FAST SPEAKER DIARIZATION BASED ON BINARY KEYS

Xavier Anguera[1] and Jean-François Bonastre[2]

[1]Telefonica Research, Barcelona, Spain
[2]University of Avignon, LIA, Avignon, France.
xanguera@tid.es, jean-francois.bonastre@univ-avignon.fr

ABSTRACT

Splitting a speech signal into speakers is the main goal of a speaker diarization system, which has become an important building block in many speech processing algorithms. Current state of the art systems are able to obtain good diarization error rates, but most of them are rather slow, which is a strong handicap in applications that require overall faster than real-time processing. In this paper we present a novel speaker diarization system which is built following a bottom-up agglomerative clustering approach and based on speaker binary keys, recently proposed for speaker modeling. After initialization, processing is entirely done over binary vectors and using exclusively binary metrics, which makes the system very fast. On tests performed using all conference meetings datasets released for the NIST RT evaluation campaigns we achieve diarization error rates just slightly worse than a classic acoustic-based system while running over 10 times faster.

Index Terms— speaker diarization, rich transcription, binary, discrete, discriminant

1. INTRODUCTION

Many current speech processing algorithms perform much better if they are queried with speech from a single speaker (*e.g.* speaker verification and identification) or with the different speakers already separated in order to perform, for example, speaker adaptation (*e.g.* speech recognition). Furthermore, speech indexing is greatly enhanced when the spoken transcripts can be associated with who spoke them. Speaker diarization is the algorithm concerned with such tasks, *i.e* the annotation of the audio signal with information on when each different speaker speaks. Such task is usually performed without prior knowledge of the identity or the number of speakers.

Among the approaches to speaker diarization that have received most continuity over recent years are the agglomerative clustering approaches [1] and the divisive approaches [2]. On the one hand, in the agglomerative clustering approach the speech signal is initially split into a number of clusters greater than the expected maximum number of speakers. Then the closest two clusters are iteratively merged until some clustering stopping criterion is met. On the other hand, divisive approaches start from a single cluster and iteratively create new ones until a stopping criterion is met.

In most current systems, that achieve state-of-the-art performances, speaker modeling is performed by using Gaussian Mixture Models (GMM) trained from Mel Frequency Ceptrum Coefficients (MFCC) extracted from the input speech using maximum likelihood (ML) or discriminative training techniques. Furthermore, many of these systems use the Bayesian Information Criterion (BIC) as a comparison metric between speakers and/or as stopping criterion, and the Viterbi algorithm in order to assign the feature vectors among the different clusters. All of the aforementioned algorithms impose a high computational load on the overall system, which, together with the iterative way they are applied, leads to accurate but very slow systems (over several times real-time). This becomes a problem when combining diarization with other systems in applications where efficiency is important.

For this reason, recently some efforts have been put into speeding up the speaker diarization processing to under real-time efficiencies [3, 4]. First, [3] showed that by avoiding the use of Viterbi decoding and applying some tricks on the iterative clustering process it is possible to lower the real-time factor of an agglomerative clustering algorithm to 0.97xRT. We think that such result is still too slow to allow for the combination of speaker diarization with other algorithms. Later on, in [4] the same research group was able to further lower the real-time factor to 0.07xRT by reimplementing some parts of the algorithm to be parallelized using a GPU. Although successful, this latest result depends on the usage of specialized hardware and the adaptation of the traditional algorithms to work on them, not being applicable for embedded devices or standard hardware.

In this paper we propose a totally novel speaker diarization system based on a speaker modeling technique recently proposed in [5], in which speaker clusters are modeled using a small vector of binary values. We use an agglomerative clustering approach and propose novel algorithms for background model training, clustering initialization and reassignment of segments, and adapt a stopping criterion proposed in [6] to work in our case. In experiments performed on *all* available NIST-RT meetings datasets our system shows performances just slightly above baseline acoustic-based results, with an efficiency comparable to GPU-based speeds but using a single CPU in standard hardware. Given the novelty of most modules in the system we consider these results still preliminary and expect to soon bring performance to state-of-the-art and also to further improve efficiency.

