INTERSPEECH 2021
30 August – 3 September, 2021, Brno, Czechia

A Thousand Words are Worth More Than One Recording: *Word-Embedding* Based Speaker Change Detection

Or Haim Anidjar[1], *Itshak Lapidot*[2,3], *Chen Hajaj*[4], *Amit Dvir*[1]

[1]Department of Computer Science, Ariel University, Israel
[2]Afeka Tel-Aviv Academic College of Engineering, ACLP, Israel
[3]Avignon University, LIA, France
[4]Department of Industrial Engineering and Management, Ariel University, Israel

orhaim.anidjar@ariel.ac.il

Abstract

Speaker Change Detection (SCD) is the task of segmenting an input audio-recording according to speaker interchanges. This task is essential for many applications, such as automatic voice transcription or Speaker Diarization (SD). This paper focuses on the essential task of audio segmentation and suggests a word-embedding-based solution for the SCD problem. Moreover, we show how to use our approach in order to outperform voice-based solutions for the SD problem. We empirically show that our method can accurately identify the speaker-turns in an audio-recording with 82.12% and 89.02% success in the Recall and F1-score measures.

Index Terms: Speech Recognition, Speaker Change Detection, Speaker Diarization, Word Embedding, Clustering.

1. Introduction

Speaker Change Detection (SCD) is a common and important task in many fields. For instance, in the tasks of automatic transcription of broadcast news and audio-indexing [1, 2], the speech signal contains different kinds of speech segments, such as music, telephone speech, and advertisements. Another example of the SCD importance is the Speaker Diarization (SD) problem [3, 4, 5, 6, 7] which generally consists of two main subtasks: (i) speaker segmentation (or SCD), and (ii) speaker segments clustering. In this paper, we tackle both the closed-set [7] (e.g. closed and limited set of TV sitcoms or radio programs) and open-set [8] variants of the SCD problem.

The SCD problem is closely related to its ancestor problem, the Change-Point-Detection (CPD) one [9, 10]. This can be seen in [11] which presented an SCD system based on LSTM Neural Networks using both acoustic data and linguistic content, or in [12] which developed a segmentation based algorithm for text-dependent speaker recognition. Additional approaches for the textual SCD were presented in [13], who designed a textual based solution for the textual SCD by feeding an LSTM Neural Network with one-hot vectors, which is inappropriate to our method as our dataset contains 81,301 different words, and thus one-hot-vectors are utilizing high computational resources. This computational demand, in turn, may conflict with the applicability of our approach, as in applications such as SD, a one-hot-vector approach would be time-consuming compared to a fixed-length approach. Moreover, one-hot-vectors are context-free, that is - the representation of two semantically similar words would probably produce extremely low vectorial correlation.

A further approach is [14] who formulated the text-based SCD as a matching problem of utterances before and after a certain decision point, by proposing a hierarchical recurrent neural network (RNN) with static sentence-level attention, that encodes sentences rather than words. As in some datasets the speaker-turns are not that frequent, speaker sentences might be way too long and thus would consume an immense amount of computation resources, unlike fixed-length word embeddings. In more complex datasets, such as the DIHARD I & II [15, 16, 17, 18] challenges, the intention is to improve the robustness of diarization systems to variability in recording equipment, noise conditions, and the conversational domain. Another example of a text-based SCD work is [8], that has proposed a hybrid module for the SCD problem, by the construction and training of a Sequence-2-Sequence [19] neural network that is trained on both lexical and acoustic features.

This paper[1] presents the benefits of tackling the SCD problem from a textual standpoint, in order to motivate the use of transcripts; in some domains (e.g., testimonies of minors or persons of interest), vocal data may not be *analyzed* (yet, exist) due to privacy or similar concerns, but only the transcript. However, as we tackle the use-case of the results returned from a Speech-2-Text engine, we can assume that the transcripts of the audio content are available. Thus, we introduce the **TSCD** approach, which stands for **T**extual **S**peaker **C**hange **D**etection. We focus on segments division, and suggest a transition from the vocal domain to the textual and word-embedding based one. In addition, some audio-based features are considered as well, i.e. meta-data features (Section 2.1).

One of the main contributions of the **TSCD** model, is that it yields a robust model which is invariant to the number of speakers in the dataset. Moreover, the TSCD model is able to replace the SCD component in supervised SD systems [7] (Section 4.1). In addition, we show that the model is independent of the speaker identity, i.e., successfully identifies the speaker-turns in an environment of unseen speakers in the conversations.

