PURITY ALGORITHMS FOR SPEAKER DIARIZATION OF MEETINGS DATA

Xavier Anguera[1,2], *Chuck Wooters*[1], *Javier Hernando*[2]

[1] International Computer Science Institute (ICSI)
1947 Center St., Suite 600, Berkeley, CA 94704, U.S.A.
[2] Technical University of Catalonia (UPC), TALP Research Group
Jordi Girona 1-3 D5, 08034 Barcelona, Spain
{xanguera,wooters}@icsi.berkeley.edu, javier@gps.tsc.upc.es

ABSTRACT

When performing speaker diarization, it is common to use an agglomerative clustering approach where the acoustic data is first split in small pieces and then pairs are merged until reaching a stopping point. When using a purely agglomerative clustering technique, one cluster cannot be split into two. Therefore, errors caused by multiple speakers being assigned to one cluster can be common. Furthermore, clusters often contain non-speech frames, creating problems when deciding which two clusters to merge and when to stop the clustering. In this paper, we present two algorithms that aim to purify the clusters. The first assigns conflicting speech segments to a new cluster, and the second detects and eliminates non-speech frames when comparing two clusters. We show improvements of over 18% relative using three datasets from the most current Rich Transcription (RT) evaluations.

1. INTRODUCTION

The goal of speaker diarization is to segment an audio recording into speaker-homogeneous regions [1]. Typically, this segmentation must be performed with little or no knowledge of the characteristics of the recording or of the speakers in the recording. Speaker diarization is sometimes referred to as the "Who spoke when?" task, although systems generally do not identify specific speakers by name.

The most common algorithm used in speaker diarization systems is hierarchical agglomerative clustering [2],[3],[4],[5], including signifcant work applying these techniques to the meetings domain [6], [7], [8], [9]. In this approach, the signal is first divided into a number of short segments (more than the estimated number of speakers), and then the segments are iteratively merged together based on their acoustic similarity. The process stops when a stopping criterion is met. After the clustering, systems may optionally resegment the data to refine the speaker boundaries.

In all of these systems, the initial clustering is very important because an error at the beginning can propagate until the end of the clustering, causing an increase in the error rate. In pure agglomerative systems, there is no explicit method to split clusters — only to merge them. One particular source of error [non-speech frames or frames from other speakers] that we observe is assigning more than one acoustic source to one cluster, making the cluster "impure". Running an agglomerative clustering system with impure models leads to errors when considering whether two models should merge or not, and when considering a stopping point.

By looking at the intermediate stages of a clustering process, we observe that cluster impurity can be studied at two different levels. At the frame level, there are non-speech frames that the speech/non-speech detector fails to detect. In speaker recognition it is common practice to use a Speech Activity Detection (SAD) algorithm that removes frames with low energy. By inspecting a Gaussian mixture model (GMM) with few Gaussians trained on the data, we observe that the non-speech frames are normally very well modelled by a few of the Gaussian mixtures and have a high likelihood. We present an algorithm that takes advantage of this fact and cleans the models before merging.

At the segment level, we observe that some clusters contain full segments from more than one speaker or acoustic source. A segment is a set of adjacent frames assigned to the same speaker cluster by the decoding algorithm; if the segment boundaries are correct it is referred as a speaker turn. We present an algorithm that locates these segments using the Bayesian Information Criterion (BIC) [10]. The segments are located by doing a ΔBIC comparison of each segment with the most representative segment in the cluster. The segment among all clusters with the lowest ΔBIC value is split from the original cluster, and assigned to a new cluster.

We have run experiments on both purification algorithms using three datasets from the NIST's RT04s and RT05s evaluations [11], two as development sets and one as an evaluation set. We show improvements over the baseline in some cases of more than 18% relative in both the development set and the evaluation set.

2. AGGLOMERATIVE SPEAKER CLUSTERING

As explained in [5] and [12], our speaker clustering system is based on agglomerative clustering. It initially splits the data into K clusters (where $K >$ number of speakers), and then iteratively merges the clusters (according to a merge metric based on ΔBIC) until a stopping criterion is met. Our clustering algorithm models the acoustic data using an ergodic hidden Markov model (HMM), where the initial number of states is equal to the initial number of clusters (K). Upon completion of the algorithm's execution, each remaining state is taken to represent a different speaker. Each state contains a set of M_D sub-states, imposing a minimum duration on the model (we use $M_D = 3$ seconds). Within the state, each of the sub-states share a probability density function (PDF) modelled via a Gaussian mixture model (GMM).