2. BINARY SPEAKER DIARIZATION ALGORITHM

The proposed speaker diarization algorithm consists of two very distinct processing blocks, as shown in Figure 1, namely the acoustic block and the binary block. The acoustic processing block is used for the initialization of the system and transformation of acoustic features into the binary domain. First, it performs standard acoustic feature extraction on the input signal to obtain MFCC features. Then a GMM-like model which we call KBM (binary Key Background Model) is trained from the data as explained in section 2.1 in order to acoustically model all speakers in the recording. Next, the KBM and acoustic features are used to obtain an initial rough clustering of the data into N_{init} clusters as explained in section 2.2. Finally, the

During the development of this work X. Anguera was partially funded by the Torres Quevedo Spanish program

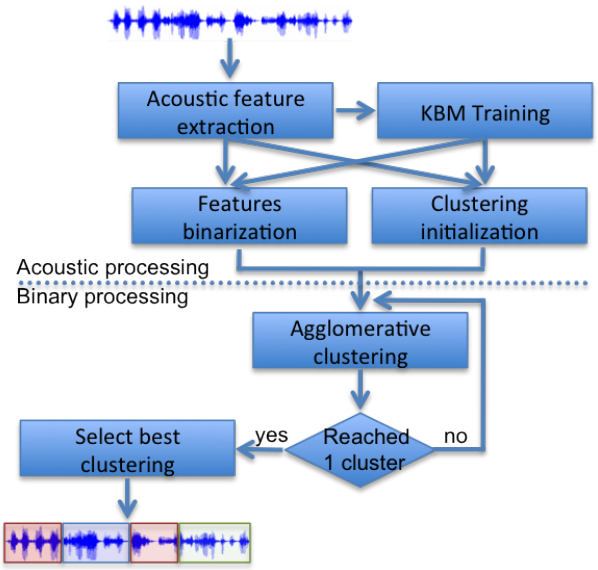

Fig. 1. *Main steps in the proposed speaker diarization system.*

"features binarization" block transforms the acoustic features into a binary representation by computing, for every feature vector, the set of N_G Gaussians with highest likelihood score given the KBM. This step, together with the KBM training, are the most computationally expensive in the system, but given that the background model does not change for the entire processing, they are only done once and then are reused throughout the rest of the diarization.

The binary processing block performs an iterative bottom-up clustering inspired by the original ICSI system in [1], but performed entirely in the binary domain. Given an initial rough clustering obtained in the initialization step, an iterative clustering is performed to bring the number of clusters from N_{init} to 1. The process goes as follows: a) Compute the binary signatures for all current clusters; b) Reassign all data among all existing signatures and retrain them using the new clustering; c) Compare all signatures with each other and merge those two with highest similarity, creating a new signature for the resulting cluster; d) save the resulting clustering and go back to (a) if the number of clusters is > 1. Once we reach one cluster, a metric inspired by that proposed in [6] is computed for each of the possible clusterings and the one with the highest value is output.

2.1. Obtaining the Binary Key Background Model

An acoustic model we call binary-Key Background Model (KBM) is used in the diarization system to convert the acoustic features into binary features. The use of a KBM for speaker binary modeling is first explained in [5] although for diarization we train it in a different way given the available data. In [5] a set of anchor speakers was used to obtain mean-adapted GMM models from an initial Universal Background Model (UBM), which were then pooled together to form the KBM. In diarization a similar procedure could be followed by training a KBM using speaker specific data outside of the tested recording, although performance could suffer due to the mismatch in acoustic background conditions between training and test sets. Instead, we propose a novel method that allows for the training of the KBM directly from the test data. One could argue that a standard GMM could be used instead (for example training it via iterative Gaussian splitting on the input data). As shown in [5] for speakerID, and in the experimental section below for speaker diarization, the proposed approach produces models that are able to generate much more discriminative binary keys than with a standard GMM training.

