EFFECTIVE METRIC-BASED SPEAKER SEGMENTATION IN THE FREQUENCY DOMAIN

Christoph Boehm and Franz Pernkopf

Signal Processing and Speech Communication Laboratory
Graz University of Technology, Austria
christoph.boehm@tugraz.at, pernkopf@tugraz.at

ABSTRACT

In this paper, we present an approach, called FREQDIST, for speaker segmentation based on a distance measurement applied in the frequency domain. To enhance the detection performance, the spectrum is reweighted using normalization techniques. Additionally, noise-like (i.e. flat) spectra are removed based on the entropy. Experiments using the TIMIT database [1] and Westdeutscher Rundfunk broadcast data show that our segmentation approach yields a good performance compared to the DISTBIC algorithm [2]. In particular, for the TIMIT data our algorithm reaches a false alarm rate (FAR) less than half of the value of the DISTBIC algorithm and a missed detection rate (MDR) of 7.0% instead of 13.1%.

Index Terms— Speaker turn detection, DISTBIC, FREQDIST.

1. INTRODUCTION

Speaker segmentation in audio data is a common task in today's speech processing. In fact, it is rather a preparatory step preceding algorithms that need single-speaker segments, e.g. to sort different speakers. Nevertheless, speaker segmentation is not a trivial problem especially if no a priori information is given for the speakers.

In general, speaker segmentation algorithms can be divided into three main approaches [3]: Silence-based methods, model-based methods, and metric-based methods. *Silence-based* methods detect a speaker turn (i.e. change of the speaker) if a silent area is present in the speech signal. However, silent segments do not imply a speaker change which results in a degraded performance. *Model-based* methods classify the data stream with the help of models, e.g. gaussian mixture models. Unfortunately, these models have to be trained before applying. Often no training data is available or at least not for all speakers. For these reasons, *metric-based* methods are widely-used [3, 4]. Those approaches measure distances between extracted features. The features of two adjacent windows which are moved over the audio stream are compared and classified as speaker turns or non-speaker turns. Many metric-based speaker segmentation approaches are based on cepstral domain features.

In this paper, a metric-based approach, called FREQDIST, is presented using frequency domain features. These features are derived by reweighting the spectrogram using normalization techniques. FREQDIST is based on the assumption that the high frequency spectral components stay rather constant as long as no speaker turn occurs [5]. The results of FREQDIST are compared to the DISTBIC algorithm introduced in [2] using the TIMIT database [1] and indexed Westdeutscher Rundfunk broadcast data.

The paper is organized as follows: Section 2 introduces our approach. The experimental setup is shown in section 3. The results are presented and discussed in section 4 and section 5. Finally, section 6 concludes the paper.

2. FREQDIST ALGORITHM

Similar to other metric-based speaker segmentation approaches, the proposed method consists of the following parts: Speaker specific properties are extracted from the input signal (*feature extraction*). Then, the similarity between neighboring windows of these features is determined (*distance measurement*) and finally, those segments with high distance values are detected and declared to be speaker turns (*classification*).

2.1. Feature Extraction

The features used for speaker segmentation are essential for its performance. They should carry information that enables the segmentation approach to distinguish between different speakers. To be able to correctly detect speaker turns, the features should vary as little as possible as long as no speaker turn occurs. And the other way round, if a speaker turn occurs, the features should vary as much as possible.

A good basis for feature extraction are the speech formants. They carry information about the current shape of the vocal tract. Since every speaker's vocal tract has a unique shape, the formants are speaker dependent and thus should provide sufficient information to detect a speaker change.

Formants vary as the vocal tract of the speaker changes. Vowels have characteristic formant frequencies. The first, the second, and sometimes even the third formant frequency (the lowest resonance frequency is the first formant frequency) do not change much for different speakers but they do change depending on the articulated vowel [5]. Thus the first two to three formants are good features for speech recognition as they characterize the spoken phone but they might not be suitable for speaker segmentation. The same assumptions are true for gliding sounds and liquids. Unfortunately, the Mel frequency cepstral coefficients (MFCC) that are often used in speaker segmentation are computed considering all available spectral components - including the "speaker-independent" formants.

