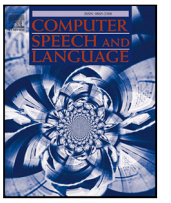
Computer Speech and Language 78 (2023) 101441

Contents lists available at ScienceDirect

Computer Speech & Language

journal homepage: www.elsevier.com/locate/csl



Language modelling for speaker diarization in telephonic interviews

Miquel India *, Javier Hernando, José A.R. Fonollosa

Universitat Politecnica de Catalunya, C/ Jordi Girona 1-3, Barcelona, 08034, Spain

ARTICLE INFO

Keywords:
Speaker diarization
Language modelling
Acoustic modelling
LSTM neural networks

ABSTRACT

The aim of this paper is to investigate the benefit of combining both language and acoustic modelling for speaker diarization. Although conventional systems only use acoustic features, in some scenarios linguistic data contain high discriminative speaker information, even more reliable than the acoustic ones. In this study we analyze how an appropriate fusion of both kind of features is able to obtain good results in these cases. The proposed system is based on an iterative algorithm where a LSTM network is used as a speaker classifier. The network is fed with character-level word embeddings and a GMM based acoustic score created with the output labels from previous iterations. The presented algorithm has been evaluated in a Call-Center database, which is composed of telephone interview audios. The combination of acoustic features and linguistic content shows a 84.29% improvement in terms of a word-level DER as compared to a HMM/VB baseline system. The results of this study confirms that linguistic content can be efficiently used for some speaker recognition tasks.

1. Introduction

Speaker diarization addresses the problem of "who spoke when" in a multi-party conversation. Without prior knowledge of any speaker nor the number of the speakers in the speech, diarization aims to identify all the speech coming from each speaker. Two different sub-tasks can be distinguished in speaker diarization: speaker segmentation and speaker clustering. Speaker segmentation searches for the speaker turn boundaries, whereas speaker clustering aims to group all the speaker turns that correspond to each speaker. Depending on the speech domain, speaker clustering needs to determine the number of speakers in the audio. This work is focused on the telephonic domain, where it is assumed that only two speakers are talking.

The most common approaches used in speaker diarization are based on the Agglomerative Hierarchical Clustering (AHC) strategy. In this strategy, the system is initialized with a speech segmentation where each segment is assumed to correspond to one speaker. This initial segmentation can be created with different approaches like (Gupta, 2015; Bredin, 2017; Yin et al., 2017; Jati and Georgiou, 2017; Tranter and Reynolds, 2004, 2006) or directly splitting the signal into homogeneous segments. AHC consists on grouping iteratively these segments until each segment is assigned to its respective speaker. Therefore, in each iteration a pair of clusters is merged and a new segmentation is created in order to refine the speaker turn boundaries. The Bayesian Information Criterion (BIC) (Tranter and Reynolds, 2004, 2006) was the conventional approach to decide which pair of clusters must be merged. Otherwise, Viterbi Decoding was the most used algorithm for the speaker re-segmentation. Speaker clusters were usually modelled either with Gaussian Mixture Models (GMM) or with i-vectors. I-vector framework (Dehak et al., 2011) combined with Probabilistic Linear Discriminant Analysis (Kenny, 2010; Prince et al., 2012) have shown a noticeable improvement in comparison with GMM approaches. This improvement has been shown for speaker clustering (Desplanques et al., 2015; Sell and Garcia-Romero, 2014; Dupuy et al., 2012; Shum et al., 2011; Woubie et al., 2016) but still not for the segmentation task.

* Corresponding author.
E-mail address: miquel.angel.india@upc.edu (M. India).

https://doi.org/10.1016/j.csl.2022.101441
Received 31 December 2021; Received in revised form 13 August 2022; Accepted 15 August 2022
Available online 18 August 2022

Deep learning has also been successfully applied for speaker diarization (Zajıc et al., 2017; Le Lan et al., 2017; Dimitriadis and Fousek, 2017; Wang et al., 2018; Sun et al., 2019; Huang et al., 2020; Wang et al., 2020; Flemotomos et al., 2020), with different approaches in both clustering and segmentation tasks. Long Short-Term Memory Networks (LSTM) have been efficiently used to detect speaker turns boundaries either using acoustic features like in Bredin (2017), Yin et al. (2017), Wisniewksi et al. (2017), Yin et al. (2018), Lin et al. (2019), or combining acoustic and linguistic content (India Massana et al., 2017; Park and Georgiou, 2018; Park et al., 2019; El Shafey et al., 2019). On the other hand, the success of speaker embeddings for speaker verification has led to use this approaches for clustering. This representation (Snyder et al., 2016; Garcia-Romero et al., 2017; Diez et al., 2019; Wan et al., 2018; Wang et al., 2018) has been explored for the clustering task, outperforming i-vectors when a lot of speech data is available.

Natural language processing is one of the research fields where deep learning have caused a bigger impact. Neural networks have led to big improvements on analyzing and understanding natural language data. The most recent methods to extract features in tasks like machine translation, data mining or natural language modelling are based on word embedding approaches. Word embeddings are numerical word representations trained to capture the contextual information of a language (Mikolov et al., 2013b,a). Several models are known to produce these vectors, from the word2vec work presented in Goldberg and Levy (2014) to character-level models such in Kim et al. (2016). Word embeddings have shown its best performance in both language modelling and machine translation tasks when they are used as inputs of Recurrent Neural Networks (RNN) (Mikolov et al., 2010) or Transformers (Vaswani et al., 2017). Works like (India Massana et al., 2017; Kim et al., 2016; Costa-jussà and Fonollosa, 2016; Sundermeyer et al., 2012; Peters et al., 2018; Howard and Ruder, 2018; Serban and Pineau, 2015), exhibit the good performance of these embeddings with RNN architectures. Transformer based approaches like (Devlin et al., 2019; Radford et al., 2019; Dai et al., 2019; Yang et al., 2019; Brown et al., 2020) have also shown state of the art results in NLP using these words representations.

In this paper we propose an alternative architecture for speaker diarization in telephonic interviews, where our main contribution is a straight-forward algorithm that combines acoustic and linguistic information. The proposed approach is considered to be used for telephonic conversations, therefore the number of speakers per audio is known in advance. Additionally, our approach classifies each of the speaker clusters with an interviewer or customer label. Although there is a lot of tasks where linguistic content and speech have been successfully combined, the joint use of these sources has still not been fully explored for speaker diarization. Moreover, in several real-life applications it is needed to implement both Automatic Speech Recognition (ASR) and diarization (Canseco-Rodriguez et al., 2004; Canseco et al., 2005), which increases the motivation on combining both systems. Call-Centers have a wide set of tasks with different scenarios where is needed to perform call-transcription. This paper will be focused on the telephonic interview scenario which is a very important case for some Call-Centers. In this scenario, speaker patterns can be extracted from linguistic content in an easier way than in other cases, because part of the speech of some speakers may be prior known. In fact, the interviewer questions are commonly known and customers speech is sometimes limited to specific sets of expressions or answers such as giving a score, say yes or no, and so on. Given this motivation, this work aims to research how to combine efficiently acoustic and linguistic data for speaker diarization in this scenario. Therefore, in this work we present a LSTM based system where acoustic features are fused with linguistic content to identify the speech coming from different speakers. LSTM networks are commonly used in language modelling tasks to predict a word given the sequence of the previous words. In this work, we will use LSTMs similarly in order to infer about the speaker who says the word. With the possibility of adding acoustic features in the network, we examine its behaviour in a scenario where linguistic content contains discriminative speaker information. This scenario is based on a real application situation, more specifically in the Call-Center context. Call-Center dialogues are composed by an operator–customer conversation where some part of the operator speech may be known a priori and the number of speakers is always known. In this work, our approach is evaluated on a database composed by telephone conversations where some interviewers make a survey to different customers. Given prior knowledge of the set of questions in each survey and the number of speakers in the conversation, the objective of this task is to identify the speech of the interviewer and the client interviewed for each recording.