In order to obtain the KBM, first single Gaussians are trained for every 2 seconds of data (with 50% overlap). These parameters were set to guarantee that each Gaussian would be acoustically centered in a speaker (and not on uttered sounds, which last much less) and to cover all the acoustic space in the recording. The resulting pool of Gaussians (600 Gaussians for a 10 minutes meeting excerpt) is thought to cover the overall acoustic space from the test data while centered on particular acoustic events/speakers in it. Note also that training these single Gaussians is quite fast. Next, we select a subset of N Gaussians from this pool to conform the KBM model. The chosen Gaussians are meant to retain full coverage of the data's acoustic space and to be most discriminant among each other. We found a single-linkage clustering strategy to work best to achieve this goal. We first define a global dissimilarity vector v_{KL2} to represent the distance between the selected Gaussians to all others in the pool and initialize it to ∞ as no Gaussians are initially selected. The selection algorithm then works as follows: a) select an initial Gaussian from the pool as the one which best models the 2s segment it was trained from (*i.e.* $\underset{i}{\text{argmax}}\ Lkld(s_i|\theta_i)$ where s_i is the ith segment data and θ_i is a single Gaussian trained on it); b) Compute the KL2 (symmetrized Kullback-leibler) divergence $S_{KL2}(\theta^{'}, \theta_j)$ between the previously selected Gaussian $\theta^{'}$ and the rest of Gaussians θ_j still not selected from the pool, and set $v_{KL2}[j] = min(v_{KL2}[j], S_{KL2}(\theta^{'}, \theta_j))$; c) Add to the KBM the Gaussian θ^k with highest single linkage dissimilarity with those already selected (*i.e.* highest $v_{KL2}[k]$); d) go back to (b) until the desired number of Gaussians in the KBM (N) is reached.

2.2. Clustering Initialization

The goal of the clustering initialization step in an agglomerative clustering approach is to obtain an initial set of N_{init} clusters containing acoustically homogeneous segments of data. In the literature there have been extensive efforts [7, 8, 9] to find a meaningful initial clustering, but an accurate and fast solution is yet to be found. In this paper we put forth a method derived from the concept of modeling the speakers acoustic space with a KBM, which works quite well in our system. It is straightforward, but left for future work, to test this method on a acoustic agglomerative clustering system.

The proposed initialization relies on the order in which Gaussians have been selected for the KBM model. As the goal of the Gaussian selection process used for the KBM is to choose, at every iteration, the Gaussian which best complements the preexisting ones in covering the acoustic space, by selecting only a few of the initially selected Gaussians we can obtain a rough acoustic modeling of all our data (the more Gaussians selected, the more fine grained the modeling will be). We therefore use the N_{init} first selected Gaussians in the KBM as seed models, where N_{init} is the initial number of clusters we desire. We then use these Gaussians to bootstrap an initial clustering into N_{init} clusters by sequentially assigning acoustic segments, with duration set to 100ms, to the cluster whose Gaussian is evaluated with highest likelihood. This generates a highly over-segmented initial clustering where multiple clusters will probably alternate in modeling the same speaker. Note that for this step we require the computation of the likelihood of every frame given every Gaussian, which is also required in section 2.3. We therefore only need to compute it once.

2.3. Speaker modeling using a binary Key

The speaker binary keys were initially introduced in [5] and shown to be effective in distinguishing among speakers. In this paper we show how they can be effectively applied for speaker diarization, bringing about important computational savings. Let us recall how to obtain them from the KBM model as discussed in [5]. A speaker binary key is an N-dimensional binary vector, $\mathbf{v}_f = \{v_f[1], \ldots, v_f[N]\}$, $v_f[i] \epsilon \{0, 1\}$ where N is the number of Gaussian mixtures in the KBM model. Setting any position in $v_f[i]$ to value 1(TRUE) indicates that the ith Gaussian in the KBM coexists in the same region of the acoustic space with the acoustic data being modeled.

The first necessary step in obtaining speaker binary keys is the evaluation of the acoustic features on the KBM model, obtaining a matrix $\mathbf{V}_G$ composed of as many rows as acoustic features we have, and N_G columns, indicating the Gaussian ID's for the best N_G matching Gaussians in the KBM for each acoustic feature. As explained above, this step needs to be performed only once during the initialization, and then can be reused every time a new binary key is requested. The number N_G of Gaussians chosen per acoustic feature is constant and set as a percentage of the total number of Gaussians (N) in the KBM. Like in [5] we set it to $N_G = 0.01N$.

