PRE-TRAINING OF SPEAKER EMBEDDINGS FOR LOW-LATENCY SPEAKER CHANGE DETECTION IN BROADCAST NEWS

Leda Sarı[⋆], *Samuel Thomas*[†], *Mark Hasegawa-Johnson*[⋆], *Michael Picheny*[†]

[⋆] Department of Electrical and Computer Engineering, University of Illinois at Urbana-Champaign, USA
[†] IBM T. J. Watson Research Center, Yorktown Heights, USA

ABSTRACT

In this work, the aim is to investigate pre-training of neural network based speaker embeddings in low-latency speaker change detection. Our end-to-end system takes two speech segments as inputs, processes them using shared Siamese layers to generate embeddings and then classifies the concatenated embeddings into same or different classes based on the speaker identities of the inputs. We investigate gender classification, contrastive loss and triplet loss based pre-training of the embedding layers and also experiment with joint training of the embedding layers along with the same/different classifier. Training is performed on 2-second single speaker segments based on the ground truth speaker segmentation. In order to speed up the test process, we take our test pairs around automatic speech recognition (ASR) based segmentation boundaries which causes a mismatch between training and testing. In addition, ASR segments tend to be shorter in which case we also operate with shorter than 2-second segments. In our experiments on the Broadcast News dataset, we showed that although i-vector based classifier performs well, triplet loss based pre-training followed by joint training of the embedding and classification layers results in better F-measure in both matched and mismatched conditions. In addition, the degradation in performance is less severe for network based embeddings as compared to using i-vectors in the variable duration test conditions.

Index Terms— Speaker change detection, sequence embedding, Siamese networks

1. INTRODUCTION

Speaker change detection is the task of finding the time instances when a different speaker starts to speak. One general approach to solve this problem is to extract features from sliding windows and compare the representations of the consecutive windows using a distance measure.

Non-neural-network approaches, either extracts features from two consecutive segments and compares the distance between them to a threshold to make a change decision or the system fits a model two the features of individual segments and their concatenation and chooses the hypothesis with higher score, this score can be Bayesian information criterion (BIC) [1] or likelihood [2].

One of the most commonly used features to represent the speaker characteristics of a speech segment is i-vector [3]. Although they have been successfully used in speaker verification applications, reliability of i-vectors depend on segment duration [4, 5]. In order to solve this problem, short speech segments are clustered using BIC, Gaussian divergence [6] or x-means algorithm [7, 8]. However, clustering methods are mainly designed for offline processing and therefore cannot be used in low-latency applications [7]. Recently, neural network based speaker embeddings are used as an altenative [9, 10] or as a complementary feature [11] for i-vectors. Studies have shown that network based embeddings can achieve better performance than using BIC based approaches on mel-frequency cepstral coefficients (MFCCs) [9, 10] or filterbank coefficients [12]. In [11], network embeddings are used in speaker classification task with a probabilistic linear discriminant analysis backend and it is shown that the embeddings achieve better performance than i-vectors especially when the inputs are short (10s). These embedding networks are trained using multiclass cross-entropy for speaker classification using large number of speakers [11, 12], using contrastive loss on two inputs processed in a Siamese architecture [13] or using triplet loss [9]. In order to map variable length sequences to fixed dimensional embeddings, they usually utilize long short-term memory (LSTM) [14] layers and averaging over time.

In addition to generating embeddings, neural networks are also used in end-to-end speaker change detection systems [15, 16, 17, 10] where the change decision is made at the end of a network instead of thresholding a distance measure. These systems can be classified into cases where the problem is reduced to taking two speech segments as input and comparing them [15, 10] or deciding if there is a change point within a given single speech segment [16, 17]. The networks that compare two segments usually have a Siamese structure where the initial few layers processing the two inputs share their weights. A similar structure is also usually used in embedding generating networks where the training objective consists of comparing the features extracted from the shared Siamese layers.

In this work, we combine learning speaker embeddings with an end-to-end approach for speaker change detection. We feed the Siamese embeddings of two segments into a fully-connected classifier for speaker change detection. We use embedding learning as a pre-training method for these Siamese layers. We investigate three pre-training mechanisms, namely gender classification which a simplified version of speaker classification, contrastive loss and triplet loss with both cosine and Euclidean distance measures. After pre-training, we either freeze the embedding layers and train the classifier alone or we jointly update them. In order to handle variable length segments, we use bidirectional LSTM (BLSTM) layers in the Siamese part. In earlier studies, similar loss functions have been used for embedding generation and then distance based change detection is applied [9, 12] or the change detection is performed by a network without explicitly training the intermediate speaker embeddings [10]. Here our aim is to use pre-training to get better intermediate features for an end-to-end speaker change detection system.