The rest of this paper is structured as follows. Section 2 illustrates the architecture of the system. Section 3 gives the details of the system setup. Experimental results are presented in Section 4. The concluding remarks and some future work are given in Section 5.

2. Architecture description

The proposed algorithm is designed to work in a scenario where linguistic data contain speaker patterns. In this context, each recording is a two-speaker conversation where a first speaker (Interviewer) interviews a second speaker (Customer). These interviews are based on a survey composed by a set of questions which are similar for all the recordings. Therefore, the aim of the task is to find when the Interviewer and the Customer are speaking in each recording. The presented system uses both acoustic and linguistic content as inputs, hence the speech signal is initially pre-processed with an acoustic feature extractor and an ASR system. The output of the ASR and the acoustic descriptors are then used as inputs of the system. Given these inputs, the system will be trained to tag each word with its respective speaker label (Interviewer, Customer).

The architecture of the proposed system is based on the iterative algorithm shown in Fig. 1. Two different networks are used, each of them fed with different inputs. The system is initialized trough *Neural Network 1*, which only uses linguistic content to create the first speaker labels for the iterative algorithm. On the other hand, *Neural Network 2* works iteratively with both acoustic and linguistic data as inputs. Both networks work with sequences of word level representations and output the speaker labels corresponding to those input sequences of words. In each iteration the output speaker labels from the previous iterations are used to create two speaker models (Interviewer, Customer), which are used to extract an acoustic speaker score from each word. These scores indicate whether that word corresponds to the Interviewer or to the Customer. Hence, at each iteration of the algorithm, the speaker labels of

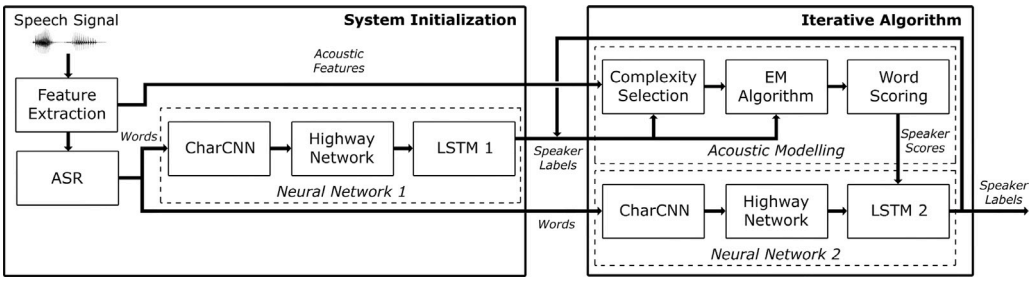


Fig. 1. System diagram.

the previous iteration are used to create the acoustic speaker scores which will be input additionally to the words in *Neural Network 2*. The algorithm is iteratively run for a few iterations.

These neural network architectures are based on the system presented in Kim et al. (2016). In our work, instead of using LSTMs to predict words, the networks are trained to tag each word with its corresponding speaker. The proposed algorithm proceeds by the following steps:

1. The system is initialized extracting both acoustic features and linguistic content. The words extracted from the ASR are introduced in *Neural Network 1*. These words are mapped into word embeddings (Section 2.1), which are the input to the first LSTM. *LSTM 1* yields an initial set of speaker labels (Section 2.2) which will be used for the acoustic speaker modelling block in the iterative algorithm.
2. Given the speaker labels either from *Neural Network 1* in the first iteration or from *Neural Network 2* in the next iterations, the two speaker acoustic models (GMMs) are created. These models are used to extract an acoustic speaker score for each word. These scores are calculated as the posterior probability of the Customer speaker GMM word given the word acoustic features (Section 2.3).
3. Acoustic speaker scores are used additionally to the words as inputs of *Neural Network 2*. In *Neural Network 2* the words are mapped into word embeddings and the concatenation of each word embedding and its acoustic speaker score is input to *LSTM 2*. The output speaker labels from *Neural Network 2* will be then used again in step 2 in a new iteration. The algorithm finishes after a few iterations and the last iteration output corresponds to the final result.

2.1. Character-level word embedding

The architecture of the proposed system (Fig. 2) contains a LSTM neural network. This recurrent neural network uses as input sequences of word embeddings. Word embeddings are word representations modelled as real value vectors mapped from its textual form. In this system word embeddings are obtained from the output of a character-level convolutional neural network (CharCNN) (Kim et al., 2016).

In any language we can define a dictionary V where each word can be represented as a vector w in $V \in \mathrm{R}^{d' \times |C'|}$. Variables d' and C' correspond to the vector and dictionary size, respectively. On the other hand, any word w can be constructed as a sequence of characters $[c_1, \ldots, c_l]$, where l is the word length. Therefore, if we define a dictionary of characters $Q \in \mathrm{R}^{d \times |C|}$, where each character of a set C is represented as a d size vector, then any word can be constructed as a matrix $\mathrm{C}^w \in \mathrm{R}^{d \times l}$. Character-based word embedding approaches map these 2D word representations into another low dimension space, which is discriminative in terms to the factor aimed to infer. Given a word w, a narrow convolution is applied between its representation C^w and a filter $\mathrm{H} \in \mathrm{R}^{d \times u}$ of width u. Applying a non linear function in the sum of this convolution and a bias term, we obtain a feature map $f^w \in \mathrm{R}^{l-u+1}$. The ith element of f^w is defined as:

$$f^w[i] = \tanh(\langle C^w[*, i : i+u-1], H \rangle + b) \tag{1}$$

where $C^w[*, i : i+u-1]$ is the i-to-$(i+u-1)$-th column of C^w and $\langle A, B \rangle = Tr(AB^T)$ is the Frobenius inner product. We apply a max-over time pooling over the feature map so as to take the most representative feature in the vector:

$$y^w = \max_i f^w[i] \tag{2}$$

where y^w is the feature corresponding to the filter H (when applied to the word w). Thus if we had a set of N filters in the network, for each word w we obtain a N size representation $y = [y^{w1}, \ldots, y^{wN}]$, where each component is the output feature of a filter. For many NLP tasks the number of filters N is used to be chosen between [100,1000].