The second step involved the transformation of an utterance, ranging from frame i_1 to frame i_2 into a binary key. In order to do so we define the accumulator vector $\mathbf{v}_c = \{v_c[1], \ldots, v_c[N]\}$, $v_c \epsilon \mathbb{N}^1$ initialized to 0, where each position $v_c[i]$ represents the same Gaussian Mixture from the KBM as $v_f[i]$. Then, for every Gaussian ID $V_G[i, j]$ with $i = i_1 \ldots i_2$ and $j = 1 \ldots N_G$ we increment the accumulator vector $v_c[V_G[i, j]] + +$.

When all frames have been processed, each position $v_c[j]$ in the accumulator vector contains the relative importance of Gaussian j in modeling the utterance we have processed. The conversion from $\mathbf{v}_c$ to $\mathbf{v}_f$ is straightforward by setting the positions in $\mathbf{v}_f$ to 1, according to the top 20% values in $\mathbf{v}_c$, and to 0 otherwise. Intuitively, the binary-keys modeling algorithm projects the acoustic location of each acoustic frame from the feature space into the space of KBM Gaussians and saves only those components with highest impact.

2.4. Cluster Comparison and Data Reassignment

The comparison between any two binary keys $\mathbf{v}_{f1}$ and $\mathbf{v}_{f2}$ is a very fast operation as it only involves bit-wise comparisons between the two vectors. Multiple possible metrics can be devised, with varying performances. For this paper we have found the similarity in Eq. 1 to work best.

$$S(\mathbf{v}_{f1}, \mathbf{v}_{f2}) = \frac{\sum_{i=1}^{N} (v_{f1}[i] \wedge v_{f2}[i])}{\sum_{i=1}^{N} (v_{f1}[i] \vee v_{f2}[i])} \quad (1)$$

where $\wedge$ indicates the boolean AND operator and $\vee$ indicates the boolean OR operator. Note that Eq. 1 can be used between two cluster keys or between a cluster key and a key obtained from a single feature segment.

Like in the ICSI diarization system described in [1], after every clustering iteration a data reassignment is performed in order to refine the segmentation and to reallocate data to the closest cluster. In the proposed algorithm such assignment is inspired by [3]. For every 1 second segment we compute its binary key by extending it by 1 second on either side (totaling 3 seconds of data). We then assign the segment to the cluster whose similarity is maximized. Note that by using a fixed segment assignment length the binary keys from the input data only need to be computed once throughout all the diarization process.

2.5. Final Clustering Selection

The proposed speaker diarization system iteratively merges the closest two clusters until it reaches a single cluster, storing the resulting clustering for each step. Then, the optimum clustering is chosen by using the T-test T_s metric proposed in [6] by adapting it to our case. Given the segment assignment algorithm described in section 2.4, every clustering C^i is composed by equal-sized segments distributed among the different clusters. First, we compute the statistics of intra-cluster and inter-cluster similarity distributions, *i.e.* the distributions of all similarities between binary keys obtained from segments in the same cluster and between all binary keys from segments in different clusters. Note that both the binary keys and the similarities can be computed only once and used for all tests. Then, assuming that both distributions are Gaussian-shaped, we obtain T_s using Eq. 2.

$$T_s = \frac{m_1 - m_2}{\sqrt{\frac{\sigma_1^2}{n_1} + \frac{\sigma_2^2}{n_2}}} \quad (2)$$

where m_1, σ_1, n_1, m_2, σ_2, n_2 are respectively the mean, standard deviation and size of the intra-cluster and inter-cluster distributions. Finally, we select the clustering that maximizes the T_s value.

3. EXPERIMENTS AND RESULTS

3.1. Databases and Evaluation setup

Like in research published in recent years, we evaluate the proposed speaker diarization system using the NIST Rich Transcription (RT) conference meeting recordings. Given the reported "flakiness" of scores obtained by speaker diarization systems we decided to evaluate our system using all available data released by NIST over the 4 years it conducted RT evaluations (namely RT05, RT06, RT07 and RT09) totaling 32 meeting excerpts in total and around 11.6 hours of evaluated data. Note that from the original NIST datasets we excluded a NIST excerpt from RT05 and the TNO excerpt from RT06 as they were reported faulty by other researchers in the field.