Our second aim is to operate in the low-latency regime which limits our input segment duration to 2s. As concluded in [7], large improvements in the speed of speaker diarization module can be obtained if automatic speech recognition (ASR) based decision boundaries are used. Based on this observation, our test setup considers comparisons only around segment boundaries. During training, we

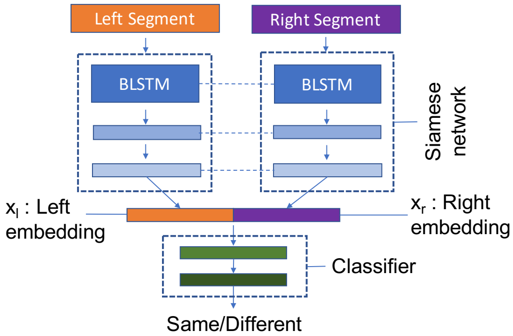


Fig. 1. Overview of the change detection network

have manually annotated speaker boundaries so our training segments are extracted based on those boundaries. In our tests, we use (a) matched condition to the training setup where the decisions are made around ground truth segments, (b) mismatched condition where the decision boundaries around which we take our two inputs are obtained from ASR. Especially the latter test condition also causes another mismatch due to shorter segment durations.

Although triplet loss training of [9] can handle variable length duration, they do not report the effect of duration mismatch between training and test segments and their change detection is based on distance thresholding. The end-to-end approach of [10] has a similar architecture to our network except they take the absolute difference after Siamese layers rather than concatenating them but they do not employ pre-training and their tests also only include fixed length segment comparison on synthetically concatenated speech segments. Both of these studies use sliding window approach but as discussed above, we use ASR based segment boundaries in our tests for low-latency applications. In [2], ASR phone boundary based tests are performed with 2s segments for fast speaker change detection but they use a non-neural-network approach. Therefore, our main contributions are (1) incorporating existing speaker embedding learning methods into an end-to-end system by pre-training, (2) comparison of speaker change detection based on i-vectors and neural network embeddings under mismatched test segment duration and (3) evaluation of change points based on ASR segment boundaries for low-latency applications.

The rest of the paper is organized as follows: In Section 2 we introduce our change detection system and describe the pretraining of the Siamese layers in the network. In Section 3, we present experimental setups and results, then we conclude the paper in Section 4.

2. SPEAKER CHANGE DETECTION

Figure 1 gives an overview of the speaker change detection network. Two speech segments are passed through shared Siamese layers and generate the embeddings. These embeddings are then concatenated and fed into the classifier which is a fully-connected network. Therefore, instead of extracting features from neural networks and then using a distance measure to compare two speech segments, we have an end-to-end approach and we achieve same/different classification through a network. After training, speaker change times are determined by evaluating the network around segment boundaries and outputting the beginning time of the right segment.

The Siamese network consists of three BLSTM layers followed by two fully connected tanh layers. Transition from the BLSTM to the fully connected layers is achieved by concatenating the forward and backward average activations over time from the last BLSTM layer. Thus we can map variable length sequences into fixed dimensional embeddings. These embedding layers are pretrained either for gender classification or using contrastive divergence loss or triplet loss. After pretraining, we either freeze the Siamese layers and learn a same/different classifier on the embeddings or update the parameters of the Siamese network along with the classifier to achieve better classification accuracy.

2.1. Pretraining with Gender Classification

In the first approach, we train a gender classification network using binary cross-entropy objective. This pretraining can be considered as a simplified version of using multi-class speaker classification to generate the embeddings. To get the gender probability $y^{(m)}$ for the m-th sample, we pass the embeddings through an affine layer followed by sigmoid nonlinearity. If the gender label is represented as with binary labels g, we can write the binary cross-entropy objective $\mathcal{L}_x$ as

$$\mathcal{L}_x = \sum_{m=1}^{M} g^{(m)} \log(y^{(m)}) + (1-g^{(m)}) \log(1-y^{(m)}) \quad (1)$$

