Strategies to Improve a Speaker Diarisation Tool

David Tavarez, Eva Navas, Daniel Erro, Ibon Saratxaga

Aholab - Dept. of Electronics and Telecommunications. Faculty of Engineering.
University of the Basque Country. Alda. Urquijo s/n 48013 Bilbao
email: david, eva, derro, ibon@aholab.ehu.es

Abstract

This paper describes the different strategies used to improve the results obtained by our off-line speaker diarisation tool with the Albayzin 2010 diarisation database. The errors made by the system have been analyzed and different strategies have been proposed to reduce each kind of error. Very short segments incorrectly labelled and different appearances of one speaker labelled with different identifiers are the most common errors. A post-processing module that refines the segmentation by retraining the GMM models of the speakers involved has been built to cope with these errors. This post-processing module has been tuned with the training dataset and improves the result of the diarisation system by 16.4% in the test dataset.

Keywords: Speaker Diarisation, Speaker Clustering, Evaluation

1. Introduction

The aim of speaker diarisation is to detect speaker changes in an audio recording and to identify which of the resulting speech segments come from the same speaker, without any prior information about the number or identity of the speakers (Tranter and Reynolds, 2006). To achieve this goal several tasks are performed, usually in a sequential way. These tasks typically include speech detection, speaker change detection, speaker clustering and resegmentation of the audio stream. To objectively assess the validity of the algorithms developed, competitive evaluation campaigns like NIST Rich Transcription[1] and Albayzin diarisation evaluation (Zelenák et al., 2010) are organized. In these campaigns, different research groups tests their algorithms with a shared database, which allows for performance comparison and helps identifying new trends.

We built a diarisation system for Albayzin 2010 evaluation campaign that obtained good results, even if it did not include any resegmentation step (Luengo et al., 2010). In this paper, the strategies proposed to cope with the errors made by this system and the improvements in the results achieved are presented.

In section 2 the baseline diarisation system presented by our group to Albayzin 2010 evaluation campaign is described. Section 3 presents the database used in the experiments. Section 4 focuses on the analysis of the errors made by the baseline system and proposes the strategies to cope with them. The results of the post-processing module developed are presented in section 5. Finally, some conclusions are drawn in section 6.

2. Baseline Speaker Diarisation System

Figure 1 shows a schematic diagram of the baseline speaker diarisation tool. The algorithm is based on an efficient implementation of a BIC change detector and an off-line speaker clustering. In the following sections, each step of the algorithm will be explained with more detail.

[1] http://www.itl.nist.gov/iad/mig/tests/rt/

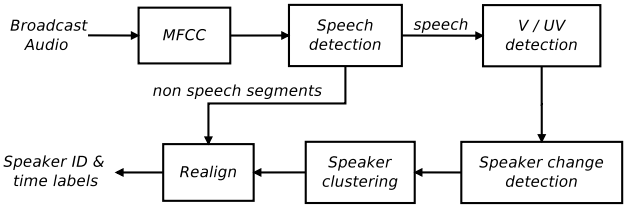

Figure 1: Diagram of the baseline diarisation system

2.1. Speech detection

A separate GMM model with 16 mixtures was trained for music, noise, clean speech, speech+music and speech+noise, using the development recordings and the audio segmentation labels provided by the Albayzin 2010 diarisation challenge (Zelenák et al., 2010) organisation. These models are used in a Viterbi segmentation in order to detect audio segments with and without speech. Development experiments showed that the addition of derivatives of MFCC provides slightly better segmentation results, therefore 12 MFCC with first and second derivatives were used for the classification. Finally the speech detection labels were post-processed in order to discard silences shorter than 500 ms. Only the segments identified as speech are then provided to the speaker change detection algorithm.

2.2. Voiced unvoiced detection

The speaker change detection step uses only voiced frames, discarding the unvoiced ones. In order to make the voiced/unvoiced (VUV) estimation, the PTHCDP algorithm described in (Luengo et al., 2007) was used. This algorithm uses cepstrum transformation and dynamic programming in order to estimate the F0 curve and the VUV information.

2.3. Speaker change detection

For the initial speaker change detection, a growing window architecture and BIC metric (Chen and Gopalakrishnan, 1998) are used. The growing window provides better

results than a fixed-size sliding window, but the computational cost is also larger. In order to reduce the time of computation as much as possible, the solution described in (Cettolo and Vescovi, 2003) is used:

- No speaker change is searched in the first and last 2 seconds of the window.
- The window grows 2 seconds every time that no change is detected.
- Once the window reaches 20 seconds, instead of growing, it becomes a sliding window.
- For each window, a speaker change is searched every 250 ms. If a change is located, the search is refined to 50 ms.
- Once a change is found, the window size is reset to 5 seconds.