Additionally to the CharCNN, one more network is implemented replacing y_t with x_t at each step in the LSTM architecture. Instead of using a typical set of fully-connected layers, those are replaced by a Highway network (Srivastava et al., 2015a,b). Highway networks are gate-based layers inspired by LSTMs, which have shown state of the art results in language modelling tasks (Kim et al.,

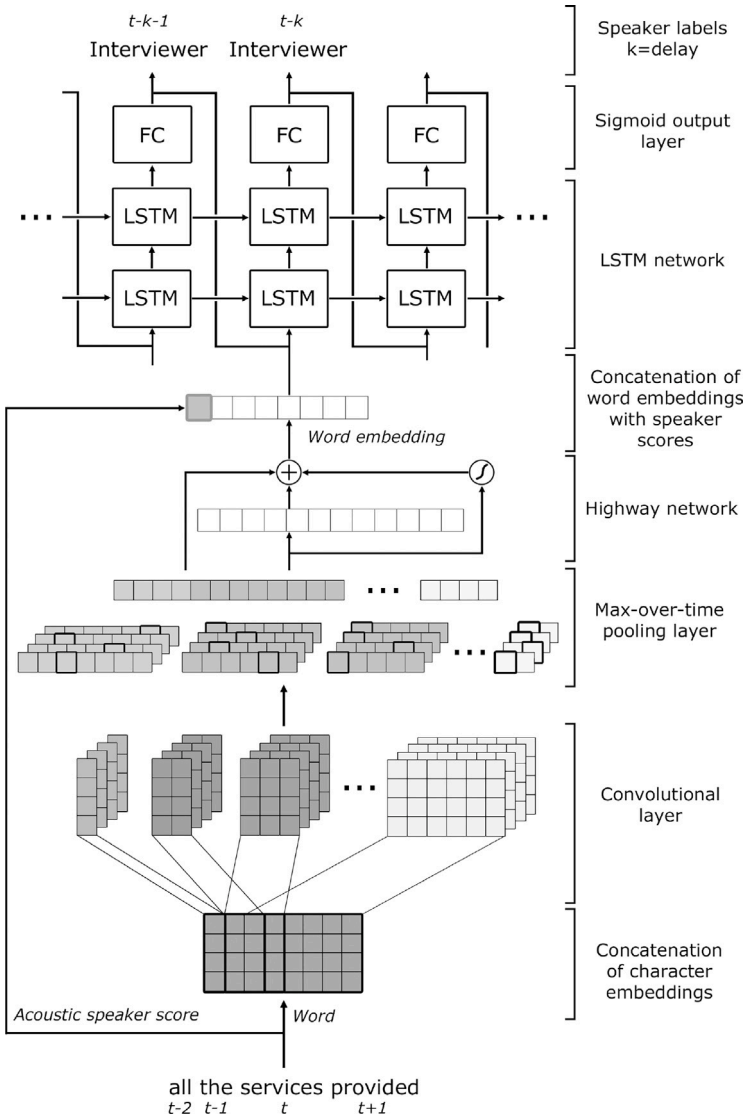


Fig. 2. Network Architecture Scheme. In Neural Network 1 the words are the only input. In Neural Network 2 acoustic speaker scores are input additionally in concatenation with word embeddings in the LSTM.

2016; Jozefowicz et al., 2016; Zilly et al., 2017). Therefore, instead of using a feed-forward layer, x_t is computed with the following function:

$$x_t = T \odot g(W y_t + b) + (1 - T) \odot y_t \tag{3}$$

$$T = \sigma(W_T y_t + b_T) \tag{4}$$

where g is a non-linear function, $\odot$ is the element-wise multiplication, T is the *transform gate* and $1 - T$ is the *carry gate*. These layer gates allow to control whether each component of x_t is obtained by a feed forward layer $g(W y_t + b)$ or it is directly carried from the input y_t. As is shown in Kim et al. (2016), these networks show better performance by modelling the interactions between the character n-grams extracted by the filters over y_t. Highway networks architecture was addressed to solve the learning issues found in large and deep networks. However, these networks are implemented in this system as an alternative of deep feed-forward networks with the aim of optimizing the data flow across the layers.

2.2. LSTM word classifier

LSTM networks are used in this work in order to assign for each word its corresponding speaker. As is shown in Fig. 2, the network assigns the speaker label to the corresponding introduced word, given its word representation x_t. *Neural Network 1* LSTM

uses only as input word embeddings, meanwhile *Neural Network 2* LSTM is fed with the concatenation of word embeddings and their respective acoustic scores. In our approach we use a two hidden layer LSTM network. Hence, the hidden state h_t of the second LSTM layer is then used as input of a last dense layer, whose output corresponds to the speaker label l_t. In this case Customer and Interviewer label l_t corresponds to outputs '1' and '0', respectively. Compared to the system presented in Kim et al. (2016), we have implemented two extensions in the LSTM so as to adapt the network for this task:

1. Scheduled Sampling: In order to improve the model accuracy and the training stability, we have applied scheduled sampling (Bengio et al., 2015). This method consists on using the previous output $\hat{l}_{t-1}$ as an additional input to x_t in the LSTM during the training. Hence in each training step, the LSTM input (Fig. 2) will be the concatenation of x_t, h_{t-1} and $\hat{l}_{t-1}$. Feeding the network with the groundtruth label leads to a faster convergence and a better model performance. In testing phase, $\hat{l}_{t-1}$ corresponds to the previous word speaker label. Therefore at time t and considering a sequence of the past speaker labels $\left[l_1, \ldots, l_{t-1}\right]$ we extract both t word Customer and Interviewer posterior probabilities. The inference is then posed as a decoding problem where we want to find the most likely sequence of speaker labels given the input sequence of words. We use the Beam Search algorithm to solve the speaker word decoding. This approach is a Viterbi decoding variation which prunes the K most likely hypothesis in each decoding step instead of considering all the paths.
2. Output delay: Given a sequence of word embeddings $x = [x_1, \ldots, x_T]$, the inferred speaker label depends on the previous steps of the sequence but not on the next ones. Therefore, the network is trained with an output delay so the model decision in step t depends not only on the past but also on k future steps. In order to obtain the delayed desired label l_t, during training the network is then fed with the word embedding x_{t+k}, the hidden state h_{t-1} and the desired output from the previous step $\hat{l}_{t-1}$.

2.3. Acoustic modelling

Given the speaker labels either obtained from *Neural Network 1* or *Neural Network 2*, two acoustic speaker models are created (Interviewer, Customer) in each iteration. The MFCC features extracted in the system initialization are used to train a Gaussian Mixture Model (GMM) per speaker. We use the speaker labels to group all the features corresponding to the words of each speaker. These clusters are then used to train the GMMs using the Expectation–Maximization (EM) algorithm. The complexity selection of each speaker model is defined by means of the following expression:

$$GM_j = \text{round}\left(\frac{R_j}{CCR}\right) \tag{5}$$

where the number of Gaussian mixtures GM_j to model speaker j is determined by the number of frames belonging to that cluster R_j divided by the Cluster Complexity Ratio (CCR). CCR (Anguera et al., 2006) is a constant value fixed across all the recordings that defines the number of frames needed per mixture in a GMM.

