INTERSPEECH 2023
20-24 August 2023, Dublin, Ireland

Lexical Speaker Error Correction: Leveraging Language Models for Speaker Diarization Error Correction

Rohit Paturi, Sundararajan Srinivasan*, Xiang Li*

AWS AI Labs

paturi@amazon.com, sundarsr@amazon.com, xiangzai@amazon.co.uk

Abstract

Speaker diarization (SD) is typically used with an automatic speech recognition (ASR) system to ascribe speaker labels to recognized words. The conventional approach reconciles outputs from independently optimized ASR and SD systems, where the SD system typically uses only acoustic information to identify the speakers in the audio stream. This approach can lead to speaker errors especially around speaker turns and regions of speaker overlap. In this paper, we propose a novel second-pass speaker error correction system using lexical information, leveraging the power of modern language models (LMs). Our experiments across multiple telephony datasets show that our approach is both effective and robust. Training and tuning only on the Fisher dataset, this error correction approach leads to relative word-level diarization error rate (WDER) reductions of 15-30% on three telephony datasets: RT03-CTS, Callhome American English and held-out portions of Fisher.

Index Terms: Speaker Diarization, Large Language Models, Automatic Speech Recognition, Error Correction

1. Introduction

Speech transcription systems have advanced significantly in the past decade but even with these remarkable advances, machines have difficulties understanding natural conversations with multiple speakers such as in broadcast interviews, meetings, telephone calls, videos or medical recordings. One of the first steps in understanding natural conversations is to recognize the words spoken and their corresponding speakers. Speaker Diarization (SD) is the process of determining "who spoke when" in a multi-speaker audio signal and is a key component in any speech transcription system. SD is used in conjunction with Automatic Speech Recognition (ASR) to assign a speaker label to each transcribed speaker turn and has widespread applications in generating meeting/interview transcripts, medical notes, automated subtitling and dubbing, downstream speaker analytics, among others (we refer to this combined system as SD-ASR in this paper). This is typically performed in multiple steps that include (1) transcribing the words using an ASR system, (2) predicting "who spoke when" using a speaker diarization (SD) system, and, finally, (3) reconciling the output of those two systems.

Recent advances in SD systems are outlined in [1] and the independent module optimized SD systems typically consists of the following main sub-tasks: (a) segment the input audio into speech segments using a Voice activity detector (VAD), (b) generate speaker segments from the speech segments by either using a uniform window size [2–4] or by detecting speaker turns [5–7], (c) extract speaker embeddings [2, 8–10] for each of the speaker segments and (d) cluster the resulting speaker embeddings using clustering algorithms like Spectral Clustering [2], Agglomerative Hierarchical Clustering [4] among others. These sub-tasks of most of the diarization systems in literature rely only acoustic information and can thus lead to speaker errors, mainly around the speaker turns. This can happen in uniform speaker segmentation as long segments very likely contain speaker turn boundaries, while short segments carry insufficient speaker information. It is also shown that detecting speaker turns using only acoustic information is also error-prone [7]. In addition to the SD errors, speakers can be attributed to the wrong words in the SD-ASR reconciliation phase due to errors in ASR word timings. Reconciliation errors can also occur in regions of speech overlap as SD can identify one of the speakers while ASR can identify words corresponding to a different speaker.

Lexical information can contain complementary information which can be very useful in accurately predicting speaker turns [6, 7]. For instance, analyzing only the written transcript of a conversation such as "how are you i am good", enables us to infer that there is likely a speaker change between the utterances "how are you" and "i am good". There have been a handful of works [6, 7, 11–14] which leverage the ASR transcripts to infuse lexical information in the SD module. In [7], lexical cues are used to estimate the speaker turns for diarization. [11] made use of turn probabilities from lexical cues in the clustering stage by enhancing the adjacency matrix. Though these approaches showed good SD improvements, these systems can still produce errors around speaker turns due to ASR and Diarization errors in overlapped speech as well are sensitive to ASR word timings as they rely on ASR timings in the diarization sub-tasks as well as in the Reconciliation phase. [12] modeled SD and ASR jointly but is confined to 2 speakers with specific distinct roles.

In this paper, we propose a Speaker Error Correction (SEC) module which can correct speaker errors at the word level without modifying the underlying ASR or the acoustic SD system. This SEC module makes use of the any of the readily available pre-trained LMs [15–18] to infuse the lexical knowledge to correct speaker errors while also leveraging speaker scores from the SD system to prevent over-corrections. The reliance on LMs also significantly reduces the amount of speaker labelled text data needed to train the system. Our approach has components which are modular and don't need paired audio, text data to train while only needing a small amount of paired data for fine-tuning. This approach is also easier to integrate with existing systems than other lexical-based diarization approaches, since the first-pass acoustic SD system can be run independently of the ASR system. Using experiments across three telephony datasets, we demonstrate that the proposed system is both effective as well as capable of generalization.