This solution provides the same accuracy as the growing-window algorithm, while keeping the window size and the amount of calculation to a minimum. Furthermore, the calculation of the BIC values is also optimised by using a buffer of cumulative sums as described in the work made by Cettolo and Vescovi (2003). Development results showed that discarding unvoiced frames and using only voiced ones decreased the diarisation error by 12%. Therefore, only voiced frames were used for the speaker change detection. Similarly, it was confirmed that the use of feature derivatives was not convenient for this task.

2.4. Speaker clustering

The speaker clustering is performed applying a hierarchical agglomerative bottom-up off-line clustering process (Hastie et al., 2009). Initially each segment detected by the speaker change detection module constitutes a different cluster. This module computes the BIC difference between each pair of clusters and selects the pair with the minor difference. If this difference is negative both clusters are combined and the cluster statistics are updated. This process is repeated until the minor BIC difference found is greater than zero.

3. Database

The baseline speaker diarisation tool was applied to the broadcast speech database used in Albayzin 2010 speaker diarisation challenge. This is a Catalan broadcast news database from the 3/24 TV channel recorded by the TALP Research Center from the UPC and annotated by Verbio Technologies. The database contains different types of speech material, like advertisements, reports, interviews, discussions and short statements. The original audio tracks were extracted at 32 kHz sample rate, 16 bit resolution, but were down-sampled to 16 kHz sample rate.

The database includes around 87 hours of audio, with the following distribution of background conditions: Clean speech: 37%; Music: 5%; Speech with music in the background: 15%; Speech with noise in the background: 40%; Other: 3%.

Although TV3 is primarily a Catalan television channel, the recorded broadcasts include about a 16% of Spanish speech segments. There are 24 recordings and the number of speakers per recording ranges from 30 to 250. Some of these speakers appear in different recordings (journalists and anchors) but most of them appear in only one recording. About 60% of the speakers in the database are male. For the Albayzin 2010 speaker diarization evaluation a subset of 8 recordings, totalling approximately 30 hours was selected for testing and the rest was used for training and developing the systems.

4. Strategies to reduce diarisation error

The results obtained by the baseline system in the training and test sets of the database are shown in Table 1. These values are calculated according to the criteria defined by NIST and the primary metric is the overall speaker diarisation error rate (DER). The main source of error is the incorrect labelling of the speakers, accounting for the 83% of the DER.

	Train set	Test set
Missed Speaker Time	2.60%	2.80%
False Alarm Speaker Time	2.30%	2.20%
Speaker Error Time	23.30%	25.10%
Overall Speaker Diarisation Error	28.25%	30.11%

Table 1: Results of the baseline system

4.1. Error Analysis

An exhaustive analysis of the errors observed in the training set has been accomplished in order to reduce the final DER. The time labels obtained were compared with reference labels provided by the Albayzin organisation to find the nature of the different errors and design appropriate techniques to treat each case separately. According to Table 1, the influence of the speaker error time (SET) in DER is obvious, so it has been studied in detail. This particular error appears in three different ways:

- Short segments from one speaker that the clustering process assigns to other speaker when the BIC has detected a speaker change that does not really exist. This type of error typically represents 2-5% of total SET and occurs after a long speaker turn.
- Different appearances of one speaker that the clustering process interprets as two or more different speakers. It is the main source of error and produces about 75% of total SET.
- Segments of speakers with short appearances that the clustering process assigns to other already identified speaker instead of creating a new cluster. It means about 20% of total SET and it is usually undetectable applying automatic procedures.

Both Missed Speaker Time (MST) and False Alarm Speaker Time (FAST) appear as a result of a malfunction of the speech detection block. FAST represents about 2%

of DER and in most cases corresponds to segments of music that the clustering process usually interprets like new speakers.

4.2. Proposed Strategies to Cope with Errors

After the analysis of the system errors, several strategies have been designed in order to reduce each type of error, focusing particularly on the SET reduction.

4.2.1. Strategy 1

Short segments suspicious of being wrong labelled, especially those located between long appearances of the same speaker, are studied and removed if considered necessary. Let us assume that one segment is suspicious of having been incorrectly labelled as X (according to its duration, for instance), being the adjacent labels A and B, respectively. First, a separate GMM G_x is trained using other segments reliably labelled as X. Similarly, two GMMs corresponding to the adjacent speakers, G_a and G_b, are trained from all the available data of these speakers (note that in the particular case when the adjacent labels are equal to A, only G_a is trained). Finally, if the vectors in the suspicious segment are better modelled by G_a or G_b than by G_x, the segment is assimilated.

