem is that sequentially constructed models can not represent speakers well due to small initial amount of data. Since the training is unsupervised, this problem also potentially leads to continual error propagation. We try to solve this critical drawback using an alternative method.

The block diagram of on-line speaker indexing process is shown in Fig. 1. Although not shown explicitly in Fig. 1, the audio classification step assumes appropriate front-end processing. The audio samples come into the system and are classified into different speech and background noise types. Only the detected speech data are used for the next stage, speaker change detection. In this next step, the system detects the end of speech data of the current active speaker. When it finds the boundary, the whole data between the speaker change points are used for clustering. After clustering, a selected model for the current speaker is adapted into the current speaker dependent model. The adapted model is moved into the reference model set, and the original model before adaptation is deleted. Next audio samples after the boundary of the current speaker come into system, and the system repeats the previous steps until all data are consumed.

2.1. Audio Classification

Generally audio data can be categorized into four broad classes: speech, music, environmental noise, silence. In speaker indexing, we only need speech/nonspeech discrimination. When there is background noise or music, it is likely to be overlapped with speech. Corrupted speech is not easily discriminated from noise. Since it is critical that we should not lose any speech data, the focus of the classification is to minimize false rejection even at the cost of false acceptance. Usually, for speech/nonspeech discrimination, zero-crossing rate and short-time energy are used [4]. It is well known that speech has a higher level of variation in zero crossing rate.

2.2. Speaker change detection using Localized Search Algorithm(LSA)

After audio classification, speech data are ready for speaker change detection. In this step, the system sequentially detects whether a speaker changes in the middle of speech analysis frame without any knowledge about the identity and number of speakers.

For this detection, we use a window which consists of two segments. A short segment cannot reflect speaker characteristics well. It has been experimentally found that segments should be at least longer than 2 seconds for robust recognition [1]. Each segment we considered was 2 seconds long, and the analysis window is 4 seconds long. The window shifts by 1 second. Within each window, the two segments are compared to detect whether they contain speech data from the same speaker or not. Although windows are overlapped by 3 seconds, it is not enough for a fine detection. We cannot detect a changing point within 1 second duration. Smaller window shifts (eg. 0.2 sec) could lead to finer resolution [3]. But computational complexity severely increases with the number of segments.

To solve this problem, we use a Localized Search Algorithm(LSA). Fig. 2 shows how this algorithm works. The algorithm is a compromise method. Firstly, the window shifts by 1 second with 1 second overlapping. When a latent change point is found, the Localized Search Algorithm starts running which
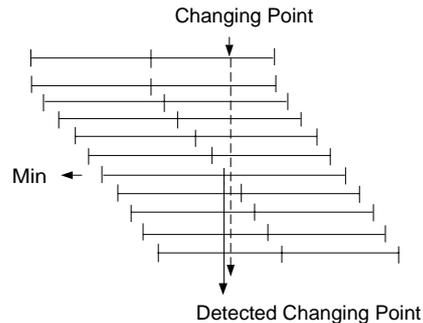

Figure 2: *Localized Search Algorithm (LSA) by using shifting windows.*

means that the window shifts by 0.2 second [3]. Two segments within the window are compared using the Generalized Likelihood Ratio(GLR) Test. Suppose there are two feature vector sets, X_1 and X_2, coming from each segment [3] [9]. Each segment contains speech from a single speaker. Hypothesis, H_0, is that the speakers in two segments are same, and H_1 is that the speakers are different. Let $L(X_1; \lambda_1)$ and $L(X_2; \lambda_2)$ be the likelihood of X_1 and X_2 where λ_1 and λ_2 represent model parameters that maximize each likelihood. Similarly let X be the union of X_1 and X_2. $L(X; \lambda_{1+2})$ is the maximum likelihood estimate for X. Then

$$GLR = \frac{L(X; \lambda_{1+2})}{L(X_1; \lambda_1)L(X_2; \lambda_2)} \qquad (1)$$

We apply thresholding on GLR to determine the latent changing point. When two segments represent the same speaker, GLR value goes up to 1, and, otherwise, it falls to zero. When GLR falls below the threshold, the second segment of current window includes the latent changing point. Then LSA is applied for a fine search. There are 10 candidates for the window which includes the real changing point. We regard the starting point of the second segment in the window that has the minimum GLR as the true changing point.