For this reason, the suggested segmentation algorithm performs a reweighting of the spectrum by normalization techniques to emphasize high frequency components. The fourth (sometimes the third) and higher formants are assumed to be more speaker-specific, i.e., they do not change much as long as the speaker does not change [5]. This property seems to come from the length differences of the vocal tract between speakers which influence mostly the higher order formants. Furthermore, the higher order formant frequencies provide a high interspeaker variability which supports the detection of speaker turns. Hence, these formants provide a good basis for features in speaker segmentation (see [6]). But they require access to the spectral components of 3 kHz and above. The speech signal has to be sampled with rates of about 16 kHz and higher to extract

features from 3 to 8 kHz.

To extract these features, the speech signal is transformed into the frequency domain using the short time Fourier transform: A window of 20 ms is moved stepwise (10 ms) over the signal. For each step the discrete Fourier transform $\mathbf{X}_i[k]$ of the windowed signal (here, a Hamming window is used) is computed, where i is the time index and k the frequency bin.

We suggest two normalization techniques to reweight the spectrum: In general, most of the energy of speech signals is contained in the low frequency part. To remove this general correlation between frequency and energy, all frequency bins of the spectrogram are normalized over time (i.e. the mean magnitude of the frequency bins is equalized):

$$\mathbf{X}_{norm\_time,i}\,[k] \;=\; \frac{1}{\frac{1}{N}\sum_{j=1}^{N}|\mathbf{X}_j[k]|}\,|\mathbf{X}_i[k]|, \qquad (1)$$

where $|\mathbf{X}_i[k]|$ denotes the magnitude spectrum and N is the number of extracted spectra. This normalization does not only adjust the energy among the spectral components but also makes the standard deviation of them comparable. This property comes from the fact that spectral components with large average energy have also higher variances. Thus equalizing the mean energy also adjusts the standard deviations. However, there is a need for a second normalization step: The energy of the spectrum varies strongly over time even if no speaker turn occurs. Normalizing the sum of all spectral components to 1 for every spectrum according to

$$\mathbf{X}_{norm\_freq,i}\,[k] \;=\; \frac{1}{\sum_{l=1}^{K}\mathbf{X}_{norm\_time,i}[l]}\,\mathbf{X}_{norm\_time,i}[k] \quad (2)$$

is usefull, where K is the number of frequency bins. This normalization is performed for each spectrum (for more information on the effect of the normalization steps see [7]).

After these normalization steps, the high frequency bins are still not very distinctive between different speakers. It turns out, that the reason for this are the unvoiced parts of the speech signal. These components have a flat spectrum and are similar for all speakers. Hence, they do not carry discriminative information for speaker segmentation. We use the entropy of the spectrogram to remove the unvoiced speech segments before performing the normalizations (Eqn. 1 and Eqn. 2).

After normalizing the energy of every magnitude spectrum to 1, we can treat it as probability density function (p.d.f.). If the spectrum is noise-like, the p.d.f. is flat and thus the entropy is large. Information entropy is defined as follows ([8]):

$$H(\mathbf{X}_i) = -\sum_{k=1}^{K} p(\mathbf{X}_i[k]) \cdot \log_2(p(\mathbf{X}_i[k])), \qquad (3)$$

where K is the number of possible values of the discrete random variable $\mathbf{X}_i$ and $p(\mathbf{X}_i[k])$ is the probability of $\mathbf{X}_i[k]$. By removing the spectra which have high entropy values, the non-flat parts remain in the data which are mostly the parts including the voiced speech information. The number of removed spectra is derived from the average entropy H_{Avg} of all spectra of the audio stream multiplied by a factor hec. If $H_{Avg} \cdot hec < H(\mathbf{X}_i)$ then spectra $\mathbf{X}_i$ is removed, otherwise it is used for further processing.

After the removal of the high entropy spectra and the two normalization steps, the high frequency speaker dependent components of 3 kHz and above are extracted for applying the distance metric.