Our clustering algorithm for the meetings domain, taken as a baseline in this work, consists of the following steps:

1. Use delay-and-sum (D&S) [13] to create one single "enhanced" channel from all input microphones.
2. Run speech/non-speech detection on the enhanced input data.
3. Extract acoustic features from the data and remove non-speech frames.

4. Create models for K initial clusters (we use $K = 10$) via linear initialization.
5. Perform iterative merging using the following steps:
   (a) Run a Viterbi decode to resegment the data.
   (b) Retrain the models via an Expectation-Maximization (EM) algorithm using the segmentation from step (a).
   (c) Select the cluster pair with the largest merge score (based on ΔBIC) that is > 0.0.
   (d) If no such pair of clusters is found, stop and output the current clustering.
   (e) Merge the pair of clusters found in step (c). The models for the individual clusters in the pair are replaced by a single, combined model.
   (f) Go to step (a).

In the meetings domain, there are several available audio channels as there are multiple microphones installed around the room. We use a variation of the D&S technique [13] to combine all data into an enhanced channel, which is then used in the speaker clustering process. This technique does not require any knowledge of the number of people or their locations, nor the locations of the microphones in the room.

For the merging and clustering stopping criteria, we use a variation of the commonly used Bayesian Information Criterion (BIC) [10]. The ΔBIC compares two possible models: two clusters belonging to the same speaker or to different speakers. The variation used was introduced by Ajmera et al. [12], [14], and consists of the elimination of the tunable parameter λ by ensuring that, for any given ΔBIC comparison, the difference between the number of free parameters in both models is zero.

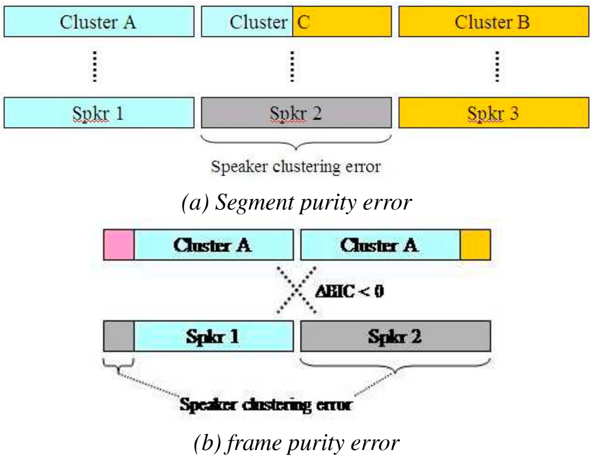
Despite the speech/non-speech detector, there is a portion of non-speech frames that are processed by the system and assigned to a cluster, corrupting it. Furthermore, the existence of misassigned speech segments deteriorates the speaker models and increases the error rate. In the next section, we propose two algorithms to help mitigate this problem.

3. CLUSTER PURIFICATION ALGORITHMS

Given the speaker clustering algorithm presented above, there will always be frames assigned to a cluster which do not belong to the modelled speaker. These frames are either silence or frames from another speaker. We can define two different sources that can cause purity problems in clusters.

One source of error occurs when a cluster is created from speech segments from multiple speakers. In standard agglomerative systems there is no mechanism to split a cluster when segments from different speakers are assigned to the same cluster. This effect causes an increase in the final speaker error as seen in Figure 1(a) for the case of two misplaced segments of two existing speakers. At the end of the processing, the mixed cluster is likely to be assigned to an non existent speaker, causing a large increase on the Diarization Error Rate (DER).

The second source of error comes from the interference of non-speech frames during cluster comparison. This is particularly true for short silences and short acoustic events that belong to the modelled speaker but do not discriminate one speaker from another. This can affect the final clustering in two ways, as seen in Figure 1(b). First, when comparing two clusters belonging to the same speaker, the confounding frames can cause ΔBIC to decide to keep them separate. Second, false alarm errors are produced when non-speech frames are assigned to one of the speakers.



(a) Segment purity error

(b) frame purity error

Fig. 1. Possible Speaker clustering errors due to clusters purity

Both sources of error are interrelated and are caused by frames that are assigned to the wrong acoustic model. The difference is the unit that we consider missassigned. In next subsections we propose solutions to both problems. The first algorithm identifies the segments that acoustically deviate most from their cluster, and splits them into a new cluster. This is referred to as "segment level" purification. The second algorithm locates the individual frames within a cluster than can cause problems in the merging state and avoids using them. It is referred to as "frame level" purification.