4.2.2. Strategy 2

Once the incorrect short segments have been taken care of, the speakers labelled with different identifiers can be treated. When these errors appear, the number of clusters, M, becomes higher than the actual number of speakers, N. We aim at determining which of these M clusters should be unified. First, the data inside each cluster are decomposed into a training set and a validation set. Then, we train one GMM for each cluster using its corresponding training set: G_i, i=1...M. Once the M models are trained, we compute an M-by-M likelihood-difference matrix L, where L_{ij} contains the likelihood of the validation set inside the ith cluster given G_i minus the likelihood of the same data given G_j. In order to increase the difference between the clusters, we apply a logarithmic scale and modify the matrix L as follows:

$$L = \begin{cases} 0 & if \quad L_{ij} \leq 1 \\ log(L_{ij}) & if \quad L_{ij} > 1 \end{cases} \qquad (1)$$

Next, for each cluster i, we calculate the mean (m) and standard deviation (σ) of the modified likelihood-difference values (each row of the L matrix). The threshold to decide which clusters should be recombined is calculated taking into account this mean and standard deviation, according to the expression thr_1=m-cσ, where c is a weighting coefficient that has been empirically set to 1.7 taking into account the training sessions of the database. In Figure 2 the likelihood-difference values for speaker 2 are displayed. The likelihood difference for speakers 1 and 25 is well below the recombination threshold (indicated by the black line), so we can combine speakers 1, 2 and 25 in a single cluster.

If for a given cluster, there are too many candidates with likelihood differences below the established threshold, the probability that they belong to the same speaker is low. Usually this indicates that there are several speakers with similar voices that should not be recombined. To avoid these incorrect cluster combinations, a secure threshold is established at m-1.4σ (less strict than the recombination threshold) and recombination is done only for clusters with less than 4 likelihood differences below this secure threshold. Besides, development experiments showed that not all the values below the combination threshold can be trusted. Speech segments corresponding to some of the labelled speakers do not contain enough data to obtain a robust GMM model, and in this case the matrix L presents many close to zero values. In most of these cases, the obtained combination threshold is below zero, so no cluster is recombined. However in a few cases, the combination threshold is barely above zero and the clusters with lower likelihood difference values are erroneously recombined. To avoid this incorrect recombination, clusters are only combined if the combination threshold is greater than a minimum allowed value. This minimum has been established in 0.25 taking into account the training sessions of the database.

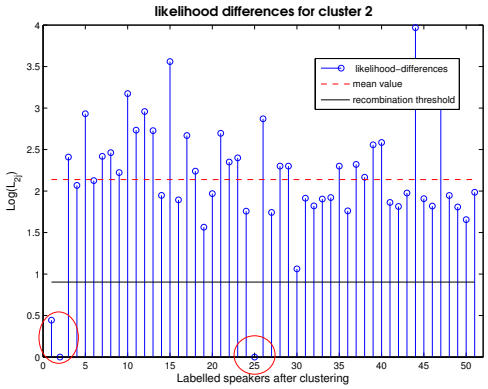

Figure 2: Difference between the likelihood of the validation set inside the cluster 2 given G_2 and the likelihood of the same data given G_j in logarithmic scale

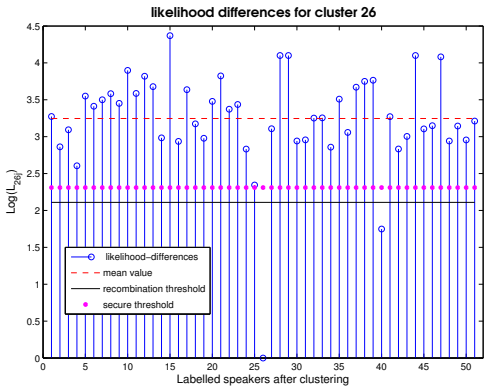

Figure 3: Difference between the likelihood of the validation set inside the cluster 26 given G_{26} and the likelihood of the same data given G_j in logarithmic scale

Figure 3 shows the likelihood differences for cluster 26: the pink dotted line indicates the secure threshold and it can be

seen that less than 4 clusters have difference values below this threshold; cluster 40 has a likelihood difference smaller than the combination threshold and then, would be combined with cluster 26. However, development experiments showed that in this case this combination is not correct, but due to the high similarity between the voices of the corresponding speakers. In order to cope with this problem a maximum value for the likelihood difference is imposed. As the mean value of the differences (m) increases, the admitted maximum likelihood-difference can be higher without losing reliability. Therefore, three different maximum values for the likelihood differences (U) are considered according to:

$$U = \begin{cases} 0.25 & if \quad m \leq 1.95 \\ 1.23 & if \quad 1.95 < m < 2.6 \\ 1.64 & if \quad m \geq 2.6 \end{cases} \tag{2}$$

5. Results

Table 2 and 3 show the obtained results for development and test sessions respectively. For each session, results for the baseline system are presented as well as the final DER obtained after applying each proposed strategy. Strategy 1 barely modifies the DER, but it increases the purity of the clusters and improves the performance of strategy 2.