For the sake of completeness, we show that our model can be easily plugged into SD systems and improve their performance [4, 7]. Given an audio recording, our method determines the points in time in which the speaker identity has changed with high probability. To the best of our knowledge, we are the first to propose a word-embedding solution that (I) solves the SCD variant of the SD problem, and (II) tackles the word-embedding based SCD problem with a dataset in the Hebrew language. Our empirical experimental evaluation process shows that the transition from voice analysis to an NLP-based approach, can accurately identify the SCD's for a given multi-participant conversation with 82.12% success reflected by the

[1]Extended version can be seen at: https://arxiv.org/abs/2006.01206

http://dx.doi.org/10.21437/Interspeech.2021-87

model's Recall measurement. Moreover, our speakers-number-independent approach outperforms recent voice-based solutions to the SD problem that depend on the number of speakers, which are based for example on D-vectors [4, 20] extraction from the speech signal.

2. Dataset and Feature Extraction

The dataset[2] used for this research was composed of 1,692 vocal audio-recordings in the Hebrew language collected over six months. The distribution of speakers per conversation is: 92% of the conversations are of 2 speakers, and the remaining 8% are of 3-5 speakers. The average length of each conversation is approximately 6 minutes and 24 seconds. The audio-recordings are heterogeneous and include TV shows, radio programs, and TV broadcasts, involving 1,240 different speakers and 81,301 unique words in total. Using a commercial Speech-2-Text engine[3], we converted the audio-recordings into 1,692 textual conversations. Each word in a textual conversation contains four parameters: the word itself, its start and end time, as well as speaker identity which is required to determine the locations of the speaker-turns. Note that our approach requires the speaker-turns only, rather than speaker identities, which are required mainly for SD. The Word-Error-Rate (WER) [21] of the Speech-2-Text engine is 7.2%, and the overlapping speech proportion in this dataset is 5.2%.

A natural obstacle we faced was class imbalance (discussed in Section 4); 98.5% of the samples are non-SCD examples, i.e., two consecutive words are tagged with the same speaker, whereas only 1.5% of the samples were labeled as SCD examples. Another obstacle was the dataset language. While our method does not assume any a-priori knowledge of the dataset, there are some challenges when analyzing a Hebrew dataset, that do not exist in English. For example, the word order in English is very flexible and not constant like in Hebrew. Verbs, are another example of such a challenge; In English, the phrase *'when they came'* is made up of three words, whereas in Hebrew, one lexeme can reflect the (i) verb, (ii) function word, and (iii) plural relationship. Thus, in Hebrew, structural elements such as morphology [22] need to be parsed to understand the meaning of a sentence.

2.1. Feature Extraction - Data Encoding

Each conversation of size n words, was divided into tuples of six words, using a sliding window. This procedure resulted in $n-5$ windows for each conversation, denoted by S=$\{S_1,S_2,\ldots,S_{n-5}\}$. To transform all textual conversations, we went through over all the conversations using the following procedure: Without loss of generality, for each conversation of size n, we iterated over all of the n words and divided the conversation into tuples of six words, using the sliding window method. The first window, S_1, was composed of the first six words $(W_1^1, W_1^2, .., W_1^6)$. Similarly, S_2 was composed of the next six word tuple $(W_2^1, W_2^2, \ldots, W_2^6)$ and so on to the last window, S_{n-5}.

For each S_i $(1 \leq i \leq n-5)$, S_i is embedded using Facebook's pre-trained[4] Hebrew Word-2-Vec [23] vectors. That is, the Word-2-Vec vectorial average is computed, for the first and the last three words in the sliding window S_i, denoted by $S_i^I = \{W_i^1, W_i^2, W_i^3\}$ and $S_i^{II} = \{W_i^4, W_i^5, W_i^6\}$, respectively. As any word in the vocabulary was represented as a multi-dimensional vector of size 300, this step in the feature engineering generated 600 features, for the average of the first three words and the last three words.