3.1. Segment Level Purification

In the presented agglomerative speaker clustering system we perform Viterbi decodings to allow speech segments to be reassigned to their closest model after each merging iteration. There are some situations where a cluster still retains speaker segments from more than one speaker; we need a mechanism to force splitting this cluster into two parts. Below we present a "segment level" purification algorithm which is executed after every merging step of the clustering algorithm. This algorithm aims to detect and extract the speech segments that are most dissimilar to the models in the following way:

1. Find the segment that best represents each model (highest likelihood normalized by the number of frames of the segment).
2. Compute, within each cluster, the ΔBIC value between the best segment (found in step 1) and each of the other segments. If all pairs have a value greater than a minimum purity (empirically set to -50) that model is labelled as "pure" and is not checked again in subsequent iterations.
3. The segment that most differs from its model's best segment is assigned to a new model. All models are retrained and the data is resegmented.

In order to avoid instability, the algorithm is run at most K times (K being the number of initial clusters). Doing so avoids clusters continuously split and merge the same segments over and over.

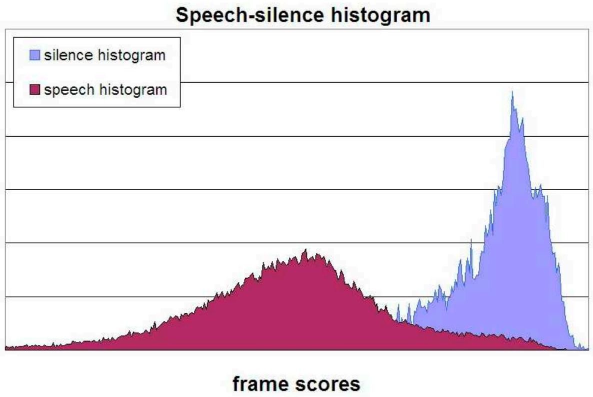


Fig. 2. *Speech-silence histogram for a full meeting*

3.2. Frame Level Purification

Due to the use of a minimum duration in the acoustic modelling, speech segments that legitimately belong to a particular cluster can be "infected" with sets of non-speech frames and frames belonging to other speakers. Such sets are too short to be taken into account by the segment-based decoding or eliminated by the model-based speech/non-speech detector. However, they cause the models to diverge from their acoustic modelling targets. This is particularly important when considering whether to merge two clusters. In speaker recognition it is common to use an energy-based Speech Activity Detection (SAD) where individual frames with energy below a threshold are rejected.

The frame level purification presented here focuses on detecting and eliminating the non-speech frames that do not help to discriminate between speakers (e.g. short pauses, occlusive silences, low-information fricatives, etc.). Given a set of acoustic vectors X that form a speaker cluster, we can separate it into two subsets: X_1, with frames that are likely to discriminate between speakers; and X_2, non-speech frames that we wish to eliminate.

Figure 2 shows the normalized histograms of the frame scores resulting from evaluating all data in a full meeting given a cluster model ($\mathcal{L}(X|\Theta_A)$) trained with this data. We separate the histogram in two, according to the reference file, between the speech frames and the non-speech frames. The scores of the non-speech frames are mainly located in the upper area. Some speech frames that also have a high score might be due to other non-speech frames that are labelled as speech in the reference. Even if we use a speech/non-speech detector, we have a residual error of around 5% of non-speech data that enters the clustering system. In order to purify a cluster, we need to eliminate as much of the non-speech frames as we can while maintaining the frames that discriminate between speakers.

Figure 3 illustrates a phenomenon that we observed when training a cluster model Θ_A, using M Gaussian mixtures, with acoustic data X. A subset (M_1) adapt their mean and variances to model the subset of speaker data (X_1), while another subset (M_2) appears to model the subset of the data (X_2) associated with non-speech information. Since the number of frames in X_1 is typically much bigger than those of X_2, $|M_1| >> |M_2|$ and, at times, $|M_2|$ may be 0 if the non-speech data is minimal. Furthermore, the variance of the non-speech Gaussian mixtures in M_2 is always much smaller than M_1. Given this, any non-speech frame evaluated by the model gets a higher score than a speech frame. We take advantage of this in the frame level purification algorithm.