2.2. Distance Measurement

The distance measurement has to detect those points that indicate a speaker turn. Prior to the distance measurement, the data $\mathbf{X}_{norm\_freq,i}\,[k]$ are smoothed in time-direction using a moving average filter. Then, two adjacent windows A and B are moved stepwise over the features $\mathbf{X}_{norm\_freq,i}\,[k]$. The Euclidian distance between the features within the windows is computed after every step

$$d_{Eucl}(\bar{\mathbf{X}}_A[k], \bar{\mathbf{X}}_B[k]) = \left(\sum_{k=1}^{K}\left|\bar{\mathbf{X}}_A[k] - \bar{\mathbf{X}}_B[k]\right|^2\right)^{1/2}, \qquad (4)$$

where $\bar{\mathbf{X}}_A[k]$ is the mean of the k^{th} frequency bin of window A and $\bar{\mathbf{X}}_B[k]$ the mean of the k^{th} bin of window B.

2.3. Classification

To reduce the number of local maxima, the results of the Euclidean distance measurement are smoothed with the help of a moving average filter. Afterwards, the decision procedure has to determine whether a local distance maximum belongs to a speaker turn or not. Here, the same approach is used as suggested in [2]. The decision depends on the differences of the considered local maximum and the two minima at its left and right side as

$$|\, d_{Eucl}^{\max} - d_{Eucl}^{\min_r} \,| > \alpha\sigma \qquad (5)$$

and

$$|\, d_{Eucl}^{\max} - d_{Eucl}^{\min_l} \,| > \alpha\sigma, \qquad (6)$$

where d_{Eucl} denotes the Euclidean distance, $d_{Eucl}^{\max}$ the considered local distance maximum, $d_{Eucl}^{\min_r}$ the local minimum at the right of $d_{Eucl}^{\max}$ and $d_{Eucl}^{\min_l}$ the local minimum at its left. Parameter α can be adapted depending on the input speech data to optimize the segmentation performance and σ denotes the standard deviation of the Euclidean distance measure for the speech utterance. If equation 5 and 6 are true for a local maximum, the position of this maximum is declared to be a speaker turn.

3. EXPERIMENTAL SETUP

We compare the performance of the FREQDIST algorithm to the DISTBIC algorithm introduced in [2]. Therefore, we use two different data sets:

1. The TIMIT database [1] contains a total of 6300 sentences, 10 sentences spoken by each of 630 speakers from 8 major dialect regions of the United States. The sampling rate is 16 kHz. The shortest speech segment is about 2.5 seconds.
2. Presseclub: The Presseclub data is a collection of broadcasts of the talkshow Presseclub of the German Westdeutschen Rundfunk WDR. Five broadcasts of about 45 minutes each were recorded and the speaker turns were labelled. The sampling rate is 22.05 kHz. On average, the speech segments are much longer compared to TIMIT.

To achieve the best performance, parameters have to be optimized. The two algorithms have some different and some common parameters. For the computations of the spectrogram, the algorithms use a window length of 20 ms and a step size of 10 ms. The other parameters are listed below.

- **Window size** (*win*) used for DISTBIC and FREQDIST: The window size defines the width of one of the two windows that are moved stepwise over the extracted feature vectors (i.e it determines the number of feature vectors).
- **Step size** used for DISTBIC and FREQDIST: The step size of the two windows is always 0.1 seconds.
- **Smoothing coefficient** (*sc*) used for DISTBIC and FREQDIST: This coefficient defines the degree of smoothing of the distance function.
- α used for DISTBIC and FREQDIST: This parameter is used within the maximum detector introduced in section 2.3.
- λ used for DISTBIC: This parameter is used as a balance operator for the Bayesian Information Criterion (see [2]).
- **Spectrogram smoothing coefficient** (*sc_spec*) used for FREQDIST: This parameter determines the size of the moving average filter introduced in 2.2.
- **High-entropy cutting** (*hec*) used for FREQDIST: This threshold parameter influences the entropy-based spectrogram segmentation introduced in section 2.1.

The performance measurement is performed with the help of the false alarm rate (FAR)

$$\text{FAR} = 100\times \frac{\text{number of false accepted turns}}{\text{number of all speaker turns} + \text{number of false accepted turns}}\% \tag{7}$$

and the missed detection rate (MDR)

$$\text{MDR} = 100 \times \frac{\text{number of missed speaker turns}}{\text{number of all speaker turns}}\%. \tag{8}$$

To determine the values of these rates, the area of acceptance (AOA) has to be defined. The area of acceptance is the area around a real speaker turn where an estimated turn is accepted to be a real turn. Because of the differences of the average speaking duration per speaker, this parameter is set to different values for different databases.