*These authors contributed equally to this work

10.21437/Interspeech.2023-1982

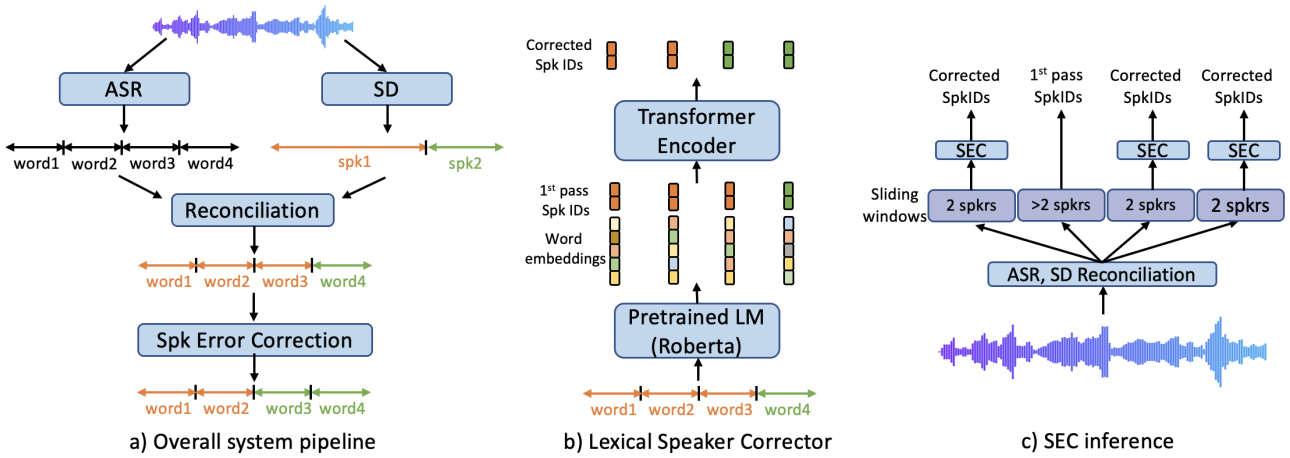

Figure 1: *(a) Speaker Error Correction as a 2nd-pass post-processing step to the traditional SD-ASR system, (b) Lexical SEC: Word embeddings from the LM, Speaker IDs from SD are passed to the Transformer Encoder to get the corrected Speaker IDs, (c) SEC inference performed on sliding windows with 2 hypothesis speakers.*

2. Speaker Error Corrector

The overall pipeline of the proposed two-pass Speaker Error Corrector (SEC) framework is shown in Fig 1a. The conventional Speaker Transcription system consists of an ASR module, a SD module and a reconciliation stage. The SEC follows the reconciliation stage and takes in two streams of inputs: acoustic features from the SD module and lexical features from the ASR module. The ASR and acoustic SD models can continue to run in parallel, making it easier to integrate with existing systems. The core component of the SEC is the Lexical Correction module which takes in the transcribed words from ASR along with the speaker labels from the SD module. These are explained in more detail in the following sub-sections.

2.1. Lexical Diarization Corrector

While lexical features have complementary information to the acoustic features and can be leveraged to correct some of the errors from a naïve reconciliation of ASR and SD, lexical features alone can't accurately predict the speaker labels especially in realistic conversations. So, we propose a simple yet efficient way to correct the speakers based on both the decisions from the 1st pass diarizer and the ASR transcriptions. Our proposed Lexical Speaker Error Corrector consists of two main components: a backbone language model (LM) and a Transformer Encoder Front-end to predict the speaker labels. After reconciling ASR and diarization outputs, we have speaker labels $\{\boldsymbol{S_i}\}_{i=1}^{N}, \boldsymbol{S_i} \in \mathbb{R}^{1\times K}$ for every word $\{\boldsymbol{W_i}\}_{i=1}^{N}$, where N is the number of words in the sequence and K is the number of speakers the SEC is trained to handle. The words W_i are tokenized and passed to the backbone LM to obtain contextual word embeddings $\{\boldsymbol{E_j}\}_{j=1}^{M}, \boldsymbol{E_j} \in \mathbb{R}^{1\times W}$ where M is the number of tokens in the word sequence and W is the word embedding dimension. The word level speaker labels S_i are mapped to token level by mapping the speaker ID corresponding to the word to its first token if the word has more than 2 tokens and assigning a special "don't care" token to any of the subsequent tokens of the word. These token level embeddings E_j are concatenated with the speaker IDs S_j to form the fused features for the Front-end Transformer Encoder as shown in Figure 1b. The posteriors from the Front-end Encoder $\{\boldsymbol{L_{ij}}\}_{j=1}^{K}, \boldsymbol{L_i} \in \mathbb{R}$ are used to optimize the classification loss on the ground-truth speaker labels.

2.2. Training Methodology

The SEC model can be trained only using speaker turn transcripts and doesn't require paired audio data and we show that training the lexical corrector on just the transcripts also improves the performance of the baseline. Since the relatively smaller number of speaker errors produced by 1st pass diarizer system limits the training of the error corrector, we train the corrector by simulating speaker errors based on the ground truth as well by simulating ASR substitution errors.