The two GMMs are then used to evaluate the set of words given each speaker model. For each word we extract a speaker score in order to refine the word labelling in each iteration. The score of each word is computed by the posterior probability of the Customer model given the features of this word. Hence, let define a word w composed by a set of features $\left[o_1, \ldots, o_M\right]$, where M is the number of frames in the word. The acoustic score is then computed as:

$$P(\text{Cus}|w) = \frac{P(w|\text{Cus})P(\text{Cus})}{P(w|\text{Cus})P(\text{Cus}) + P(w|\text{Int})P(\text{Int})} \tag{6}$$

where Cus and Int refer to Customer and the Interviewer models and $P(\text{Cus})$ and $P(\text{Int})$ refer to their respective priors. Each speaker model j is defined with a Ω_j GMM, composed by GM_j Gaussian mixtures. The acoustic score of a word w respect to the speaker j modelled with Ω_j is defined as:

$$P(w|SPK_j) = \sum_i \log P(o_i|\Omega_j) \tag{7}$$

$$P(o_i|\Omega_j) = \sum_k w_{jk} P(o_i|\Omega_{jk}) \tag{8}$$

where $P(o_i|\Omega_j)$ corresponds to the o_i (ith frame assigned to w) likelihood given Ω_j GMM, $P(o_i|\Omega_{jk})$ is the likelihood of o_i given the kth mixture of Ω_j and w_{jk} is the corresponding mixture weight. The posterior probability $P(Cus|w)$ of each word will be used as the acoustic speaker score input to *Neural Network 2* LSTM.

3. Experimental setup

The proposed system will be evaluated in a real Call-Center database against a conventional speaker diarization system. The details of the scoring metrics for this task, the database and baseline used and the system setup are given in this section.

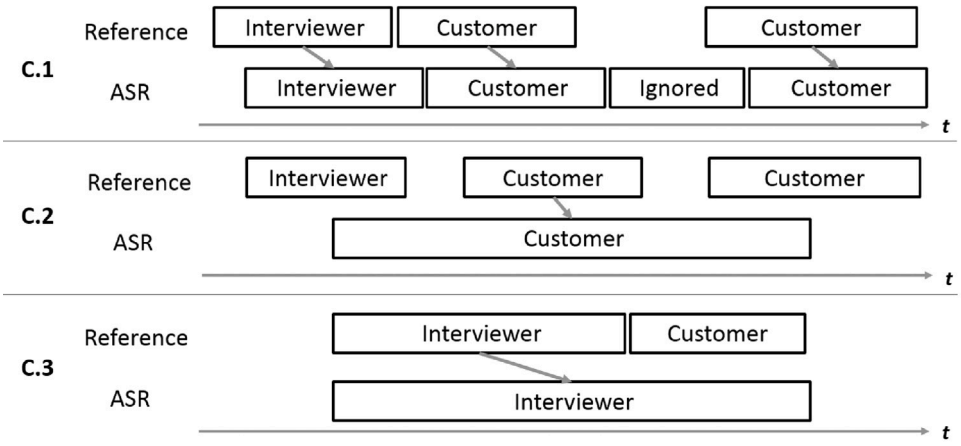


Fig. 3. ASR Groundtruth Labelling: Boxes represent word segments with its respective speaker label. Arrows indicate the label assignation between transcription word labels to the ASR words. C.1, C.2 and C.3 correspond to condition 1, 2 and 3 in the direct overlapping criterion.

3.1. Scoring metrics and criterion

The most common metric used in speaker diarization is the Diarization Error Rate (DER). This metric considers three kind of different errors: Miss Speech (MISS), False Alarm (FA) and Speaker Error Rate (SER). The speech activity detection in this system is directly produced by the ASR system, where word-time stamps can be used as a very accurate speech segmentation. Hence the FA and MISS errors in our system are only produced by the ASR output and not by our diarization approach. For instance, in order to evaluate the performance of the presented approach, the FA and MISS are ignored and only the SER will be considered for the experiments. On the other hand, conventional DER is computed in terms of time duration. However, the algorithm presented works in word terms. Therefore, we have used a DER variation called Word-level Diarization Error Rate (WDER) (Park and Georgiou, 2018). This metric is computed as the percentage of words that are assigned to a wrong speaker to the total number of words.

In order to evaluate the presented approach, it is needed to use a modified reference that uses the same word segmentation as the one produced by the ASR. We have used a direct overlapping criterion so as to assign speaker labels from the manual transcription to the word segmentation created by the ASR. Fig. 3 shows graphically how this criterion is applied according to the following conditions:

1. Given two time overlapped transcription and ASR words, the transcription word label is assigned to the ASR word if their overlap is bigger than half the time duration of the ASR word.
2. If (1) is not fulfilled but the overlap is bigger than half the time duration of the transcription word, the label is also assigned to the ASR word.
3. If there is more than one transcription word overlapped to one ASR word. The label assigned corresponds to the word with the maximum overlap.
4. The ASR words that do not have transcription words time overlapped nor they fulfil the previous conditions are not evaluated.

3.2. Database and scenario analysis

The database used for this work is a set of recordings from a Call-Center. This data has been obtained from a project with a private company, hence is not publicly available. Each of the recordings from this database contains a survey in Spanish of approximately 5 min duration. The survey context distinguishes two speakers: the Interviewer and the interviewed Customer. The questions asked in the survey are common for all the training and test recordings but with different speakers. This database is composed by 270 telephone recordings where we used 240 for training and the other 30 for the test. This test partition is composed by a set of 18,299 words, where 14,498 words correspond to the Interviewer and 3,801 words correspond to the interviewed people (Customer). The word labels are known from a manual annotation with its time stamps, where only word speaker labels are used for training. This manual annotation also contains special tokens which include noisy and overlap labels. These tokens have been removed for both training and test steps.

One of the main problems of this dataset is the unbalanced amount of speech signal between the two speakers. The interviewer speech comprise approximately more than the 79% percent of the recordings. Furthermore, the interviewed Customer participation is reduced to short speech segments due to the content and the structure of the survey. Fig. 4 shows the turn duration distribution from the test partition. A considerable part of the interviewer speech is based on the questions asked on the survey. Hence, their speaker turns are larger (more than 6 words) than the Customer ones, whose speech is mainly based on short answers (1 up to 5 words). The lack of speech signal from the Customer prevents to perform a reliable diarization using only acoustic information.

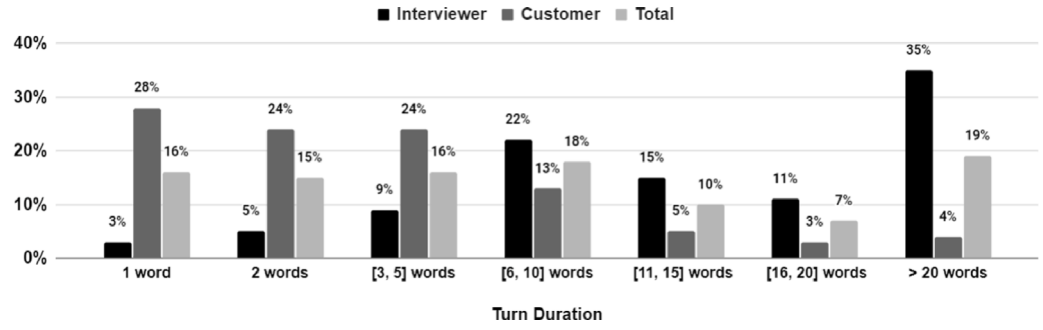


Fig. 4. Turn duration distribution.

In terms of clustering, speaker turns reduced to few words cannot be efficiently modelled with only acoustic features. In terms of speaker segmentation, cluster initialization is not accurately performed if is created splitting the signal into homogeneous segments. The unbalanced amount of speech of the two speakers produces that uniform initial segments are very likely to contain both speakers speech or only from the Interviewer one. Non uniform speaker segmentation approaches based on clustering methods did not also perform well due to the short speaker turns duration.