Speaker change detection step is important for the next step, speaker clustering. We can get speech data that is longer than 2 seconds. It is very helpful for better speaker clustering, because more speech data usually help representing a speaker better. However this step cannot be ensured to be perfect. If we falsely detect a speaker changing point, we can compensate for the error through the speaker clustering step. However, if we skip the real changing point, the clustering step cannot recover it. For these reasons, we tightly detect changing points to avoid the skip, although some points can be wrongly detected as changing points.

2.3. Reference models for initialization

When the on-line process starts, there is no prior knowledge of speakers. Therefore, it is very difficult to build a speaker model with just a few segments of incoming data. Only the data seen thus far can be used for modeling due to the characteristics of the on-line process. Such models that are roughly built can cause severe clustering errors. To get over this difficulty, we use a reference model set which is predetermined before on-line processing [7] [8]. The idea is similar to what has been proposed for robust training in Automatic Speech Recognition(ASR). We build generic models of speakers which are independent of test set speakers with the assumption that some

speakers of the reference set are acoustically close to the test speaker [11]. Although we do not know the exact number of speakers, we assume that the number is finite. With this assumption, an initial set of models (eg. 16) is built through training with data not directly related to the test condition. This reference model set can make it possible for on-line system to run without training.

2.4. Clustering and model adaptation

In this step, the segments obtained from the speaker change detection are indexed in terms of speakers, and then the corresponding models are adapted with the new data. For clustering, we use speaker models from our predetermined reference model set. "Speaker independent" models are used for clustering, and selected models are adapted to speaker dependent models. The likelihood of a speaker segment is calculated using the reference model set, and a model with maximum likelihood is selected. Then the selected model is adapted by Maximum a Posteriori(MAP) scheme. As the amount of data increases towards infinity, the MAP estimate converges to the ML estimate [10]. The MAP adaptation on a Gaussian Mixture Model(GMM) is straightforward [5]. Given the adaptation vectors $X = \{x_1, x_2, ..., x_T\}$, we compute the probability, $Pr(i|x_t)$:

$$Pr(i|x_t) = \frac{w_i p_i(x_t)}{\sum_{l=1}^{M} w_l p_l(x_t)} \qquad (2)$$

where w_i is the weight of each mixture of GMM, and p_i is the probability of input, x_t, in each mixture. M is the number of mixtures. In this system, means, $\hat{\mu}$, and weights, $\hat{w}$, of GMM are updated as:

$$\hat{\mu}_i = \alpha_i^m E_i(x) + (1 - \alpha_i^m)\mu_i \qquad (3)$$

$$\hat{w}_i = [\alpha_i^p n_i / T + (1 - \alpha_i^p)w_i]\gamma \qquad (4)$$

where γ is a scale factor. α_i^m and α_i^p are data-dependent adaptation coefficients which are defined as:

$$\alpha_i^p = \frac{n_i}{n_i + r_\rho} \qquad (5)$$

where r_ρ is the fixed relevance factor. n_i are the sufficient statistics of mixtures, and $E_i(x)$ are the re-estimation of mixtures which are defined as:

$$n_i = \sum_{t=1}^{T} Pr(i|x_t) \qquad (6)$$

$$E_i(x) = \frac{1}{n_i} \sum_{t=1}^{T} Pr(i|x_t)x_t \qquad (7)$$

We assume that speaker models in the reference set are independent of the testing speech data. From this assumption, we can expect that the initial adaptation/learning rate to the "true" speaker data should be rapid.