For each dataset we compute the Diarization Error Rate (DER), which is the most standard metric used in diarization, measuring the percentage of the overall time in which data is not given the right label (a label being either an individual speaker, multiple speakers, or non-speech). In the reported DER we include errors coming from non-speech and from overlapped speech, even though we are not doing anything in this paper to detect the later. In addition to the DER we also report the real-time (xRT) factor, computed as the ratio between the time it takes to run the diarization (excluding speech activity detection –SAD– and feature extraction) and the total speech-labelled time output by the SAD system (which accounts for $34,520$ sec. in our test data).

For comparison purposes we compute the same metrics for an in-house implementation of the system in [1] that we use as acoustic baseline. All experiments were run on MacBook Pro equipped with an Intel Core 2 Duo 2.53GHz with 3Mb Cache and 8Gb RAM, where all processing was done in a single core. All algorithms were implemented in C and compiled using full optimizations. For both systems we extracted 19-order MFCC every 10ms using the HTK toolkit over the beamformed output in the MDM evaluation condition. Finally, we borrowed the SAD (Speech Activity Detection) labels from Univ. Avignon/Eurecom's system, obtained as described in [10].

3.2. Evaluation Results

Figure 2 shows the DER scores of the proposed system for different values of N. Recall that N is the number of Gaussians used in the

KBM and therefore the number of bits in the speaker binary key. Although results for the individual datasets are quite "flaky" we can see that the time-weighted average converges around 27% for $N > 500$ and remains constant thereafter. For $N < 500$ the binary key for any given speaker is not discriminant enough to differentiate it from other speakers. We choose two working points, $N = 512$ as the fastest system, and $N = 896$ as the optimum.

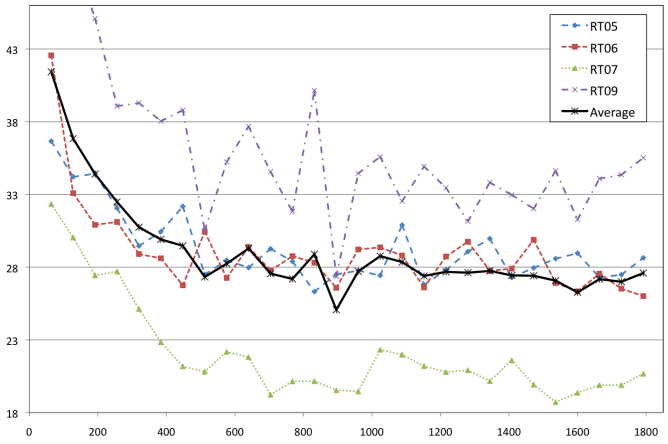

Fig. 2. *Diarization Error Rate (DER) as a function of N.*

Table 1 shows DER and real-time factor results for the baseline system as well as for the proposed system under several different configurations. The first row indicates the baseline results, which runs in over 41k seconds, *i.e.* 1.19xRT. Next rows show two different configurations of the proposed system to analyze how some of its modules work. In the first one the KBM is replaced by a standard GMM, trained using a divisive approach over all data by using EM. We see how results are far from optimal, implying the capital importance of using a good KBM for the binarization of the acoustic data. Next, we show results obtained by our system where for each excerpt we manually chose the number of final clusters with optimum DER. We see how results here are in all cases better than the acoustic baseline, indicating that the diarization via binary keys can indeed outperform the acoustic-based diarization. This test does not entail that the T_s metric is not effective in choosing the best clustering, as in many cases the optimum and resulting DER were the same or very similar. Finally, the last two rows show results for the complete system, in the previously selected working points. The optimum system is quite fast and less than 2% worse than the baseline, while the fastest system achieves 0.103xRT while performing $\sim 4\%$ worse in DER than the baseline. A comparison for each individual meeting except between the system output with $N = 896$ and the baseline shows that 44% of the meetings achieve better DER scores using binary keys.