3.3. Baseline

With the aim of analyzing the impact of linguistic content in the proposed scenario for the speaker diarization task, we have selected a baseline that only models speaker traits from the audio features. The presented approach is compared with the Bayesian Hidden-Markov-Model (HMM) based algorithm proposed in Diez et al. (2018). Given the database scenario and its conditions presented in the Section 3.2, this algorithm has been considered as baseline due to its capacity to robustly estimate speaker models from very short speech segments. This system uses only acoustic features, works in the frame level and follows a Bayesian HMM topology. Each speaker state is represented as a low-dimensional vector y_s, given the following Joint Factor Analysis (JFA) based equation:

$$\mu_s = \mu^{UBM} + V y_s \tag{9}$$

where given a Universal Background Model (UBM) trained as a GMM, the super-vector of concatenated Gaussian component means for a speaker s is posed as the sum of the UBM mean super-vector and the product of an eigenvoice matrix V and its corresponding y_s vector. This eigenvoice matrix V is trained so as to project the speaker variability into a low-dimensional sub-space, although with this procedure the inter-channel variability is also modelled. Both UBM and V matrix training and also y_s extraction are described with more detail in Dehak et al. (2011), Kenny et al. (2007). Given this speaker modelling procedure, the Bayesian HMM topology is defined so as to assign for each speech frame its corresponding speaker state. In this HMM topology there can be multiple states per speaker, which all share the same specific distribution. Therefore a sequence of D states can correspond to the same speaker imposing a minimum speaker turn duration. An iterative Variational Bayes (VB) based procedure is then used to infer about the speaker assigned to each speech frame. This algorithm allows to perform iteratively both segmentation and clustering tasks, where the stopping criterion is also defined through a VB based equation.

In order to evaluate the baseline with the word level scoring metric defined in 3.1, we apply an overlapping criterion to tag each word given the frame labelling output from the baseline. Therefore, the label from the time overlapped frames to a word is directly assigned to that word. If there are overlapped frames from both labels, the label assigned to the word corresponds to the one with more overlapped frames.

3.4. Optimization and setup

The ASR system used in our approach to extract the words from the speech signal is based on Povey et al. (2016). We have implemented the ASR with the Kaldi toolkit (Povey et al., 2011) and its performance in the proposed database is about a 6% Word Error Rate (WER), with a 1.4% of insertions, a 0.4% of deletions and a 4.2% of substitutions. Following the same criterion applied to the manual annotation, special tokens have been removed from the ASR output. The neural networks were trained with the manual transcription words and tested with both transcription and ASR output words. On the other hand, the acoustic modelling was applied extracting MFCCs features. The extraction was done using 10 ms shifted Hamming windows, where each frame contains 20 MFCCs coefficients. Hamming window length was set to 30 ms. Speaker modelling was implemented by means of the EM algorithm, where the CCR ratio in order to define GMM mixtures was set to 7 s per Gaussian.

The two neural networks were trained by truncated back-propagation trough time (Werbos, 1990; Graves, 2013). RMSprop was used with an initial 0.1 learning rate and the back-propagation was done for 35 steps. The learning rate was decayed by a 0.5 factor if validation perplexity did not improve by more than 1.0 after an epoch. Both networks were trained for 14 epochs with 20 size

Table 1
WDER evaluation with different word input sources.

	Oracle (manual transcription)		
	Interviewer	Customer	Total
HMM/VB baseline	3.22	50.04	13.05
NN1	2.99	10.08	4.47
NN2	1.68	4.34	2.23
NN2 (iterative)	**1.67**	**3.5**	**2.05**
	ASR		
	Interviewer	Customer	Total
HMM/VB baseline	3.5	51.32	13.55
NN1	3.51	13.44	5.34
NN2	1.74	5.04	2.35
NN2 (iterative)	**1.61**	**3.62**	**1.98**

batches using binary cross entropy loss. For regularization we used dropout (Hinton et al., 2012) with probability 0.5. The dropout was applied on the LSTM input to hidden layers (except on the initial Highway to the LSTM layer) and the hidden-to-output sigmoid layer. Gradient updating was constrained to normalize gradient to 5. If the L_2 norm was above 5 in the batch, it was normalized again before the updating.

Neural network architectures were setup similar to the large model presented in Kim et al. (2016). The CharCNN was setup with a set of h = 500 filters. These filters had the next range of widths w=[1,2,3,4,5,6], with the following number of filters per width [25,50,75,100,100,200] respectively. Character embeddings had a d=15 size and $tanh$ was the non-linear function applied in the convolutional step. The Highway network was set with only one hidden layer and Rectified Linear Units (ReLU) as activation functions. Both LSTMs were equally setup except for the speaker acoustic score introduced additionally in the *Neural Network 2*. LSTMs were composed by 2 hidden layers, with 150 nodes per layer. Instead of using softmax-layer, the output layer was based on only one sigmoid activation with k=2 delay steps.

The baseline system was setup similar to Diez et al. (2018) but considering the proposed domain and the database size. We used 20 MFCC as features, we trained a 512 mixtures UBM-GMM with diagonal-covariance and the speaker latent variable y_s size was set to 300. The VB inference setup is the same than (Diez et al., 2018) except for D which was tuned to impose a 0.5 s minimum turn duration. Additionally, the system was tuned to directly force the algorithm to finish with two speakers.

4. Results

The proposed approach has been thoroughly evaluated against the mentioned baseline in a Call-Center database. In order to analyze the individual and joint contribution of both acoustic and language modelling, two different outputs of the system have been evaluated. In one hand, the speaker labels from the *Neural Network 1* output (NN1) will be used to analyze the performance of the system using only the linguistic content. On the other hand, the *Neural Network 2* output will be used to evaluate the joint performance of both linguistic and acoustic features. We will consider the first iteration speaker labels (NN2) and the labelling when the system converges (NN2 (Iterative)). Furthermore, the WDER of both speakers (Interviewer, Customer) has also been computed in order to be more accurate in the results analysis.

Two different evaluations are presented in the following subsections so as to analyze the behaviour of the proposed approach. In Section 4.1, the different blocks of the system are evaluated and we analyze the performance of the algorithm when we use either the manual transcription words as inputs or the ones created by the ASR. In Section 4.2, the WDER of the systems will be evaluated for different speaker segment turn durations.

4.1. Global analysis

Table 1 shows the WDER for the different systems with both input conditions: manual transcription (Oracle) and ASR. The baseline shows the worst performance in both conditions with a WDER higher than 10%. On the other hand, the presented system shows a WDER lower than 6% in both conditions for all the tested outputs. NN1 shows a 4.47% and a 5.34% WDER for Oracle and ASR conditions, respectively. Thus the proposed system outperforms the baseline with only using linguistic content as input. NN2 has shown the best performance of all the evaluated systems. With only one iteration, using both acoustic features and linguistic content the system shows a 2.23% WDER for Oracle conditions and a 2.35% for the ASR ones. After a few iterations the best results are achieved with a 2.05% and 1.98% WDER for both Oracle and ASR conditions, respectively. Therefore, the combination of both acoustic and linguistic data provides the best results in the proposed scenario.