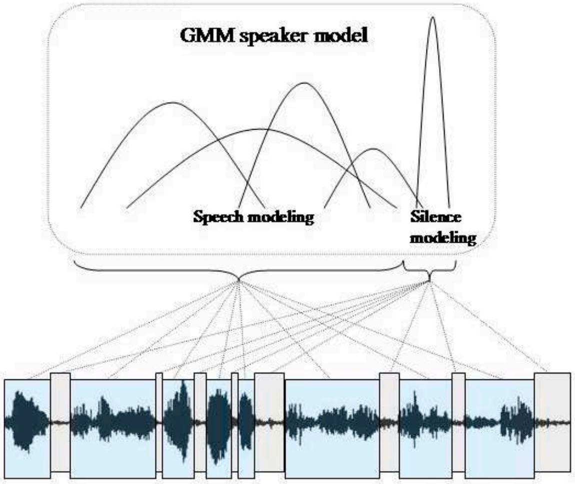


Fig. 3. *Observed assignment of frames to Gaussians*

$$\bar{\mathcal{L}}(x[i]|\Theta_A) = \frac{1}{Q} \sum_{j=-Q/2}^{Q/2-1} \sum_{m=1}^{\widetilde{M}} log\Big(W_A[m]\mathcal{N}_{A,m}(x[i+j])\Big) \qquad (1)$$

We consider two metrics to measure this phenomenon, both using equation 1 where Q is used to average the measure around the desired value; $W_A[m]$ is the mixture weight and $\mathcal{N}_{A,m}(x[i+j])(x[\cdot])$ is the result of evaluating $x[\cdot]$ on the gaussian mixture $\mathcal{N}_{A,m}(x[i+j])$:

Metric 1 A standard smoothed likelihood (over 100ms) of a frame, with $\widetilde{M} = M$ (all mixtures in model Θ_A) and smoothing factor $Q = 5$ (using 10ms acoustic frames).

Metric 2 The smoothed likelihood (over 100ms) on a smaller set of Gaussian mixtures that include the mixtures assigned to non-speech. We used the 50% of mixtures with smallest variance ($\widetilde{M} = M_{non-speech}$).

We apply the frame level algorithm when comparing two models using the ΔBIC metric in the following way:

1. Retrieve all frames assigned to each of two clusters and use either metric for each frame in both clusters.
2. Eliminate the P% of frames in each cluster with the highest smoothed values, where P is a value to be optimized according to the data.
3. Train two new models with the remaining data and use them for computing the ΔBIC metric.

4. EXPERIMENTS

We have tested the two purification algorithms presented here using three existing datasets that have been used in recent NIST Rich Transcription (RT) evaluations in the meetings domain [11]. The first two datasets are the RT04s evaluation and development sets, with a total of 16 meetings, an average duration of 10 minutes, and 1 to 10 channels available per meeting. We used this database as our development set. As an evaluation set, we used the RT05s set, with 10 shows and the same average characteristics as the development set. The evaluation set also has four meeting sources that did not exist in the development set.

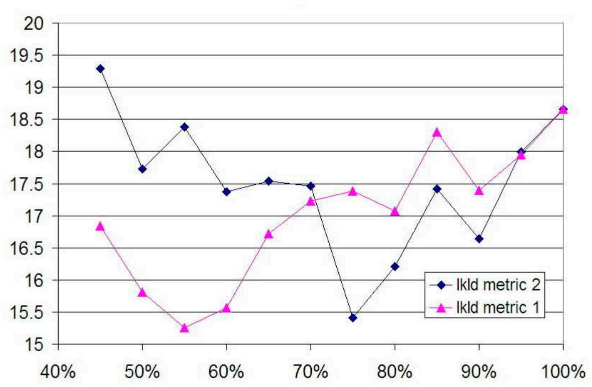


Fig. 4. *DER for different values of P for both frame-based metrics*

The system performance is measured in terms of Diarization Error Rate (DER) as it is used in the NIST RT Evaluations. An optimal one-to-one mapping of reference speaker identity to system output identity is performed, and the error is computed as the percentage of time that the system assigns the wrong speaker label.

For the frame purification algorithm, in Figure 4, we show the DER for different possible values of the P parameter for both proposed metrics on the development set. The optimum values are $P = 55\%$ for metric 1 and $P = 75\%$ for metric 2. We applied the algorithms to the evaluation data using these values.

For both algorithms, Table 1 shows the DER when applying them on the evaluation and development sets.

Purif. algorithm	Development	evaluation
baseline	18.66%	18.46%
Segment-based	16.96%	17.99%
Frame-based (using Energy)	18.40%	18.18%
Frame-based (metric 1)	**15.26%**	17.31%
Frame-based (metric 2)	15.41%	**15.09%**
Frame(met. 1)+Segment	16.50%	16.71%
Frame(met. 2)+Segment	17.78%	18.08%

Table 1. Comparison of the DER using different purification algorithms

The segment-based purification performs better in the development data than in the evaluation data, and in both cases it outperforms the baseline system. The Frame-based algorithm using metric 1 achieves the best performance on the development data, but using metric 2 we obtain more robust results in both datasets. We tried combining each of the frame-based metrics with the segment-based, and the DER is lower than the baseline but does not outperform using frame-based alone.