2.2. Pretraining with Contrastive Loss

In the second approach, we use a pair of inputs and use contrastive loss as our objective. In this setup, we try to minimize the distance between the embeddings of two inputs if they are uttered by the same speaker and we try to maximize the distance otherwise. Let s_l and s_r denote the speakers of the left and right segments, respectively. Also let δ to be the indicator function and Δ_{cd} be a margin parameter. For a set of M training segments, the contrastive divergence loss $\mathcal{L}_{cd}$ that we want to minimize is

$$\begin{aligned}\mathcal{L}_{cd} = &\sum_{m=1}^{M} \delta[s_l^{(m)} = s_r^{(m)}] d(x_l^{(m)}, x_r^{(m)}) \\ &+ \delta[s_l^{(m)} \neq s_r^{(m)}] \max(0, \Delta_{cd} - d(x_l^{(m)}, x_r^{(m)})) \quad (2)\end{aligned}$$

where $d(.,.)$ is a distance measure such as Euclidean or cosine distance and x_l and x_r are the embeddings obtained from the network.

2.3. Pretraining with Triplet Loss

In the third approach, we use triplets and we try to minimize the triplet loss. In this setup, we have an anchor segment, a positive segment uttered by the same speaker as the anchor and a negative segment uttered by a different speaker. The aim is to find embeddings such that for a given triplet, anchor and the negative sample are separated more than the anchor and positive pair with a margin Δ_{tri}. If we denote the embeddings for the anchor, positive and negative samples using x_a, x_p and x_n, respectively, then the triplet loss $\mathcal{L}_{tri}$ that we want to minimize is written as

$$\mathcal{L}_{tri} = \sum_{m=1}^{M} \max(0, \Delta_{tri} + d(x_a^{(m)}, x_p^{(m)}) - d(x_a^{(m)}, x_n^{(m)})) \quad (3)$$

As in (2), $d(.,.)$ denotes the distance measure.

Once the Siamese layers are pretrained using one of the above methods, we either freeze and use them as feature extractor to generate the input features for the binary same/different classifier or we refine the Siamese layers while training the binary classifier.

3. EXPERIMENTS

3.1. Dataset

For our speaker change detection experiments we use 144 hours of audio from the Hub4 acoustic training data set collected between May 1996 and January 199 [18, 19]. This data set (BN-144) which covers 230 TV shows is manually transcibed and annotated with speaker information.

In the experiments, we used fixed set of pairs or triplets of inputs and did not resample the lists. All training segments are 2s and they are extracted from long enough seech segments after silence removal around manually labeled change points. To get our pairs, from each speaker we sample n_s segments, for each segment, we sample n_p segments among all segments of that speaker, then sample n_p speakers from the remaining speaker set and sample one segment from each of them to get different pairs. The dataset has more segments spoken by male speakers therefore the resulting pairs mainly have male-male comparisons. To achieve a balance, we subsampled the initial list of pairs to get equal number of same gender and different gender comparisons and at the end we have 500000 pairs. For triplets, we followed an approach similar to [9] where we picked n random samples from each of S speakers, for each speaker we generated all possible pairs from those n segments, which gave the anchor-positive pairs, and then for each pair we randomly chose a single negative segment from the remaining $(S-1)n$ segments. This resulted in 329000 triplets.

3.2. Training setups

Our 100 dimensional i-vectors [3] are extracted using the maximum likelihood criteria to train a 2048 component, 40-dimensional diagonal covariance based UBM followed by i-vector extraction matrices. To train these systems, vectors of 9 consecutive PLP features are spliced together and projected down to 40 dimensions using LDA. The i-vector extraction systems are trained on the BN-144 dataset described above.

The embedding network consists of three BLSTM layers followed by two fully connected tanh layers. Transition from the BLSTM to the fully connected layers is achieved by concatenating the forward and backward average activations over time from the last BLSTM layer. In contrastive loss and triplet loss training with Euclidean distance, we also added an L2-normalization layer after the second fully connected layer so that the norms of the embeddings and therefore the distance between the embeddings become bounded. Inputs of the Siamese layers are 19-dimensional PLP features, appended with their deltas and delta-deltas.

Same/different classifier network is a 3-layer fully connected network with rectified linear unit nonlinearity. Its input is the concatenation of either PLPs, i-vectors or the Siamese embeddings of the left and right segments. Since our embeddings are 32 dimensional, the input of the classifier is 64 dimensional. The layers of the classifier have 64 units. Final layer has a single output node with sigmoid nonlinearity.