The training of both neural networks is done using the words from a manual transcription as inputs. However, in the testing phase we have evaluated the system with both manual transcription and ASR words. This evaluation has been done in order to analyze how the word error introduced by the ASR decreases the system performance. As it is was previously shown in Table 1, the baseline performance is worst than the initial speaker labels produced by NN1 for both conditions. NN1 WDER shows a relative error improvement of 65.74% in comparison with the HMM/VB system in the Oracle condition. In the ASR condition, NN1 also

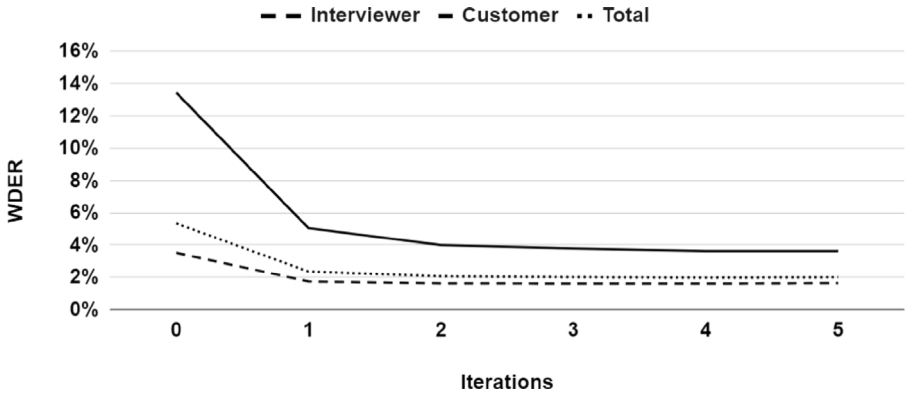


Fig. 5. Iterative algorithm WDER parametrized by the number of iterations run in the system. The results shown correspond to the ASR condition. Iteration 0 corresponds to the initial speaker labels produced by *Neural Network 1*.

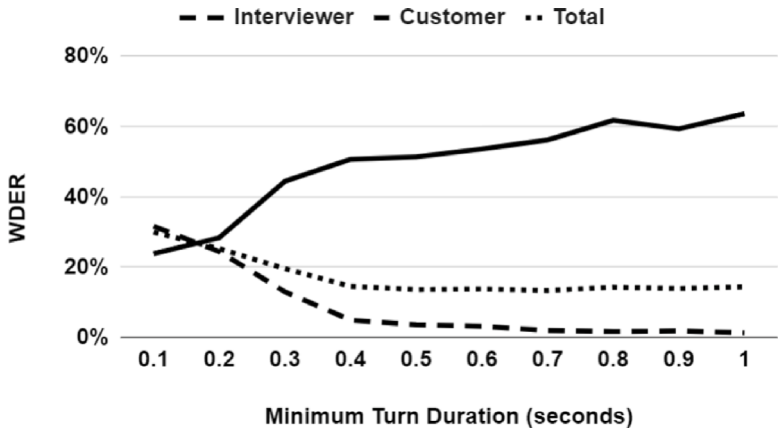


Fig. 6. HMM/VB WDER parametrized by minimum turn duration applied on the model. The results shown correspond to the ASR condition.

outperforms the baseline but with less margin. The relative WDER improvement between NN1 and the HMM/VB baseline is 60.59%. Although there is a system performance decrease caused by the WER from the ASR, NN1 still outperforms the baseline system using only linguistic content. Otherwise, we have also analyzed how the decreased performance produced by the ASR is less significant when we use acoustic data in the system. Iterative NN2 WDER shows a relative error improvement of 82.91% in comparison with the HMM/VB system in the Oracle condition. In the ASR condition, this relative error improvement is similar with a 82.65% in comparison to the HMM/VB system. Therefore, despite the word error introduced by the ASR, the use of acoustic data in the iterative algorithm leads to almost an identical performance when using the manual transcription as input.

The iterative algorithm is initialized with the speaker labels provided by NN1. Hence, the number of iterations needed in the system to converge depends on the labelling produced by the system using only linguistic content. Fig. 5 shows the WDER of the iterative algorithm in relation to the number of iterations run for the ASR condition. In this figure we see that the system converges after 2 or 3 iterations. The speaker labelling created by NN1 corresponds to the iteration 0 with a Interviewer 3.51% WDER and a Customer 13.44% WDER. In the first iteration the WDER is decreased to 1.74% and 5.04% for both Interviewer and Customer, respectively. In the following iterations the system already converges around a 1.61% Interviewer WDER and a 3.62% Customer WDER. These results indicate that the initial speaker labels from NN1 are already very accurate. Therefore, NN2 only needs a few iterations to refine the speaker labelling with the addition of acoustic data.

4.2. Turn duration segment analysis

Conventional speaker diarization systems performance decreases when speaker turns are very short. The proposed baseline is based on a HMM topology that assumes a minimum turn duration in the model so as to avoid over-segmentation. This restriction increases the robustness of the system but also decreases the accuracy on the smaller segments. Fig. 6 shows the WDER of the HMM/VB system in relation to the speaker turn duration condition applied on the model. As it is shown, there is a trade-off between the Interviewer and Customer WDER, which depends on the turn duration parameter. This trade-off is correlated to the average speaker turn duration of each speaker. In Fig. 4 is shown that most of the Interviewer segments have more than 6 words length

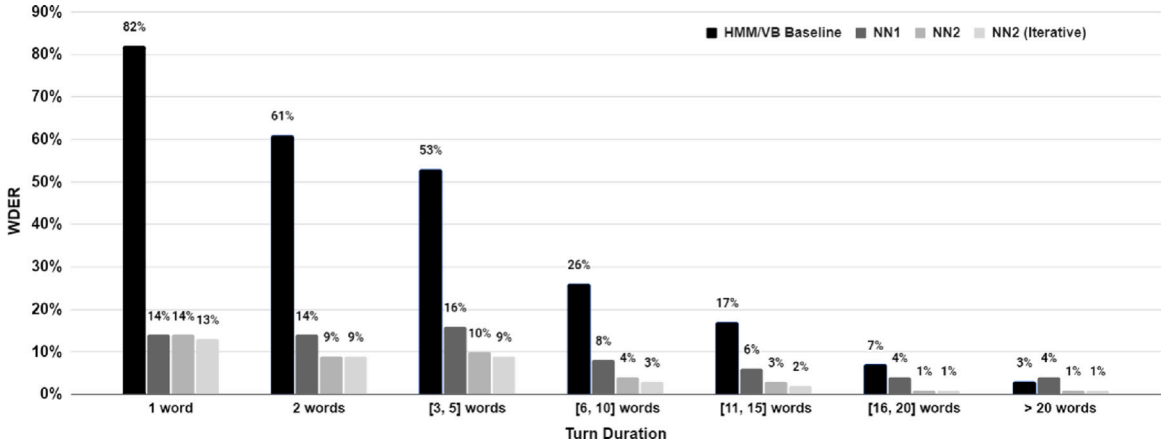


Fig. 7. Total WDER evaluation for different speaker turn duration. These results are extracted for the ASR condition.

and the Customer ones are shorter. Furthermore, there is more speech from the Interviewer than the Customer in the recordings. Therefore, if we decrease the minimum turn duration parameter, the Customer WDER increases but the Interviewer WDER decreases. This trade-off makes very difficult to set-up this kind of systems correctly.