In addition, the feature-engineering process generated an additional 13 meta-vocal based features, for each S_i, as follows:

- **Features (1)-(6)** Duration of each word in S_i, i.e. duration of $\{W_i^1, W_i^2, \ldots, W_i^6\}$;
- **Features (7)-(12)** Speech rate of each word, represented by the length of the word (in characters), divided by its duration (i.e., end time minus the start time);
- **Feature (13)** Time elapsed between the 3^{rd} word W_i^3, and the 4^{th} word W_i^4 in S_i.
- Finally, a label ($Same$ or $Split$) was added; if there was a speaker change between the 3^{rd} and 4^{th} words, the instance was tagged as $Split$, otherwise it was $Same$.

3. Neural Network Based Classification

The feature engineering process resulted in 1,129,570 instances of sliding windows, which were converted into feature-vectors. These sliding windows were divided into training and test sets, such that sliding windows gathered from 80% of the conversations were used for training, and the remaining 20% for test. Thus, out of the 1,692 files in the dataset, 1,354 conversations served for training and the remaining 338 for test. Clearly, in the speaker-independent experiments the training and test sets are speaker disjoint, whereas the speaker-dependent experiments are including the same speakers in the training and test sets.

The chosen architecture was a Fully-Connected Neural Network with 5 layers: (i) - Input layer of size 613, (ii)-(iv) - for the 1^{st}, 2^{nd} and 3^{rd} hidden layers with a ReLU activation function follows each one, of sizes 307, 154, 77, and (v) - Output layer of size 2. In addition, a dropout layer was added between any two layers, with dropout value of $p = 0.5$. As the problem is a supervised classification one, the Cross-Entropy [24] loss was used. As for the Neural Network's learning rate, the Adam Optimizer [25] was used, with an initial value of $\alpha = 1 \cdot 10^{-4}$. Consequently, a Softmax [26] activation function was chosen to represent the network's output, in the form of a probability vector of size 2 (for the classes $Split$ and $Same$).

The last thing that was considered before training the neural network was the imbalanced ratio between the examples from the $Split$ and $Same$ classes, as in a given natural conversation, the proportion of speech is higher than the number of turns. Thus, the relative weight of the examples from each class was taken into account; when defining the loss function of the neural network, the weight of each class was defined as the inverse of the number of its examples.

4. Experimental Evaluation

In this section we show that using NLP techniques for SCD in a multi-speaker environment can outperform classical speech techniques. Recall that unlike typical voice-based solutions [4, 7] which require speaker identities, the TSCD model only needs to know whether an interchange has occurred or not.

[2]One can get access to a subset of conversations from the dataset, as well as to the code, at the following Github link: github.com/honeyJarPhD/TextualSpeakerChangeDetection. For full dataset access, please e-mail the corresponding author.

[3]Visit Almagu official website at: www.almagu.com/voicetotext

[4]The Word-2-Vec pre-trained vectors vocabulary in Hebrew is available at https://fasttext.cc/docs/en/crawl-vectors.html

Our evaluation was conducted on a Dell XPS 8920 computer, Windows 10 64Bit OS with 3.60GHz Intel Core i7-7700 CPU, 16GB of RAM, and NVIDIA GeForce GTX 1050 GPU 2GB, using PyTorch (v1.4.0), and scikit-learn (v0.23.2).

4.1. Improving a Diarization System with TSCD

In order to emphasize the importance and influence of an SCD component over an SD system, the TSCD model was examined as a commutative SCD component to the D-vector based SCD component presented by [4] for the UIS-RNN [7] training. By employing speech analysis for the SD task, Zhang et al. [7] proposed a fully supervised SD system, based on the extraction of D-vectors from input utterances [4]. Denote by D-VEC the model from [4], this section presents a comparison of SCD components, between the TSCD, UIS-RNN and D-VEC ones. The performance of the components was calculated with the Word-level Diarization Error Rate (WDER) [27, 28], as well as the standard diarization metric, i.e. Diarization Error Rate (DER) [3, 29], using the following procedure:

(**1**) **Interchanges Classification** - For each tested audio-recording file, denoted by $tf_i (1 \leq i \leq 338)$, we computed the speaker interchanges through tf_i's timeline using the TSCD and D-VEC models output. The set of points in time for which the TSCD and D-Vec models have detected speaker interchanges in tf_i are denoted as C_{TS_i} and C_{DV_i}, respectively.