For comparison we also tried using a Speech Activity detector (SAD) filtering, as is commonly used in the speaker recognition community. When comparing two models we filter out the frames whole energy falls within the 5% of lowest energies of all the recording. The results in the development and evaluation set, although they achieve an improvement over the baseline, are worse that the presented purification systems.

5. CONCLUSION

In this paper, we present two novel techniques for cluster purification in a speaker diarization system using agglomerative clustering. The first technique allows cluster splitting within an agglomerative speaker clustering system by finding speech segments that are assigned to a cluster but are very different from it, and assigning them to a new cluster. The second technique detects and avoids using non-speech frames when comparing two clusters for merging or assessing the clustering stopping criterion. We show that both techniques work well on meetings data, achieving improvements in DER of more than 9% and 18% relative respectively.

6. ACKNOWLEDGEMENTS

We would like to thank Adam Janin, James Fung, Ascension Gallardo and Nikki Mirghafori for their help and helpful comments. This work was partially supported by the European Union 6th FWP IST Integrated AMI (Augmented Multi-Party Interaction, FP6-506811).

7. REFERENCES

[1] D.A. Reynolds and P. Torres-Carrasquillo, “Approaches and applications of audio diarization,” in *ICASSP’05*, Philadelphia, PA, March 2005, pp. 953–956.

[2] C. Barras, X. Zhu, S. Meignier, and J.-L. Gauvain, “Improving speaker diarization,” in *Fall 2004 Rich Transcription Workshop (RT04)*, Palisades, NY, November 2004.

[3] D.A. Reynolds and P. Torres-Carrasquillo, “The MIT Lincoln Laboratories RT-04F diarization systems: Applications to broadcast audio and telephone conversations,” in *Fall 2004 Rich Transcription Workshop (RT04)*, Palisades, NY, November 2004.

[4] R. Sinha, S. E. Tranter, J. J. F. Gales, and P. C. Woodland, “The cambridge university march 2005 speaker diarisation system,” in *European Conference on Speech Communication and Technology (Interspeech)*, Lisbon, Portugal, September 2005, pp. 2437–2440.

[5] Chuck Wooters, James Fung, Barbara Peskin, and Xavier Anguera, “Towards robust speaker segmentation: The ICSI-SRI fall 2004 diarization system,” in *Fall 2004 Rich Transcription Workshop (RT04)*, Palisades, NY, November 2004.

[6] Xavier Anguera, Chuck Wooters, Barbara Peskin, and Mateu Aguilo, “Robust speaker segmentation for meetings: The ICSI-SRI spring 2005 diarization system,” in *Rich Transcription 2005 Spring Meeting Recognition Evaluation*, Edinburgh, Great Brittain, July 2005.

[7] Dan Istrate, Corinne Fredouille, Sylvain Meignier, Laurent Besacier, and Jean-Francois Bonastre, “NIST RT05S evaluation: Pre-processing techniques and speaker diarization on multiple microphone meetings,” in *NIST 2005 Spring Rich Transcrition Evaluation Workshop*, Edinburgh, UK, July 2005.

[8] Steve Cassidy, “The macquarie speaker diarization system for RT05S,” in *NIST 2005 Spring Rich Transcrition Evaluation Workshop*, Edinburgh, UK, July 2005.

[9] David van Leeuwen, “The TNO speaker diarization system system for NIST RT05s for meeting data,” in *NIST 2005 Spring Rich Transcrition Evaluation Workshop*, Edinburgh, UK, July 2005.

[10] S. Shaobing Chen and P.S. Gopalakrishnan, “Speaker, environment and channel change detection and clustering via the bayesian information criterion,” in *Proceedings DARPA Broadcast News Transcription and Understanding Workshop*, Virginia, USA, Feb. 1998.

[11] “NIST rich transcription evaluations, website: http://www.nist.gov/speech/tests/rt,” .

[12] Jitendra Ajmera and Chuck Wooters, “A robust speaker clustering algorithm,” in *ASRU’03*, US Virgin Islands, USA, Dec. 2003.

[13] Xavier Anguera, Chuck Wooters, and Javier Hernando, “Speaker diarization for multi-party meetings using acoustic fusion,” in *IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, Puerto Rico, USA, November 2005.

[14] Jitendra Ajmera, Iain McCowan, and Herve Bourlard, “Robust speaker change detection,” *IEEE Signal Processing Letters*, vol. 11, no. 8, pp. 649–651, 2004.