Session	Baseline DER	Strategy 1	Strategy 2
session1	22.17%	21.88%	27.15%
session2	24.58%	24.50%	13.56%
session3	23.10%	22.99%	17.47%
session4	27.47%	27.31%	27.31%
session5	14.15%	14.15%	11.14%
session6	21.22%	21.19%	16.06%
session7	24.84%	24.87%	24.03%
session8	27.26%	27.26%	19.75%
session9	28.92%	29.61%	27.81%
session10	34.75%	34.62%	22.86%
session11	27.94%	28.13%	16.39%
session12	27.42%	27.42%	25.29%
session13	31.92%	31.64%	30.57%
session14	41.16%	41.26%	25.66%
session15	32.50%	32.52%	21.94%
session16	32.06%	31.97%	23.23%

Table 2: Results of the strategies for development sessions

As displayed in Table 2 and 3, almost all the sessions achieve a DER reduction when both strategies are applied. Also in Table 4 it can be seen that the improvement obtained by including the post-processing block is considerably high. The final DER is reduced by 21.5% for development part of the database and 16.4% for test part, which proves the validity of the proposed system.

The same post-processing module has been applied with no modification to another diarisation system that also participated in Albayzin 2010 evaluation campaign (Luengo et al., 2010). This system has a similar architecture to the one presented here, but works online. The addition of the developed post-processing module eliminates the online character of the system, but improves the results by a 18.18% as can be seen in Table 5.

Session	Baseline DER	Strategy 1	Strategy 2
session17	34.92%	34.89%	26.03%
session18	31.35%	31.48%	24.88%
session19	27.14%	27.14%	20.28%
session20	34.72%	35.06%	29.71%
session21	34.20%	34.09%	18.02%
session22	33.06%	33.18%	34.38%
session23	24.92%	25.14%	23.16%
session24	22.99%	23.26%	21.81%

Table 3: Results of the strategies for test sessions

Session	Baseline DER	Strategy 1	Strategy 2
Development	28.25%	28.24%	22.17%
Test	30.11%	30.24%	25.17%

Table 4: Total results of the post-processing module

Session	Baseline DER	After post-processing
Development	26.77%	21.44%
Test	27.17%	22.23%

Table 5: Total results of the post-processing module for the online diarisation system

6. Conclusions

Our off-line speaker diarisation tool has been described and the errors made by this tool when working with the Albayzin 2010 speaker diarisation challenge database have been presented and analysed. Two strategies have been proposed to deal with each type of error: removing the incorrect short segments and combining the clusters that correspond to the same speaker. These two strategies have been implemented and tuned using the training dataset. A post-processing module that applies the two strategies has been built and added to the baseline off-line diarisation system, with a 16% improvement in the results for the test dataset. The same post-processing module has been used with another diarisation system to check its generalisation capabilities and the results have also been improved by 18% in this case. New strategies to cope with FAST must also be considered and tests using different diarisation databases should also be made.

7. Acknowledgements

The authors would like to thank Iker Luengo for the development of the baseline diarisation system.

This work has been partially supported by UPV/EHU (Ayudas para la Formación de Personal Investigador), the Basque Government (Berbatek, IE09-262) and the Spanish Ministry of Science and Innovation (Buceador Project, TEC2009-14094-C04-02).

8. References

M. Cettolo and M. Vescovi. 2003. Efficient audio segmentation algorithms based on the bic. In *International Conference on Acoustics, Speech, and Signal Processing (ICCASP 03)*, volume 6, pages 537–540, April.

S. S. Chen and P. S. Gopalakrishnan. 1998. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *DARPA speech recognition workshop*, volume 6, pages 127–132.

T. Hastie, R. Tibshirani, and J. Friedman. 2009. *The Elements of Statistical Learning (2nd edition)*. Springer.

I. Luengo, I. Saratxaga, E. Navas, I. Hernáez, J. Sánchez, and I. Sainz. 2007. Speaker, environment and channel change detection and clustering via the bayesian information criterion. In *International Conference on Acoustics, Speech, and Signal Processing (ICCASP 07)*, pages 1057–1060, Honolulu, USA, April.

I. Luengo, E. Navas, I. Saratxaga, I. Hernáez, and D. Erro. 2010. AhoLab Speaker Diarisation System for Albayzin 2010. In *FALA 2010*, pages 393–396, Vigo.

S. E. Tranter and D. A. Reynolds. 2006. An overview of automatic speaker diarization systems. *IEEE Trans. on Audio, Speech and Laguage processing*, 14(5):1557–1565.

M. Zelenák, H. Schulz, and J. Hernando. 2010. Albayzin 2010 evaluation campaign: Speaker diarization. In *VI Jornadas en Tecnologa del Habla and II Iberian SLTech Workshop*, pages 301–304, Vigo, Spain, November.