(**2**) **Segments Extraction & D-vector Re-Computation** - For each tf_i and its speaker segments bounded by C_{TS_i}, C_{DV_i}, the corresponding D-vectors were recalculated. For each $c_{ts_j} \in C_{TS_i}$, $(1 \leq j \leq |C_{TS_i}|)$ and for each $c_{dv_k} \in C_{DV_i}$, $(1 \leq k \leq |C_{DV_i}|)$, a 256-dimensional D-vector was computed, framed by any speaker interchange of the two models, denoted by c_{ts}, c_{dv}. That is, according to the TSCD model's output, the first D-vector is framed from time $t = 0$ in tf_i up to $t = c_{ts_1}$; and so on to the last segment (starting in $t = c_{ts_{|C_{TS_i}|}}$ up to the end of tf_i). Clearly, the computation for each segment was done over a D-vector, since after the TSCD model's prediction, the models are compared in the same manner (D-vectors computation). Then, a parallel D-vector extraction was done for the D-VEC model's output, in accordance with all of its speaker interchanges $c_{dv} \in C_{DV_i}$;

(**3**) **Clustering and Classification** - Having the D-vectors that represent the speaker interchange sets (C_{TS_i}, C_{DV_i}) of any tf_i, the final step to implement a diarization system is a clustering module for the TSCD and D-VEC [4] models, and classification of speaker identities for the UIS-RNN [7] model. The details of each model supplementation were split into an SD system, as follows;

- (a) **TSCD** - For each tf_i and its speaker segments, and for each such a segment in C_{TS_i}, two clustering algorithms were applied over the TSCD model's output. The clustering algorithms are the K-Means and Spectral Clustering [30] ones, as appeared in [7] for comparison to baseline algorithms. The similarity metric that is used is the cosine similarity (as in [5]).

  As for the selection of the amount of clusters (i.e. number of speakers) for each algorithm, this number is adaptively determined as in [5], i.e. the "elbow" of the derivatives of conditional Mean Squared Cosine Distances[5] (MSCD) between each embedding to its cluster centroid. Finally, the DER and WDER were calculated over the clustering results over C_{TS_i} speaker's segments of tf_i.

- (b) **D-VEC** - Similar to the TSCD model, for each tf_i and its speaker segments in C_{DV_i}, the K-Means and Spectral Clustering algorithms were applied, as well as for the selection of the amount of clusters. Finally, the DER and WDER were calculated over the clustering results over C_{DV_i} speaker's segments of tf_i.
- (c) **UIS-RNN** - As the UIS-RNN model is a supervised SD system, then for each tf_i and its speaker segments in C_{DV_i}, the UIS-RNN was employed in order to assign speaker identities for each segment. As such, each segment classification was needed. Finally, the DER and WDER were calculated over the classified speaker identities over C_{DV_i} speaker's segments of tf_i.

(**4**) **DER & WDER Calculation** - For each model, the DER and WDER measures are calculated for each tested audio-recording $\{tf_i\}_{i=1}^{338}$. The final reported DER & WDER for all the tested audio-recordings is calculated as:

- $\frac{\sum DER(tf_i)_{i=1}^{338}}{338}$ for DER;
- $\frac{\sum WDER(tf_i)_{i=1}^{338}}{338}$ for WDER.

Table 1 suggests that the TSCD model as an SCD component outperforms the overall diarization system's performance, of the proposed ones in [4, 7].

Table 1: *DER & WDER results table for SD systems [4, 7] based on D-vectors extraction and TSCD model's output.*

Model	DER	WDER
TSCD-KM	**11.45**	**10.28**
TSCD-SPEC	11.73	10.33
D-VEC-KM	15.84	17.50
D-VEC-SPEC	15.92	17.57
UIS-RNN	26.47	29.21

Our hypothesis for this DER & WDER values degradation, compared with the English datasets the were used in the baselines methods [4, 7], is due to the following reasons:

(i) Unlike [4, 7], our dataset is in the Hebrew language rather than English. The availability of resources, training data, and benchmarks in English leads to a disproportionate focus on the English language. Thus, in this case, the Hebrew language is discriminated disproportionately and suffers from performance degradation, as can be seen in our work (yet, it poses a great challenge, and this is why we tackled this dataset);

(ii) The dataset originates from TV shows and panel interviews with many speakers in each transcription. Unlike datasets that have been used in [4, 7] (like the well-known TIMIT dataset or the ICSI one), the overlap between speakers is a bit higher in our dataset. Correspondingly, we can expect a higher DER or WDER than the ones seen in [4, 7].