3. Experiments

We used two audio data sources: HUB-4 Broadcast News Evaluation English Test Material(1999) and Speaker Recognition Benchmark NIST Speech(1999). For the reference model set, we used 16 speakers (8 male speakers and 8 female speakers) who were randomly selected from the training data in Speaker Recognition Benchmark NIST Speech(1999). Since this corpus

consists of telephone conversations, there is no significant background noise to adversely affect recognition. For each speaker model training, about one minute of speech data were used. We tested our on-line speaker indexing system also using independent portions of these two data corpora. Test 1 was executed with about 600seconds audio data from the Speaker Recognition Benchmark NIST Speech(1999), and test 2 with 4200seconds audio data from HUB-4 Broadcast News Evaluation English Test Material(1999). Broadcast news data contain speech, music, and background noise. Since long silences have a bad effect for speaker recognition, we eliminated silence which is longer than 100msec and lower than -40dB. After the audio classification and silence elimination, we got 488seconds audio data for the first test and 3336seconds audio data for the second test.

Both these experimental data are sampled at different rates; hence we adjusted both of them to be sampled at 8000Hz. As feature vectors, we used 33 channels, 32 dimensional Mel Cepstrum vectors. We also used 30msec Hamming window that is shifted by 10msec. Speaker models were Gaussian Mixture Models(GMM) with 8 mixtures.

4. Results and discussion

Sequential speech data extracted from the input audio stream were chopped into segments by speaker changes (using LSA, Sec. 2.2), and the segments were classified through the reference model set (Sec. 2.3, 2.4). The result of our experiment is shown as Table 1. Total length of speech in the table refers to the temporal length of the input audio stream filtered through audio classification step. Total length of mismatch means the temporal length of the period that an indexed speaker is not a real speaker. The result of test 1 was much better than that of test 2. There are a couple of reasons for this difference. One of them is that training data and test data were from the same corpus, Speaker Recognition Benchmark NIST Speech(1999), although speakers for training are different from speakers for test. But HUB-4 Broadcast News Evaluation English Test Material(1999) was used for test 2. The other reason is that the NIST testing data are not as much corrupted by background noises as the second testing data; moreover, the speaker changing rate is lower in test 1 data than in test 2 data. When speaker changing rate is high, each speaker speaks for too short a time to recognize an identity robustly.

Table 1: *Speaker indexing error rate (Margin of Errors = 0.1 sec) (Error rate = Total length of mismatch / Total length of speech)*

	Test 1	Test 2
Total length of speech	488.02 sec	3,335.79 sec
Total length of mismatch	16.95 sec	525.17 sec
Error rate	3.47%	15.74%

Table 2: *Distribution of contiguous errors*

Errors $\leq$	0.5 sec	1 sec	2 sec	4 sec	8 sec
Test 1	59%	94%	100%	-	-
Test 2	36%	42%	48%	67%	81%

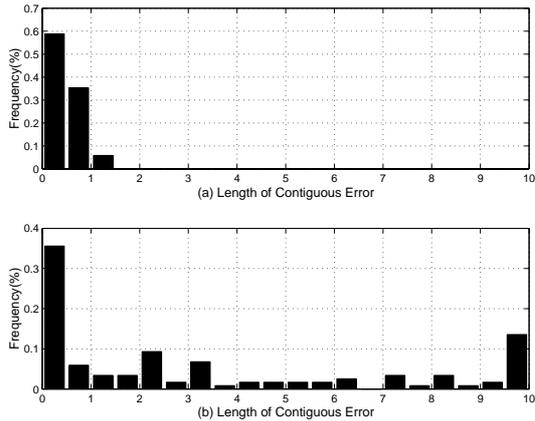The contiguous speaker segmentation/indexing errors are

Figure 3: *Histogram of contiguous errors(sec): (a)Test 1, (b)Test 2.*

much more relevant to us. The distribution of contiguous error length is statistically meaningful. The frequency of errors may be an important factor for analysis, but it is more important to see how long each error frame lasts. Fig. 3 shows that test 1 had a more ideal distribution. Over half of the contiguous errors were distributed under 500 msec, and all of them were under 1.5 sec. But in test 2, about a third of errors were under 500 msec, and about 10 % of errors were over 10 seconds due to some error propagation. Table 2 shows the distribution in detail. When we see both Table 1 and Table 2, each length of 81% out of the 15% errors in test2 is shorter than 8 seconds, and each length of 94% out of 3% errors is shorter than 1 second.