3.3. Test setups

Ten audio files are taken as test data which are not used in training. However, 117 out of 174 test speakers also have data in training set. Audio pairs used in testing are determined based on speech segment boundaries. These boundaries are determined from either the segments obtained from an ASR system or the segments based on the ground truth speaker labeling with inter-speaker silences removed. Although we trained our networks using 2s clips determined from long enough ground truth segments with silences removed, during test time we do not always have segments that are at least 2s. Therefore, as shown in Figure 2, for each segment type we performed two types of tests. In the first case, we still used fixed length (2s) data. For each boundary, we took 2s of data before and after the boundary, neglecting the cases where we have multiple speakers within that 2s. In the second case, we took variable length clips depending on the lengths of the segments. If they are shorter than 2s, we took the whole segment, otherwise we took 2s of data from the beginning (end) from the segment after (before) the boundary. We did not use longer segments to avoid latency problems. Note that especially in the variable length segment case, there is a significant amount of mismatch between training and testing because we do not have 2s clips as input and also the left and right clip possibly have different durations. For the variable length case, average segment duration for the ASR based segments are 1.49s whereas for the ground truth based ones it is 1.54s.

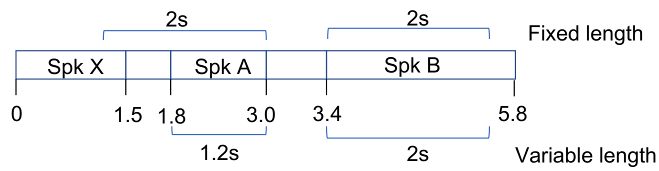


Fig. 2. Extraction of test segments around boundaries for fixed and variable length cases

3.4. Results

Table 1 shows the same/different classification accuracy on the validation data. First two rows show the performance of the classifier (C) when its input is concatenated PLP features or i-vectors of the two segments. Rest of the table summarizes the performance of Siamese layers followed by a classifier (S+C). In these rows, we use pre-training with different objectives, which are binary cross entropy for gender classification, contrastive or triplet loss. The distance measure used in the objective is cosine or Euclidean, and the embedding layers are frozen or updated during classifier training. As shown in the table, fine-tuning the embedding layers during classifier training improves the accuracy in all cases. If we freeze the Siamese embedding layers after pre-training, that is use the Siamese network as a fixed feature extractor, we see that using the triplet loss as the objective leads to better accuracy. This implies that they lead to more dicriminative features as compared to contrastive loss or gender classification. If we check the mean and variance of the Eucliden distances of embeddings of same and different pairs, we also observed that after training with the triplet loss, the average distance between different pairs get much higher than that of the similar pairs and the variances of the distances decrease for both same and different pairs.

Although the single sigmoid layer at the end of the classifier implies a default threshold of 0.5 for change decision, we can thereshold the output to make change decisions when the network output has a higher value, that is more confident about a change. By varying this threshold, we obtained the precision-recall curves for the test data which are shown in Figures 3.(a)-(d). They show the results obtained for ASR boundary based variable length and 2s segments, and also for ground truth boundary based variable length and 2s segments. In each figure, we include the curves for i-vector classification (i-vector) and Siamese network setups pre-trained with triplet loss (Tri) depending on the distance measure, which is cosine (Cos) or Euclidean (Eucl) distance, and depending on whether we freeze (F) or jointly train (T) the embedding layers.

In all four figures, Euclidean distance based triplet loss pretrain-

Table 1. Validation data same/different classification accuracy (%) depending on the pre-training method, distance measure used in the objective and whether embedding layers are frozen

Feat.	Net.	Pre-training	Distance	Freeze Embed.	Accu.
PLP	C	-	-	-	52.2
i-vector	C	-	-	-	86.6
PLP	S+C	Gender	-	Yes	76.9
PLP	S+C	Gender	-	No	78.1
PLP	S+C	Cont. loss	Cosine	Yes	76.7
PLP	S+C	Cont. loss	Cosine	No	87.3
PLP	S+C	Cont. loss	Euclidean	Yes	77.4
PLP	S+C	Cont. loss	Euclidean	No	87.5
PLP	S+C	Triplet loss	Cosine	Yes	84.6
PLP	S+C	Triplet loss	Cosine	No	87.9
PLP	S+C	Triplet loss	Euclidean	Yes	82.7
PLP	S+C	Triplet loss	Euclidean	No	89.0

Table 2. F-measures on the test data for threshold 0.5 depending on the model and the segments used for testing