In addition, for TSCD-based SCD systems, one can note that the DER results are a bit higher than the WDER ones, both for the TSCD model and the method in [4, 7]. As the speakers-turns are only considered between one speaker and another, rather than one speaker and its silent parts as well, we can expect that the DER would be a bit higher than the WDER.

4.2. Number of Speakers Influences UIS-RNN

The work presented in [7] utilized the speech embedding extraction module in [4], by learning the extracted speaker-discriminative embeddings, or D-vectors, from input utterances.

[5] the cosine distance is defined as $d(x, y) = \frac{1-cos(x,y)}{2}$.

In order to show the TSCD model's robustness to the number of speakers, and compare it to the speech embedding extraction module, the SD problem was translated into an SCD instance.

For this purpose, the TSCD model was compared with the one presented in [7], with a variable number of speakers $k \in \{8, 20, 50, 100, 200, 250, 500, 1000\}$ in the dataset, and with 10 different random speakers sampling's identities and utterances, using the following procedure:

(**1**) Splitting the dataset into textual speaker utterances. Recall that each word duration is known (by subtracting the "to" column, from the "from" column), as well as the speaker's identity;

(**2**) Segmenting each audio file according to its timeline speaker utterances. Then, the neural network presented in [4] was trained to extract the speech embedding vectors for each speaker utterance, i.e. the D-vectors;

(**3**) The D-vectors extracted for each speaker utterance formed a matrix of size $N_{seg} \times M_{emb}$, where N_{seg} represents the number of segments extracted for each speaker utterance, and M_{emb} is the D-vector which $\in \mathbb{R}^{256}$. The segments for each speaker utterance were extracted using a Voice Activity Detector (VAD) engine ($librosa$, $webrtcvad$ Python libraries);

(**4**) Then, the speech embedding matrices were split into training and test sets, with a random 80%-20% division of the data samples respectively;

(**5**) The work in [7] presents a fully-supervised SD model termed UIS-RNN, and assigns a speaker identity to each speaker utterance in the dataset. As such, the UIS-RNN presented in [7] was trained by the training matrices, each of which was represented by a set of D-vectors extracted as a function of the number of segments in each speaker utterance.

For each k, this procedure was run 10 times, each time with a randomly chosen set of speakers. In order to calculate the error rate of each result for each k, the model's inference results were converted into SCD based results, i.e., a binary classification testing. For each speaker utterance in the test set, the UIS-RNN model produced a classification vector of size N_{seg}. Hence, it was possible to conduct **Type I, II** [31] errors analyses for each classification vector, which later resulted in Precision, Recall, and F1-Score calculations. For a given conversation split into $U \in \mathbb{N}$ speaker utterances, the **Type I, II** errors analysis was as follows:

(i) **Type I** - occurs any time when in a given speaker utterance u_i (for $1 \leq i \leq |U|$), the first segment label in its corresponding classification vector had different values than each of the classification vector elements. More specifically, for a classification vector c_i (that represents the identity assignments for u_i segments) of size ℓ, an error is counted each time $c_i[j]$ is not equal to $c_i[1]$ for $2 \leq j \leq \ell$. Each such mismatch implies that the UIS-RNN detected a change point and tagged $Split$, rather than $Same$;

(ii) **Type II** - occurs any time when the last segment of a speaker utterance u_i classification, and the first segment of u_{i+1} classification are equal (for $1 \leq i \leq |U| - 1$). This is due to the fact that between any proceedings u_i and u_{i+1} there must have been an interchange, i.e. the UIS-RNN tagged $Same$, rather than $Split$.

The results of this comparison are presented in Table 2. In addition, the models were compared with the whole dataset, i.e. with all of the 1,240 speakers. It is clear that from 250 speakers and above, the TSCD model outperforms the UIS-RNN model in all measurements. Both models maintained their Precision due to the extreme class imbalance, but as the number of speakers in the dataset increases to 250 and above - the TSCD outperforms the UIS-RNN. The UIS-RNN model's Recall (and as such the F1-Score) deteriorated since many more speaker identities were needed to be classified.