Our proposed system infers directly over each word, hence is not needed to impose any temporary restriction. Table 1 shows that the Customer WDER is still higher than the Interviewer one for all the experiments of the presented approach. However, the relation of both speaker errors is lower compared to the baseline system. In order to analyze the behaviour of the proposed system in different length turns, we have evaluated the WDER for different intervals of speaker segment durations. Fig. 7 shows the total WDER of all the system blocks for several turn lengths. As it was expected, the WDER increases as shorter are the turns for all the systems. The baseline system has more than 50% WDER in turns shorter than 6 words. The proposed system outputs show better results in short turns with a global WDER between 9% and 16%. In turns larger than 6 words, the baseline system performance improves. Despite this improvement, our proposed system still outperforms the baseline for almost all the turn durations. Only the HMM/VB system shows better results in turns larger than 20 words compared to NN1, where only linguistic content is used. The benefit of using either only linguistic content or both linguistic and acoustic data in the system for different segment turns can also be analyzed from Fig. 7. The relative improvement between NN1 and Iterative NN2 for the total WDER is 7.14% for 1 word turns and 35.7% for 2 word turns. If we do the same analysis for long turns, the WDER relative improvement is about 75% for both (Le Lan et al., 2017; Huang et al., 2020) word turns and turns larger than 20 words. Thus acoustic data refines better the labelling produced by NN1 in the larger turns rather than in the shorter ones. Therefore, linguistic content can be efficiently used for tagging very short speaker turns, where acoustic data is less discriminative. On the other hand, the addition of acoustic data shows better results in larger speaker turns, where acoustic features are more effective.

5. Conclusion

In this paper we have investigated the combination of linguistic content and acoustic features for speaker diarization. We tested LSTM neural networks in order to merge acoustic and language modelling. This combination have outperformed the HMM/VB based baseline system where only acoustic data is used. Moreover, we have shown that language modelling is able to work better in situations where acoustic modelling performance is worse, such as in classifying short speech segments. The results indicate that with linguistic content, speaker diarization performance is less sensitive to decrease in short speaker turn conversations. For future work, it seems interesting to explore different acoustic based approaches that could perform efficiently with very short utterances. Additionally and considering that our work has only been tested for telephonic interviews, it would be also interesting to extend our approach to be used in scenarios with less correlation between linguistic content and speaker identities.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

The authors do not have permission to share data.

Acknowledgements

This work was partially supported by the Spanish Project DeepVoice (TEC2015-69266-P) and by the projectPID2019-107579RB-I00/ AEI / 10.13039/501100011033.

References

Anguera, X., Wooters, C., Hernando, J., 2006. Automatic cluster complexity and quantity selection: Towards robust speaker diarization. In: International Workshop on Machine Learning for Multimodal Interaction. Springer, pp. 248–256.

Bengio, S., Vinyals, O., Jaitly, N., Shazeer, N., 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In: Advances in Neural Information Processing Systems. pp. 1171–1179.

Bredin, H., 2017. Tristounet: triplet loss for speaker turn embedding. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 5430–5434.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J.D., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., et al., 2020. Language models are few-shot learners. Adv. Neural Inf. Process. Syst. 33, 1877–1901.

Canseco, L., Lamel, L., Gauvain, J.-L., 2005. A comparative study using manual and automatic transcriptions for diarization. In: IEEE Workshop on Automatic Speech Recognition and Understanding, 2005. IEEE, pp. 415–419.

Canseco-Rodriguez, L., Lamel, L., Gauvain, J.-L., 2004. Speaker diarization from speech transcripts. In: Proc. ICSLP, Vol. 4. pp. 3–7.

Costa-jussà, M.R., Fonollosa, J.A.R., 2016. Character-based neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers). Association for Computational Linguistics, pp. 357–361. http://dx.doi.org/10.18653/v1/P16-2058, URL http://www.aclweb.org/anthology/P16-2058.

Dai, Z., Yang, Z., Yang, Y., Carbonell, J.G., Le, Q., Salakhutdinov, R., 2019. Transformer-XL: Attentive language models beyond a fixed-length context. In: Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics. pp. 2978–2988.

Dehak, N., Kenny, P.J., Dehak, R., Dumouchel, P., Ouellet, P., 2011. Front-end factor analysis for speaker verification. IEEE Trans. Audio Speech Lang. Process. 19 (4), 788–798.

Desplanques, B., Demuynck, K., Martens, J.-P., 2015. Factor analysis for speaker segmentation and improved speaker diarization. In: Proc. Interspeech 2015. pp. 3081–3085.

Devlin, J., Chang, M.-W., Lee, K., Toutanova, K., 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In: Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Vol. 1. Long and Short Papers, pp. 4171–4186.

Diez, M., Burget, L., Matejka, P., 2018. Speaker diarization based on Bayesian HMM with eigenvoice priors. In: Odyssey. pp. 147–154.

Diez, M., Burget, L., Wang, S., Rohdin, J., Cernockỳ, J., 2019. Bayesian HMM based x-vector clustering for speaker diarization. In: Proc. Interspeech 2019. pp. 346–350.

Dimitriadis, D., Fousek, P., 2017. Developing on-line speaker diarization system. In: Proc. Interspeech 2017. pp. 2739–2743.

Dupuy, G., Rouvier, M., Meignier, S., Esteve, Y., 2012. I-vectors and ILP clustering adapted to cross-show speaker diarization. In: Thirteenth Annual Conference of the International Speech Communication Association.

El Shafey, L., Soltau, H., Shafran, I., 2019. Joint speech recognition and speaker diarization via sequence transduction. In: Proc. Interspeech 2019. pp. 396–400.

Flemotomos, N., Georgiou, P., Narayanan, S., 2020. Linguistically aided speaker diarization using speaker role information. In: Proc. Odyssey 2020 the Speaker and Language Recognition Workshop. pp. 117–124.

Garcia-Romero, D., Snyder, D., Sell, G., Povey, D., McCree, A., 2017. Speaker diarization using deep neural network embeddings. In: 2017 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 4930–4934.

Goldberg, Y., Levy, O., 2014. word2vec explained: Deriving Mikolov et. al.'s negative-sampling word-embedding method. arXiv preprint arXiv:1402.3722.

Graves, A., 2013. Generating sequences with recurrent neural networks. arXiv preprint arXiv:1308.0850.

Gupta, V., 2015. Speaker change point detection using deep neural nets. In: 2015 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 4420–4424.

Hinton, G.E., Srivastava, N., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.R., 2012. Improving neural networks by preventing co-adaptation of feature detectors. arXiv preprint arXiv:1207.0580.

Howard, J., Ruder, S., 2018. Universal language model fine-tuning for text classification. In: Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers). pp. 328–339.

Huang, Z., Watanabe, S., Fujita, Y., García, P., Shao, Y., Povey, D., Khudanpur, S., 2020. Speaker diarization with region proposal network. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 6514–6518.

India Massana, M.À., Rodríguez Fonollosa, J.A., Hernando Pericás, F.J., 2017. LSTM neural network-based speaker segmentation using acoustic and language modelling. In: Proc. Interspeech 2017. pp. 2834–2838.

Jati, A., Georgiou, P., 2017. Speaker2Vec: Unsupervised learning and adaptation of a speaker manifold using deep neural networks with an evaluation on speaker segmentation. In: Proc. Interspeech 2017. pp. 3567–3571.