4. RESULTS: TIMIT

Table 1 shows the FAR and MDR of the FREQDIST and the DISTBIC algorithm using 200 speaker turns. The test data consisted of randomly chosen speakers. Each speaker speaks 3 concatenated sentences of altogether 7.5 seconds on average. The performance of both algorithms was measured using parameter sets optimized for data of 7.5 seconds average speaking duration per speaker. Table 2 shows the parameters.

Table 1: FAR and MDR of DISTBIC and FREQDIST using 7.5 seconds TIMIT data and 200 speaker turns.

	AOA (s)	**FAR (%)**	**MDR (%)**
DISTBIC	0.5	17.1	19.6
	1.0	10.4	13.1
FREQDIST	0.5	14.4	16.1
	1.0	5.1	7.0

Table 2: Optimized parameter values for TIMIT data.

	win	α	λ	*sc*	*sc_spec*	*hec*
DISTBIC	6	60	1	2	-	-
FREQDIST	5	20	-	2	20	1.1

Table 3: FAR and MDR of DISTBIC and FREQDIST using TIMIT data with added noise and 200 speaker turns.

	SNR (dB)	**FAR (%)**	**MDR (%)**
DISTBIC	10	21.89	21.11
	20	16.26	14.57
	40	12.37	14.57
	60	10.36	13.07
	80	10.36	13.07
	100	10.36	13.07
FREQDIST	10	52.6	17.59
	20	30.21	17.59
	40	6.03	6.03
	60	4.62	6.53
	80	5.12	7.04
	100	5.13	7.04

The results show that both algorithms work well with this data set. The FAR and the MDR reach values of far below 20%. The FREQDIST algorithm performs better than the DISTBIC algorithm. The FREQDIST algorithm reaches a FAR less than half of the value of the optimized DISTBIC algorithm and a MDR of 7.0% instead of 13.1%.

In the next experiment, we study the behavior of the algorithms' dependency on the average speaking duration. We change the duration from 2.5 seconds (i.e. one sentence of the TIMIT database) to 20 seconds on average (i.e. 8 concatenated sentences). Again, we use 200 randomly selected speakers. The parameters of the algorithms are set to the same optimized parameters as before. In this case both algorithms perform quite similar (see figure 1a and 1b). For longer speaking durations than 7.5 seconds, the MDR stays nearly constant but the FAR increases to 50% for 20 seconds. For smaller durations, the FAR does not exceed 20% but the MDR can be as large as 65%. Unfortunately, the strong influence of the average speaking duration on the performance makes a reliable application of the algorithms in a real-world scenario questionable. To get good performance results, the parameters - especially the window size - needs to be adapted depending on the speaking duration. But without a priori information on the data, this adaptation is not possible.

Finally, table 3 shows the segmentation performance measured with added white Gaussian noise. For this experiment, the AOA is set to 1 second. Again, the parameters are set to the values of table 2. Adding noise does not influence the algorithms' performance as long as the signal to noise ratio (SNR) is higher than 20dB. If the SNR is smaller, the performance of the FREQDIST algorithm degrades. The FREQDIST algorithm responds sensitive on changes in the high frequency parts of the spectrum. In real-world data these spectral components are prone to be polluted by background noise or changes of the head-microphone position.