Table 2: *Results table for the comparison of the TSCD and the UIS-RNN. For each k, the average values of the Precision, Recall and F1-Score are presented for 10 data samplings. For every result, the Standard-Deviation σ is between* 0.13 - 3.36. *For 1,240 speakers, the process was run only once ($\sigma = 0$).*

Random Model List - **TSCD, UIS-RNN**						
K	**TSCD**			**UIS-RNN**		
	Prec.	Rec.	F1.	Prec.	Rec.	F1.
8	**96.56**	95.11	95.78	95.94	**97.74**	**96.65**
20	**96.34**	94.43	**95.30**	94.87	**96.67**	95.15
50	**96.54**	91.21	93.55	94.56	**96.60**	**94.98**
100	**96.90**	87.42	91.50	94.26	**95.83**	**95.03**
200	**97.09**	84.50	89.79	94.48	**92.14**	**93.29**
250	**97.03**	**83.72**	**89.30**	95.84	82.05	87.76
500	**97.17**	**83.44**	**89.15**	96.05	79.40	86.18
1000	**97.13**	**82.77**	**88.71**	96.04	79.21	86.04
1240	**97.19**	**82.12**	**89.02**	95.72	79.37	86.78

5. Conclusions and Future Work

In this paper we demonstrated how to solve the SCD component of the SD problem, using a word-embedding based technique. Better results and greater robustness have been achieved, compared to two recently developed voice-based solutions to the SD problem, as well as robustness to the introduction of previously unseen speakers. Moreover, existing SD systems have been improved, with the TSCD model as their SCD component. Yet, the SCD problem is not completely solved. Hence, and due to the strength of the textual approach, one possible future research would be to address the SCD problem from the multi-lingual aspect, so that one solution would address multiple languages at once. In addition: (1) more sophisticated network architectures, as well as novel feature engineering, may improve our results, and we encourage this direction for future research; (2) Moreover, we believe that further research should be done regarding the trade-off between the WER values of Speech-2-Text engines, and the efficiency of our method. That is, the method proposed in this paper would benefit from evaluation using more complex speech conditions, such as extreme noise/reverberation and higher overlapped conditions (DIHARD I & II). Consequently, the performance of our method with respect to different number of speakers in each conversation may be considered as well. In addition, we utterly consider in the future to exploit both the speech signal and the textual content in order to improve both SCD and SD systems, using multi-model techniques that combine this hybrid information.

6. Acknowledgements

The authors wish to thank IFAT Group that provided the dataset for this research. In addition, this work was supported by the Ariel Cyber Innovation Center in conjunction with the Israel National Cyber directorate in the Prime Minister's Office.

7. References

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

[2] L. Lu and H.-J. Zhang, "Speaker change detection and tracking in real-time news broadcasting analysis," in *Proceedings of the tenth ACM international conference on Multimedia*, 2002, pp. 602–610.

[3] M. Hrúz and Z. Zajíc, "Convolutional neural network for speaker change detection in telephone speaker diarization system," in *ICASSP*, 2017, pp. 4945–4949.

[4] L. Wan, Q. Wang, A. Papir, and I. L. Moreno, "Generalized end-to-end loss for speaker verification," in *ICASSP*, 2018, pp. 4879–4883.

[5] Q. Wang, C. Downey, L. Wan, P. A. Mansfield, and I. L. Moreno, "Speaker diarization with lstm," in *ICASSP*, 2018, pp. 5239–5243.

[6] S. H. Yella and H. Bourlard, "Overlapping speech detection using long-term conversational features for speaker diarization in meeting room conversations," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 22, no. 12, pp. 1688–1700, 2014.

[7] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *ICASSP*, 2019, pp. 6301–6305.

[8] T. J. Park and P. Georgiou, "Multimodal speaker segmentation and diarization using lexical and acoustic cues via sequence to sequence neural networks," *arXiv preprint arXiv:1805.10731*, 2018.

[9] I. Barnett and J.-P. Onnela, "Change point detection in correlation networks," *Scientific reports*, vol. 6, p. 18893, 2016.

[10] C. Truong, L. Oudre, and N. Vayatis, "Selective review of offline change point detection methods," *Signal Processing*, p. 107299, 2019.

[11] M. À. India Massana, J. A. Rodríguez Fonollosa, and F. J. Hernando Pericás, "Lstm neural network-based speaker segmentation using acoustic and language modelling," in *INTERSPEECH*, 2017, pp. 2834–2838.

[12] C. Luo, X. Wu, T. F. Zheng, and L. Wang, "Segmentation-based method for text-dependent speaker recognition in embedded applications," *APSIPA ASC*, 2010.