Table 1. *Comparison of Results on DER and real-time factor*

System	RT05	RT06	RT07	RT09	Ave.	xRT
Acoustic Baseline	24.96	24.32	17.39	26.80	23.23	1.19
Standard GMM	40.06	40.08	35.88	41.20	39.24	–
Optimum Clustering	23.89	18.71	16.87	26.94	21.37	–
Binary Keys 896G	27.50	26.58	19.54	27.36	25.06	0.175
Binary keys 512G	27.50	30.44	20.81	30.54	27.32	**0.103**

Finally, we compared the real-time factor achieved by our system with other current state-of-the-art systems. Note that such speed results can not always be directly compared due to the differences in the machines being used for the processing. To the best of our knowledge, most current diarization systems display real-time factors well above 1. One exception is the ICSI system in [3] where by applying several tricks on their regular system they achieved 0.88xRT using a single core on an Intel Xeon 2.8 GHz machine. Later on, in [4] a CPU/GPU system running as fast as 0.07xRT was proposed, although this system is heavily tuned to use all processing power from multiple parallel processors available in the GPU and therefore out of the scope of our objectives.

4. CONCLUSIONS AND FUTURE WORK

In this paper we propose a novel speaker diarization algorithm having speed as its strongest feature. Current state-of-the-art systems have achieved decent diarization error rates, although they are very slow and thus are not very suitable as processing modules for bigger systems where speed is an issue. Our proposal deviates completely from the classical acoustic modeling of speakers by applying a recently proposed approach to speaker modeling through binary keys, which are very fast to compute and can effectively differentiate between speakers. Acoustic modeling is then limited to the initialization step. In tests performed using all NIRT-RT meetings datasets we achieve only slightly worse error rates than a well known acoustic-based implementation, but our system is up to 10 times faster. As this is a totally novel system we believe there is still room for plenty of improvements in the system which we hope will soon be matching state-of-the-art performances.

5. REFERENCES

[1] Jitendra Ajmera and Chuck Wooters, "A robust speaker clustering algorithm," in *Proc. of IEEE Workshop on Automatic Speech Recognition and Understanding*, US Virgin Islands, USA, Dec. 2003.

[2] S. Meignier, J.-F. Bonastre, and S. Igounet, "E-HMM approach for learning and adapting sound models for speaker indexing," in *Proc. Odyssey Speaker and Language Recognition Workshop*, Chania, Creete, June 2001, pp. 175–180.

[3] Y. Huang, O. Vinyals, G. Friedland, C. Muller, N. Mirghafori, , and C. Wooters, "A fast-match approach for robust, faster than real-time speaker diarization," in *Proc. IEEE Workshop on Automatic Speech Recognition and Understanding*, Kyoto, Japan, December 2007, pp. 693–698.

[4] G. Friedland, J. Ching, and A. Janin, "Parallelizing speaker-attributed speech recognition for meeting browsing," in *Proc. IEEE International Symposium on Multimedia (to appear)*, Taichung, Taiwan, December 2010.

[5] Xavier Anguera and Jean-Francois Bonastre, "A novel speaker binary key derived from anchor models," in *Proc. Interspeech*, 2010.

[6] Trung Hieu Nguyen, Eng Siong Chng, and Haizhou Li, "T-test distance and clustering criterion for speaker diarization," in *Proc. Interspeech*, 2008.

[7] Xavier Anguera, Chuck Wooters, and Javier Hernando, "Friends and enemies: A novel initialization for speaker diarization," in *Proc. ICSLP*, Pittsburgh, USA (to appear), September 2006.

[8] G. Friedland, O. Vinyals, Yan Huang, and C. Muller, "Prosodic and other long-term features for speaker diarization," *IEEE TASLP*, vol. 17, no. 5, pp. 985–993, July 2009.

[9] Tin Lay Nwe, Hanwu Sun, Bin Ma, and Haizhou Li, "Speaker diarization in meeting audio for single distant microphone," in *Proc. Interspeech*, Makuhari, Japan, 2010.

[10] C. Fredouille and N. Evans, "The influence of speech activity detection and overlap on the speaker diarizationfor meeting room recordings," in *Proc. Interspeech*, September 2007.