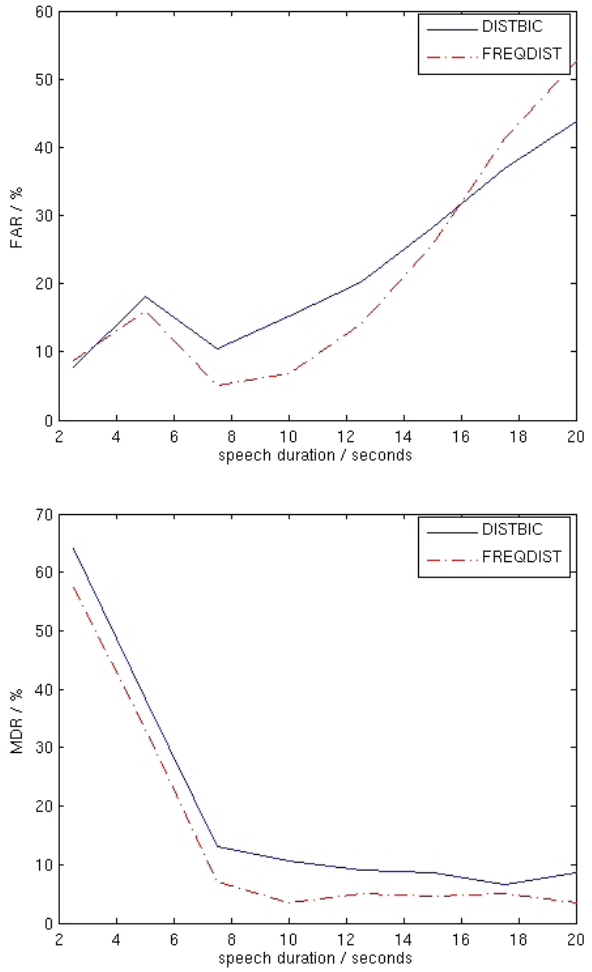

Fig. 1: FAR (a) and MDR (b) of the DISTBIC and the FREQDIST algorithm for different average speaking durations (AOA = 1s).

5. RESULTS: PRESSECLUB

Table 4 and 5 present the FAR and the MDR of the DISTBIC and the FREQDIST algorithm using two different Presseclub broadcasts. Here, the speaking durations are changing continually. High FARs of between 40% and 60% and MDRs of between 55% and 60% are not remarkable. The results using the Presseclub data show the effect of the inflexible behavior of both algorithms. The results are achieved by setting the parameters to the optimized values shown in table 6.

6. CONCLUSION

A new approach to speaker segmentation in the frequency domain is presented. It reweights the spectrum using normalization techniques. Additionally, noise-like spectra are removed based on the entropy. For the TIMIT corpus we achieve better performance compared to the DISTBIC algorithm. Our method even works well at low noise levels. For Westdeutscher Rundfunk real-world data the FREQDIST algorithm performs slightly better. These results show that higher frequency components are an interesting and useful source of speaker dependent features that can be used to enhance the performance of speaker related algorithms.

Table 4: FAR and MDR of DISTBIC and FREQDIST using Presseclub data: Broadcast 1.

	AOA (s)	FAR (%)	MDR (%)
DISTBIC	2	57.7	61.4
FREQDIST	2	42.9	57.9

Table 5: FAR and MDR of DISTBIC and FREQDIST using Presseclub data: Broadcast 2.

	AOA (s)	FAR (%)	MDR (%)
DISTBIC	2	60.4	55.3
FREQDIST	2	50.0	55.3

Table 6: Optimized parameter values for Presseclub data.

	win	α	λ	*sc*	*sc_spec*	*hec*
DISTBIC	8	120	1	2	-	-
FREQDIST	6	150	-	2	150	1.1

7. REFERENCES

[1] L. Lamel, R. Kassel, and S. Seneff, "Speech database development: Design and analysis of the acoustic-phonetic corpus," in *Proceedings of the DARPA Speech Recognition Workshop, Report No. SAIC-86/1546*, 1986.

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

[3] X. Anguera Miró, *Robust Speaker Diarization*, Ph.D. thesis, Universitat Politcnica de Catalunya, 2006.

[4] S.E. Tranter and D.A. Reynolds, "An overview of automatic speaker diarization systems," *IEEE Transaction on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1557–1565, 2006.

[5] P. Vary, U. Heute, and W. Hess, *Digitale Sprachsignalverarbeitung*, B.G. Teubner Stuttgart, 1998.

[6] M.R. Sambur, "Selection of acoustic features for speaker identification," in *IEEE Transactions on Acoustics, Speech, and Signal Processing*, vol. 23, pp. 176–182. 1975.

[7] C. Boehm, "Unsupervised speaker segmentation in one-channel speech data," M.S. thesis, Graz University of Technology, 2007.

[8] T. Cover and J. Thomas, *Elements of Information Theory*, John Wiley & Sons, 1991.