The Intelligent Voice System Description for the First DIHARD Challenge

Abbas Khosravani, Cornelius Glackin, Nazim Dugan, Gérard Chollet, Nigel Cannings

Intelligent Voice Limited, St Clare House, 30-33 Minories, EC3N 1BP, London, UK

1. Abstract

This document describes the Intelligent Voice (IV) speaker diarization system for the first DIHARD challenge. The aim of this challenge is to provide an evaluation protocol to assess speaker diarization on more challenging domains with the speech across a wide array of challenging acoustic and environmental conditions. We developed a new frame-level speaker diarization built on the success of deep neural network based speaker embeddings, known as *d*-vectors, in speaker verification systems. In contrary to acoustic features such as MFCCs, frame-level speaker embeddings are much better at discerning speaker identities. We perform spectral clustering on our proposed LSTM-based speaker embeddings to generate speaker log likelihood for each frame. A HMM is then used to refine the speaker posterior probabilities through limiting the probability of switching between speakers when changing frames.

2. Data resources

Switchboard corpora (LDC2001S13, LDC2002S06, LDC2004S07, LDC98S75, LDC99S79) which consists of conversational telephone speech data from around 2.5k speakers were used to train our LSTM-based neural network. We had limited time to extend the training of our system on larger corpora and with challenging domain, however, reasonable results has been obtained using only these corpora.

3. Detailed description of algorithm

3.1. Acoustic features

For speech parameterization we used 40-dimensional filterbank features. These features are extracted at 8kHz sample frequency using Librosa toolkit with 32 ms frame length and 10 ms overlap. For each utterance, the features are centered using a short-term (3s window) cepstral mean and variance normalization (ST-CMVN).

3.2. Frame-level embeddings

The *i*-vector based systems have been the dominating approach for both speaker verification and diarization applications. However, with the recent success of deep neural networks, a lot of efforts have been made into learning fixed-dimensional speaker embeddings (*d*-vectors) using an end-to-end network architecture that could be more effective relative to *i*-vectors on short segments [1, 2, 3, 4]. We employed a generalized end-to-end model using an LSTM-based neural network [2]. The network architecture is shown in Fig 2. It consists of a stacked bidirectional LSTM with a projection layer. The LSTM layers map the input sequence of feature vectors into a sequence of speaker embeddings. An average layer followed by a length-normalization layer can produce a fixed dimensional representation for the input segment. Training is based on processing a large number of utterances in the form of a batch that contains N speakers, and M utterances. Each utterance could be of arbitrary duration. But to train the network in batch, they need to be of the same duration. To handle this we zero pad each segment to a fixed duration and then use a masking value to skip zero time-steps. This way we can train the network on variable length speech segments. We used variable length speech segments ranging from 3-5 seconds without overlap and construct batches with 60 speakers, each having 10 different segments. In the loss layer, a generalized end-to-end (GE2E) loss builds a similarity matrix that is defined based on the cosine similarity between each pair of input utterances. During the training, we want the embedding of each utterance to be similar to the centroid of all that speaker's embeddings, while at the same time, far from other speakers' utterances. A detailed description of GE2E training can be found at [3].

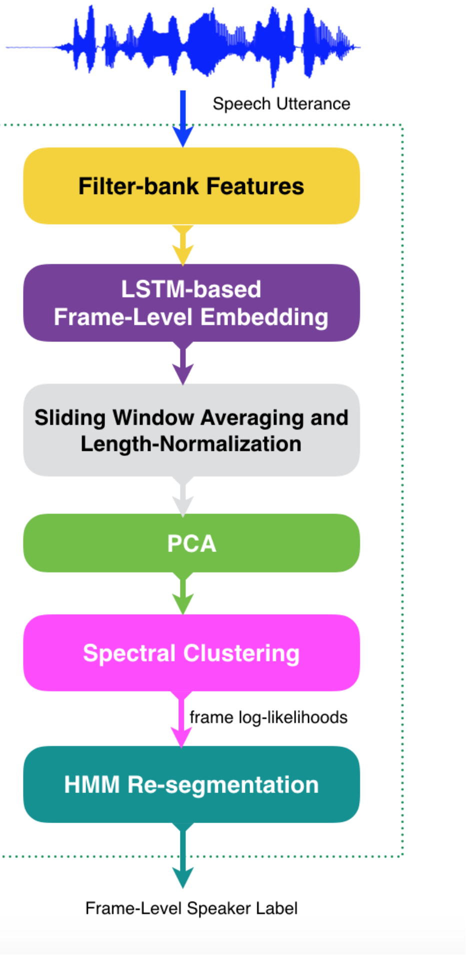
Speech Utterance
Filter-bank Features
LSTM-based Frame-Level Embedding
Sliding Window Averaging and Length-Normalization
PCA
Spectral Clustering
frame log-likelihoods
HMM Re-segmentation
Frame-Level Speaker Label