Jozefowicz, R., Vinyals, O., Schuster, M., Shazeer, N., Wu, Y., 2016. Exploring the limits of language modeling.

Kenny, P., 2010. Bayesian speaker verification with heavy-tailed priors. In: Odyssey. p. 14.

Kenny, P., Boulianne, G., Ouellet, P., Dumouchel, P., 2007. Joint factor analysis versus eigenchannels in speaker recognition. IEEE Trans. Audio Speech Lang. Process. 15 (4), 1435–1447.

Kim, Y., Jernite, Y., Sontag, D., Rush, A.M., 2016. Character-aware neural language models. In: AAAI. pp. 2741–2749.

Le Lan, G., Charlet, D., Larcher, A., Meignier, S., 2017. A triplet ranking-based neural network for speaker diarization and linking. In: Proc. Interspeech 2017. pp. 3572–3576.

Lin, Q., Yin, R., Li, M., Bredin, H., Barras, C., 2019. LSTM based similarity measurement with spectral clustering for speaker diarization.

Mikolov, T., Chen, K., Corrado, G., Dean, J., 2013a. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.

Mikolov, T., Karafiát, M., Burget, L., Cernockỳ, J., Khudanpur, S., 2010. Recurrent neural network based language model. In: Proc. Interspeech 2010. p. 3.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G.S., Dean, J., 2013b. Distributed representations of words and phrases and their compositionality. In: Advances in Neural Information Processing Systems. pp. 3111–3119.

Park, T.J., Georgiou, P., 2018. Multimodal speaker segmentation and diarization using lexical and acoustic cues via sequence to sequence neural networks. In: Proc. Interspeech 2018. pp. 1373–1377.

Park, T.J., Han, K.J., Huang, J., He, X., Zhou, B., Georgiou, P., Narayanan, S., 2019. Speaker diarization with lexical information. In: Proc. Interspeech 2019. pp. 391–395.

Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K., Zettlemoyer, L., 2018. Deep contextualized word representations. In: Proceedings of NAACL-HLT. pp. 2227–2237.

Povey, D., Ghoshal, A., Boulianne, G., Burget, L., Glembek, O., Goel, N., Hannemann, M., Motlicek, P., Qian, Y., Schwarz, P., Silovsky, J., Stemmer, G., Vesely, K., 2011. The kaldi speech recognition toolkit. In: IEEE 2011 Workshop on Automatic Speech Recognition and Understanding. IEEE Signal Processing Society, IEEE Catalog No.: CFP11SRW-USB.

Povey, D., Peddinti, V., Galvez, D., Ghahremani, P., Manohar, V., Na, X., Wang, Y., Khudanpur, S., 2016. Purely sequence-trained neural networks for ASR based on lattice-free MMI. In: Proc. Interspeech 2016. pp. 2751–2755.

Prince, S., Li, P., Fu, Y., Mohammed, U., Elder, J., 2012. Probabilistic models for inference about identity. IEEE Trans. Pattern Anal. Mach. Intell. 34 (1), 144–157.

Radford, A., Wu, J., Child, R., Luan, D., Amodei, D., Sutskever, I., 2019. Language models are unsupervised multitask learners. OpenAI Blog 1 (8), 9.

Sell, G., Garcia-Romero, D., 2014. Speaker diarization with PLDA i-vector scoring and unsupervised calibration. In: Spoken Language Technology Workshop (SLT), 2014 IEEE. IEEE, pp. 413–417.

Serban, I.V., Pineau, J., 2015. Text-based speaker identification for multi-participant opendomain dialogue systems. In: NIPS Workshop on Machine Learning for Spoken Language Understanding. Montreal, Canada.

Shum, S., Dehak, N., Chuangsuwanich, E., Reynolds, D., Glass, J., 2011. Exploiting intra-conversation variability for speaker diarization. In: Twelfth Annual Conference of the International Speech Communication Association.

Snyder, D., Ghahremani, P., Povey, D., Garcia-Romero, D., Carmiel, Y., Khudanpur, S., 2016. Deep neural network-based speaker embeddings for end-to-end speaker verification. In: Spoken Language Technology Workshop (SLT), 2016 IEEE. IEEE, pp. 165–170.

Srivastava, R.K., Greff, K., Schmidhuber, J., 2015a. Highway networks. arXiv preprint arXiv:1505.00387.

Srivastava, R.K., Greff, K., Schmidhuber, J., 2015b. Training very deep networks. In: Advances in Neural Information Processing Systems. pp. 2377–2385.

Sun, G., Zhang, C., Woodland, P.C., 2019. Speaker diarisation using 2D self-attentive combination of embeddings. In: 2019 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 5801–5805.

Sundermeyer, M., Schlüter, R., Ney, H., 2012. LSTM neural networks for language modeling. In: Proc. Interspeech 2012. pp. 194–197.

Tranter, S., Reynolds, D.A., 2004. Speaker diarisation for broadcast news. In: Odyssey.

Tranter, S.E., Reynolds, D.A., 2006. An overview of automatic speaker diarization systems. IEEE Trans. Audio Speech Lang. Process. 14 (5), 1557–1565.

Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, Ł., Polosukhin, I., 2017. Attention is all you need. In: Advances in Neural Information Processing Systems. pp. 5998–6008.

Wan, L., Wang, Q., Papir, A., Moreno, I.L., 2018. Generalized end-to-end loss for speaker verification. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 4879–4883.

Wang, Q., Downey, C., Wan, L., Mansfield, P.A., Moreno, I.L., 2018. Speaker diarization with lstm. In: 2018 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 5239–5243.

Wang, J., Xiao, X., Wu, J., Ramamurthy, R., Rudzicz, F., Brudno, M., 2020. Speaker diarization with session-level speaker embedding refinement using graph neural networks. In: 2020 IEEE International Conference on Acoustics, Speech and Signal Processing. ICASSP, IEEE, pp. 7109–7113.

Werbos, P.J., 1990. Backpropagation through time: what it does and how to do it. Proc. IEEE 78 (10), 1550–1560.

Wisniewksi, G., Bredin, H., Gelly, G., Barras, C., 2017. Combining speaker turn embedding and incremental structure prediction for low-latency speaker diarization.

Woubie, A., Luque, J., Hernando, J., 2016. Improving i-vector and PLDA based speaker clustering with long-term features. In: Proc. Interspeech 2016. pp. 372–376.

Yang, Z., Dai, Z., Yang, Y., Carbonell, J., Salakhutdinov, R.R., Le, Q.V., 2019. Xlnet: Generalized autoregressive pretraining for language understanding. In: Advances in Neural Information Processing Systems. pp. 5753–5763.

Yin, R., Bredin, H., Barras, C., 2017. Speaker change detection in broadcast TV using bidirectional long short-term memory networks. In: Proc. Interspeech 2017. pp. 3827–3831.

Yin, R., Bredin, H., Barras, C., 2018. Neural speech turn segmentation and affinity propagation for speaker diarization. In: Proc. Interspeech 2018. pp. 1393–1397.

Zajıc, Z., Hrúz, M., Müller, L., 2017. Speaker diarization using convolutional neural network for statistics accumulation refinement. In: Proc. Interspeech 2017). pp. 3562–3566.

Zilly, J.G., Srivastava, R.K., Koutnık, J., Schmidhuber, J., 2017. Recurrent highway networks. In: International conference on machine learning. PMLR, pp. 4189–4198.