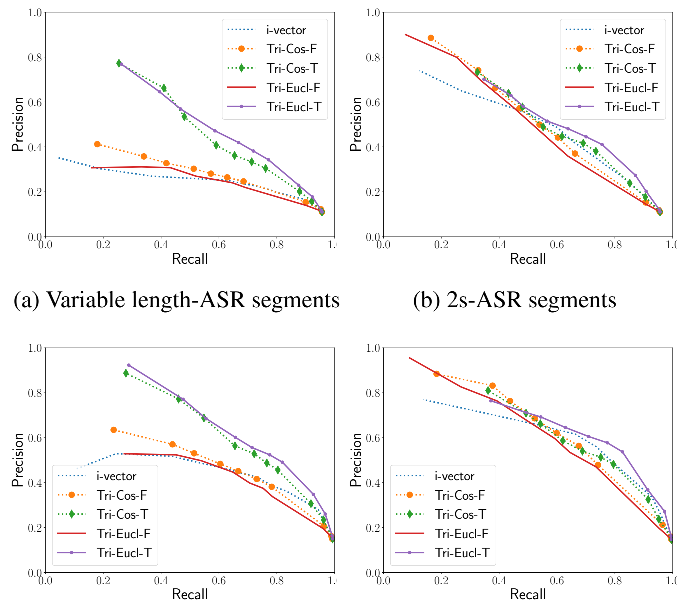
	ASR boundary		Ground truth boundary	
	Variable	2-second	Variable	2-second
i-vector	0.3150	0.4902	0.5036	0.6109
Tri-Cos-F	0.3626	0.4752	0.5130	0.5820
Tri-Cos-T	0.4354	0.5014	0.5821	0.5999
Tri-Eucl-F	0.3332	0.4591	0.4722	0.5736
Tri-Eucl-T	**0.4746**	**0.5323**	**0.6141**	**0.6511**

ing followed by fine tuning during classifier training achieves the best performance. Joint training of the Siamese layers with the classifier achieves better performance than freezing the embeddings. For the tests with variable length segments (Figures 3a and 3c), frozen embeddings perform as good as i-vectors, jointly trained embeddings perform better than the frozen ones. For 2s segments (Figures 3b and 3d), Siamese networks perform better than the i-vector in all models.

For the default threshold 0.5 imposed by the sigmoid output, we report the F-measures for four types of segments used in testing depending on the classifier model in Table 2. In each column, best F-measure is highlighted. As seen from the table, joint training of embedding layers with the classifier after Euclidean distance based triplet loss pre-training achieves the best performance for all segment types. For threshold 0.5, performance of the frozen embedding layers is slightly worse than i-vectors in tests with 2s segments. However, the reduction in performance is less severe for Siamese network based architectures as compared to the i-vector based setup when the test segments have variable duration rather than 2s. This results from the fact that i-vector estimation becomes harder for shorter segments whereas the BLSTM based Siamese embeddings are more robust against changes in input duration.

Although Table 2 reports the results for threshold 0.5 which does not require any threshold tuning, we observed that having a threshold around 0.7 achieves a better F-measure in all cases. But the conclusions from Table 2 are still applicable for this threshold.

We also combined the i-vector and embedding based systems by averaging their sigmoid outputs. In the variable length case, low performance of i-vector deteriorates the F-measure in the combined



(a) Variable length-ASR segments (b) 2s-ASR segments

(c) Variable length-ground truth (d) 2s-ground truth

Fig. 3. Precision-recall curves for i-vector and Siamese network based change detection depending on the type of the test segments.

system as compared to the Siamese network. However, in the 2s conditions, where i-vectors perform reasonably well, combined system achieves better F-measure than the individual systems showing the complementary nature of the two systems. Combined i-vector and Tri-Eucl-T system has an F-measure of 0.5599 and 0.6783 for 2s ASR based and ground truth based test segments, respectively. These are around 5% relatively higher than Tri-Eucl-T on 2s segments.