Figure 1: *System diagram for the proposed diarization system.*

In the test phase, we remove the average as well as the

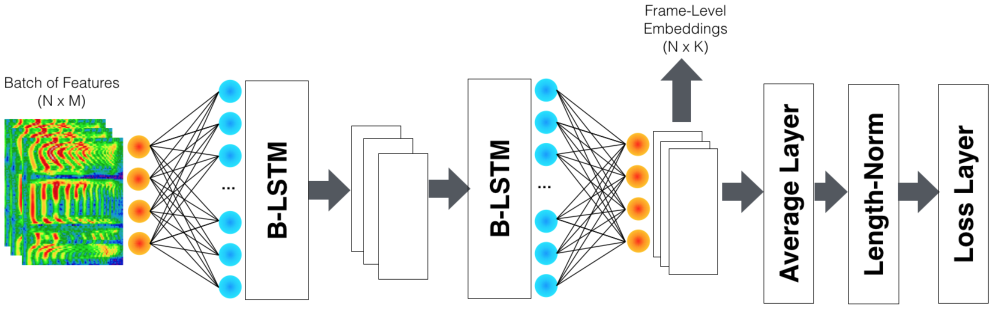

Figure 2: *LSTM-based neural network architecture used to extract frame-level speaker embeddings.*

length-normalization layer to produce frame-level embeddings as shown in Fig 2. The whole test utterance will be fed into the network to produce a sequence of speaker embeddings corresponding to each frame of the input (for long utterances we use a sliding window of 20 seconds long with 10 seconds overlap and average the results). A Speech Activity Detection (SAD) may be used to feed only the speech portion of the utterance to the network. In our experiment with conversational telephone speech, the word boundaries generated by an ASR is used as speech segments. Due to the fact that the network is trained on length-normalized average of frame-level embeddings, we incorporate a sliding window of a few frames to average and length-normalize each frame. Our experiments indicates that a window of 30-50 frames produces the best performance. Finally, a principle component analysis (PCA) is incorporated to reduce the dimensions of the resulting length-normalized embeddings (we used 5 dimension in our experiments) so as to be ready for clustering.

3.3. Speaker estimation

To estimate the number of clusters a simple heuristic based on the eigenvalues of the affinity matrix is used [5].

3.4. Clustering method

We employed a spectral clustering which is able to handle unknown cluster shapes. It is based on analyzing the eigenstructure of an affinity matrix. A more detailed analysis of the algorithm is presented in [6]. We used an Euclidean distance measure to form a nearest neighbor affinity matrix on the frame-level embeddings. To mitigate the computational complexity of the spectral clustering, especially when the number of frames are too large, we can employ sampling at a specific rate.

3.5. Re-segmentation details

The clustering algorithms are typically followed by a re-segmentation algorithm that refines the speaker transition boundaries. This could be either in the feature space like MFCC or in the factor analysis subspace [7]. Speaker diarization in factor analysis space allows us to take advantages of speaker specific information. However, the effectiveness of this technique is proportional to the length of the speech segment and thus is not suitable for spontaneous speech scenario, especially in conversational speech with fast speaker turn changing. By contrast, lower-level acoustic features such as MFCCs are not quite as good for discerning speaker identities, but can only provide sufficient temporal resolution to witness local speaker changes. The proposed framework for diarization provides a stronger speaker representation at the frame level, making it more suitable for spontaneous speech with fast speaker turn changes. As a result, when combined with an HMM to refine the speaker posterior probabilities through limiting the speaker transitions [7], the system is able to detect very short turn changes. The speaker log likelihoods for the HMM are computed by the spectral clustering algorithm as described in the previous section.

4. Hardware requirements

We used two Intel Xeon CPU (E5-2670 @ 2.60GHz and 8 cores), 64G of DDR3 memory, 200G disk storage and an NVIDIA TITAN X GPU (12G of memory) to train our network. We used keras API with tensorflow backend for system development. Training time takes almost a week to process around half a million segments of 3-5 seconds long. To process a single 10 minute recording the system execution times is 17 seconds, that is more than 35 times faster than real time on multi-core CPU and GPU.

5. Conclusions

We proposed a frame-level speaker diarization framework that operates in the deep neural network embedding space. We found that, spectral clustering algorithm followed by an HMM to constrain the speaker transitions, contributes to the success of this framework.

We evaluated our proposed approach the first DIHARD diarization evaluation challenge data. Both the development and evaluation data focus on diarization on challenging corpora. Our system was trained on a small set of conversational speech data which totally differs from both the development and evaluation data. This is due to the limited access to the training data proposed by the evaluation plan and limited time to train the network on more corpora. We obtained a diarization error rate of 32.15% and 36.73% on the DIHARD development and evaluation sets. However, the results indicate the effectiveness of the proposed approach on challenging domains.

6. References

[1] D. Garcia-Romero, D. Snyder, G. Sell, D. Povey, and A. McCree, "Speaker diarization using deep neural network embeddings," in *Acoustics, Speech and Signal Processing (ICASSP), 2017 IEEE International Conference on*. IEEE, 2017, pp. 4930–4934.

[2] J. Rohdin, A. Silnova, M. Diez, O. Plchot, P. Matejka, and L. Bur-

get, “End-to-end dnn based speaker recognition inspired by i-vector and plda,” *arXiv preprint arXiv:1710.02369*, 2017.

[3] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, “Generalized end-to-end loss for speaker verification,” *arXiv preprint arXiv:1710.10467*, 2017.

[4] C. Li, X. Ma, B. Jiang, X. Li, X. Zhang, X. Liu, Y. Cao, A. Kannan, and Z. Zhu, “Deep speaker: an end-to-end neural speaker embedding system,” *arXiv preprint arXiv:1705.02304*, 2017.

[5] S. Shum, N. Dehak, and J. Glass, “On the use of spectral and iterative methods for speaker diarization,” in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[6] A. Y. Ng, M. I. Jordan, and Y. Weiss, “On spectral clustering: Analysis and an algorithm,” in *Advances in neural information processing systems*, 2002, pp. 849–856.

[7] G. Sell and D. Garcia-Romero, “Diarization resegmentation in the factor analysis subspace,” in *Acoustics, Speech and Signal Processing (ICASSP), 2015 IEEE International Conference on*. IEEE, 2015, pp. 4794–4798.