[13] Z. Zajíc, D. Soutner, M. Hrúz, L. Müller, and V. Radová, "Recurrent neural network based speaker change detection from text transcription applied in telephone speaker diarization system," in *International Conference on Text, Speech, and Dialogue*, 2018, pp. 342–350.

[14] Z. Meng, L. Mou, and Z. Jin, "Hierarchical rnn with static sentence-level attention for text-based speaker change detection," in *Conference on Information and Knowledge Management*, 2017, pp. 2203–2206.

[15] G. Sell, D. Snyder, A. McCree, D. Garcia-Romero, J. Villalba, M. Maciejewski, V. Manohar, N. Dehak, D. Povey, S. Watanabe *et al.*, "Diarization is hard: Some experiences and lessons learned for the jhu team in the inaugural dihard challenge." in *Interspeech*, vol. 2018, 2018, pp. 2808–2812.

[16] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "First dihard challenge evaluation plan," *2018, tech. Rep.*, 2018.

[17] J. Patino, H. Delgado, and N. W. Evans, "The eurecom submission to the first dihard challenge." in *INTERSPEECH*, 2018, pp. 2813–2817.

[18] N. Ryant, K. Church, C. Cieri, A. Cristia, J. Du, S. Ganapathy, and M. Liberman, "The second dihard diarization challenge: Dataset, task, and baselines," *arXiv preprint arXiv:1906.07839*, 2019.

[19] J. Gehring, M. Auli, D. Grangier, D. Yarats, and Y. N. Dauphin, "Convolutional sequence to sequence learning," in *ICML*, 2017, pp. 1243–1252.

[20] J. Jung, H. Heo, I. Yang, S. Yoon, H. Shim, and H. Yu, "D-vector based speaker verification system using raw waveform cnn," in *2017 International Seminar on Artificial Intelligence, Networking and Information Technology (ANIT 2017)*. Atlantis Press, 2017.

[21] Y.-Y. Wang, A. Acero, and C. Chelba, "Is word error rate a good indicator for spoken language understanding accuracy," in *2003 IEEE Workshop on Automatic Speech Recognition and Understanding (IEEE Cat. No. 03EX721)*. IEEE, 2003, pp. 577–582.

[22] Y. Goldberg and M. Elhadad, "Word segmentation, unknown-word resolution, and morphological agreement in a hebrew parsing system," *Computational Linguistics*, vol. 39, no. 1, pp. 121–160, 2013.

[23] T. Mikolov, I. Sutskever, K. Chen, G. S. Corrado, and J. Dean, "Distributed representations of words and phrases and their compositionality," in *Advances in neural information processing systems*, 2013, pp. 3111–3119.

[24] Z. Zhang and M. Sabuncu, "Generalized cross entropy loss for training deep neural networks with noisy labels," in *Advances in neural information processing systems*, 2018, pp. 8778–8788.

[25] D. P. Kingma and J. Ba, "Adam: A method for stochastic optimization," *arXiv preprint arXiv:1412.6980*, 2014.

[26] Z. Qin and D. Kim, "Softmax is not an artificial trick: An information-theoretic view of softmax in neural networks," *arXiv preprint arXiv:1910.02629*, 2019.

[27] P. Deléglise, Y. Esteve, S. Meignier, and T. Merlin, "Improvements to the lium french asr system based on cmu sphinx: what helps to significantly reduce the word error rate?" in *CISCA*, 2009.

[28] T. J. Park, N. Kanda, D. Dimitriadis, K. J. Han, S. Watanabe, and S. Narayanan, "A review of speaker diarization: Recent advances with deep learning," *arXiv preprint arXiv:2101.09624*, 2021.

[29] L. Sun, J. Du, T. Gao, Y.-D. Lu, Y. Tsao, C.-H. Lee, and N. Ryant, "A novel lstm-based speech preprocessor for speaker diarization in realistic mismatch conditions," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2018, pp. 5234–5238.

[30] G. Sun, Y. Cong, Q. Wang, J. Li, and Y. Fu, "Lifelong spectral clustering." in *AAAI*, 2020, pp. 5867–5874.

[31] A. Banerjee, U. Chitnis, S. Jadhav, J. Bhawalkar, and S. Chaudhury, "Hypothesis testing, type i and type ii errors," *Industrial psychiatry journal*, vol. 18, no. 2, p. 127, 2009.