4. CONCLUSIONS

In this work, we presented an end-to-end speaker change detection setup that consists of Siamese layers for speaker embedding generation and a classifier that makes same/different decisions. We investigated gender classification, contrastive loss and triplet loss based pre-training of the embedding layers. Since our aim is to operate in the low-latency regime, we trained our networks using 2s speech segments extracted around ground truth speaker boundaries and during test time we evaluated them on 2s or shorter segments. Also to speed up test time, we used ASR based segment boundaries around which we got our test pairs. We compared neural network embeddings with i-vectors on the Broadcast News dataset. Experimental results showed that pre-training using triplet loss with Euclidean distance followed by joint training of the classifier achieved higher F-measure in both matched and mismatched (duration and boundary) cases than i-vectors. For low-latency latency applications, test comparisons should be made around ASR based segment boundaries. These segments tend to be shorter than 2s and therefore causes a mismatch in duration. In experiments, we showed that in this condition, neural network embeddings still perform better than i-vectors and performance degradation is less as compared to i-vectors. Therefore, lower dimensional network embeddings not only provide a more compact representation but they are also more robust to the mismatch between training and testing conditions as compared to i-vectors.

5. REFERENCES

[1] Scott Chen, Ponani Gopalakrishnan, et al., "Speaker, environment and channel change detection and clustering via the bayesian information criterion," in *Proc. DARPA broadcast news transcription and understanding workshop*. Virginia, USA, 1998, vol. 8, pp. 127–132.

[2] Daben Liu and Francis Kubala, "Fast speaker change detection for broadcast news transcription and indexing," in *Sixth European Conference on Speech Communication and Technology*, 1999.

[3] Najim Dehak, Patrick J Kenny, Réda Dehak, Pierre Dumouchel, and Pierre Ouellet, "Front-end factor analysis for speaker verification," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 19, no. 4, pp. 788–798, 2011.

[4] Achintya Kumar Sarkar, Driss Matrouf, Pierre Michel Bousquet, and Jean-François Bonastre, "Study of the effect of i-vector modeling on short and mismatch utterance duration for speaker verification," in *Proc. Interspeech*, 2012, pp. 2662–2665.

[5] Ahilan Kanagasundaram, Robbie Vogt, David B Dean, Sridha Sridharan, and Michael W Mason, "I-vector based speaker recognition on short utterances," in *Proceedings of the 12th Annual Conference of the International Speech Communication Association*. International Speech Communication Association (ISCA), 2011, pp. 2341–2344.

[6] Claude Barras, Xuan Zhu, Sylvain Meignier, and J-L Gauvain, "Multistage speaker diarization of broadcast news," *IEEE Transactions on Audio, Speech, and Language Processing*, vol. 14, no. 5, pp. 1505–1512, 2006.

[7] Dimitrios Dimitriadis and Petr Fousek, "Developing on-line speaker diarization system," in *Proc. Interspeech*, 2017, pp. 2739–2743.

[8] Dan Pelleg, Andrew W Moore, et al., "X-means: Extending k-means with efficient estimation of the number of clusters.," in *ICML*, 2000, vol. 1, pp. 727–734.

[9] Hervé Bredin, "Tristounet: triplet loss for speaker turn embedding," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5430–5434.

[10] Arindam Jati and Panayiotis Georgiou, "An unsupervised neural prediction framework for learning speaker embeddings using recurrent neural networks," *Proc. Interspeech*, pp. 1131–1135, 2018.

[11] David Snyder, Daniel Garcia-Romero, Daniel Povey, and Sanjeev Khudanpur, "Deep neural network embeddings for text-independent speaker verification," in *Proc. Interspeech*, 2017, pp. 999–1003.

[12] Renyu Wang, Mingliang Gu, Lantian Li, Mingxing Xu, and Thoms Fang Zheng, "Speaker segmentation using deep speaker vectors for fast speaker change scenarios," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 5420–5424.

[13] Neil Zeghidour, Gabriel Synnaeve, Nicolas Usunier, and Emmanuel Dupoux, "Joint learning of speaker and phonetic similarities with siamese networks.," in *Proc. Interspeech*, 2016, pp. 1295–1299.

[14] Sepp Hochreiter and Jürgen Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.

[15] Sree Harsha Yella, Andreas Stolcke, and Malcolm Slaney, "Artificial neural network features for speaker diarization," in *Spoken Language Technology Workshop (SLT)*. IEEE, 2014, pp. 402–406.

[16] Ruiqing Yin, Hervé Bredin, and Claude Barras, "Speaker change detection in broadcast tv using bidirectional long short-term memory networks," in *Proc. Interspeech*. ISCA, 2017.

[17] Marek Hrúz and Zbyněk Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2017, pp. 4945–4949.

[18] "1996 english broadcast news speech (HUB4)," `https://catalog.ldc.upenn.edu/LDC97S44`.

[19] "1997 english broadcast news speech (HUB4)," `https://catalog.ldc.upenn.edu/LDC98S71`.