5. Conclusions

We presented a novel method for on-line speaker indexing. For addressing the unsupervised on-line process without any prior speaker information, a generic reference model set was inserted into the general speaker indexing algorithm. This set helped the on-line speaker indexing system to overcome the difficulty similar to that due to the lack of data for building speaker specific models in ASR. This modeling resulted in the better performance of speaker clustering which was our aim. Yet another requirement for better clustering is accurate speaker change detection. When speaker change detection is accurate, the performance of clustering gets better.

On-line speaker indexing has not been investigated widely. This paper represents a step toward achieving on-line speaker indexing. We used telephone conversation data and broadcast news data to evaluate the performance of our algorithm. Total error rates were 3.47% in test 1 and 15.74% in test 2. These errors have the distributions in the concept of contiguousness [Fig. 3 and Table 2].

There are three topics worth consideration to further improve the overall performance of on-line speaker indexing: devising optimal strategies for building the reference model set, robustly detecting speaker changes, and adapting speaker models. In this paper, we just used the speech data of randomly selected speakers for the reference model set. This method can not guarantee the optimal distribution of speaker models. Some of models can be severely overlapped, and some are apart, even if this formation can be thought to be natural. We also need to know which number of speaker models in the reference mod-

els set is optimal. We will need more experiments with more optimal strategies in the distribution of speaker models. Another important step, speaker change detection, was performed with GLR. This method is good, but has room for improvement using other statistical approaches. For model adaptation, we used MAP. This is good for controlling the speed of adaptation, but better adaptation methods can be explored to optimize both speed and accuracy.

7. References

[1] Kwon, S. and Narayanan, S., "Speaker Change Detection Using a New Weighted Distance Measure ", International Conference on Spoken Language Processing, vol. 4, p.2537-2540, 2002.

[2] Yang, J., Zhu, X., Gross, R., Kominek, J., Pan, Y., and Waibel, A., "Multimodal People ID for a Multimedia Meeting Browser", Proceedings of 7th ACM International Conference on Multimedia, Part 1, p.159-168, 1999.

[3] Rosenberg, A., Gorin, A. , and Parthasarathy S., "Unsupervised Speaker Segmentation of Telephone Conversations", International Conference on Spoken Language Processing, vol. 1, p.565-568, 2002.

[4] Lu, L., Zhang,H. -J., and Jiang, H., "Content Analysis for Audio Classification and Segmemtation", IEEE Trans. on Speech and Audio Processing, Vol. 10, p.504-516, No. 7, 2002.

[5] Liu, M., Chang, E., and Dai, B. -Q., "Hierarchical Gaussian Mixture Model for Speaker Verification", International Conference on Spoken Language Processing, vol. 2, p.1353-1356, 2002.

[6] Nishida, M. and Ariki, Y., "Speaker Indexing for News Articles, Debates, and Drama in Broadcasted TV Programs", IEEE International Conference on Multimedia Computing and systems, vol. 2, p.466-471, 1999.

[7] Siohan, O., Lee, C. -H., and Surendran, A., "Background Model Design for Flexible and Portable Speaker Verification Systems", in Proc. ICASSP, vol. 2, p.825-828, 1999.

[8] Padmanabhan, M., Bahl, L. R., Nahamoo, D., and Picheny, M. A., "Speaker Clustering and Transformation for Speaker Adaptation in Speech Recognition Systems", IEEE Transactions on Acoustics, Speech, and Signal Processing, Vol. 6, No. 1, pp. 71-77, 1998.

[9] Solomonoff, A., Mielke, A., Schnidt, M., and Gish, H., "Clustering Speakers by Their Voices", Proc. IEEE International Conference on Acoustics, Speech, and Signal Processing, vol. 2, p.12-15, 1998.

[10] Woodland, P. C., "Speaker Adaptation: Techniques and Challenges", in Proc. IEEE Workshop Automatic Speech Recognition and Understanding, Keystone, Colorado, Dec., p.85-90, 1999.

[11] Wu, J. and Chang, E., "Cohorts Based Custom Models for Rapid Speaker and Dialect Adaptation", Eurospeech2001, pp. 1261-1264, 2001.