We define the probability of ASR errors as P_{ASR} and the probability of speaker errors as P_{Spk}. Setting $P_{ASR} = 1$ implies that all the words in the training transcripts are substituted with random words and $P_{ASR} = 0$ implies the original ground-truth transcripts. Similarly, $P_{Spk} = 1$ implies all the speaker labels are randomly substituted whereas $P_{Spk} = 0$ implies the ground-truth speaker labels. We simulate ASR, Speaker errors using a curriculum learning paradigm [19] to make sure that we don't under or over correct the speakers and balance the information flow from the SD labels and ASR word lexical information. We start the curriculum for P_{Spk} at a low value and increase P_{Spk} as the training progresses. Conversely, P_{ASR} starts at a high value at the first epoch and decreases as the training progress. The intuition for this curriculum with P_{ASR} being higher and P_{Spk} being lower in the initial epochs is to train the model without any meaningful lexical information and to train the model to at least copy the 1st pass speaker labels in the initial epochs. More meaningful lexical information with a smaller P_{ASR} is used in the later epochs along with a higher P_{Spk} to train the model on more complex speaker errors as the training progresses.

In addition to the errors simulated text data, we also use paired audio data to train, fine-tune the model on real data. For this, we generate speaker labels using the baseline 1st SD and use the ground-truth speaker labels as the targets. In this work, we train the SEC on two speaker cases, i.e., $K = 2$

2.3. Inference Setup

During inference, we perform error correction on sliding windows with a fixed number of ASR transcribed words as shown in Fig 1c. Though the lexical corrector is trained to only correct two speakers locally, we can still handle use-cases where more than two speakers are detected globally in the audio. We achieve this

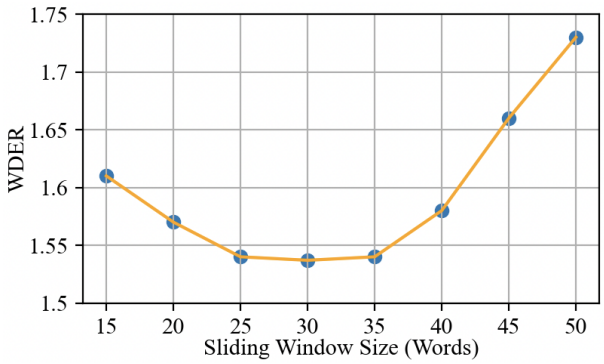

Figure 2: *Window Size tuning on the validation set*

by only correcting sliding windows comprising of two speakers and by bypassing the remaining windows as shown in Figure 1c. The size of the sliding window is a parameter we tune on a validation set.

3. Experiments

3.1. Data and Metrics

In this work, we use the full Fisher dataset [20, 21] to train the Speaker Corrector system. We split the Fisher data into train, validation and test splits as defined in [22]. We also the Fisher train set to fine-tune the backbone LM model as well as to train, fine-tune the Corrector model. We only use the Fisher validation split for tuning our model. For evaluation, in addition to Fisher test split, we use the standard dev, test splits of CALLHOME American English (CHAE) [23] and RT03-CTS [24] which are majorly two speaker calls. We also evaluate on the two-speaker only set of CHAE, the CH-109 dataset [25] by fixing the number of clusters to 2 as well as automatically determining the number of speakers in the 1st pass SD system.

In order to evaluate the full ASR, SD system, we use the Word Diarization Error Rate (WDER) proposed in [12] as it aptly captures both ASR and SD errors at the word level. We also account for words transcribed in regions of speech overlap in the WDER metric. This is achieved by using asclite [26] as it can align multiple speaker hypotheses against multiple reference speaker transcriptions and can also efficiently handle words in regions of speaker overlaps.

3.2. Baseline System

Our baseline SD system follows the pipeline in [2] and consists of a speaker embedding model followed by Spectral Clustering and the number of speakers is identified using the maximum eigengap of the Spectral Clustering. The speaker embedding model is based on a ResNet-34 architecture trained with a combination of classification, metric loss [27] and channel loss [28] on about 12k speakers and 4k hours of CTS data . We use a uniform speaker segmentation [3, 4] with a duration of 500ms to extract the speaker embeddings followed by the Clustering phase for the SD system. Our baseline SD system is comparable to state-of-the-art diarization systems across several datasets and achieves a a DER of 3.72 and SER of 1.1 on CHAE test set which is a stronger baseline than the one reported in [11]. We use a hybrid ASR system [29–31] with a Conformer Acoustic model [32] and a n-gram Language model trained on several tens of thousands of audio, text data. For the reconciliation phase, the SD system provides speaker turns with time boundaries and

a)

Reference (Overlapping Speech)

Start time	End time	Spkr	Text
21.10	25.73	B	you heard the topic yes do you consider any countries a threat
23.43	24.33	A	((yeah))

Baseline Hypothesis

<spk2> you heard the topic yes do you <spk1> **consider** <spk2> any countries a threat

Corrected Hypothesis

<spk2> you heard the topic yes do you **consider** any countries a threat

b)

Reference (Errors around Speaker Turns)

Start time	End time	Spkr	Text
380.51	386.22	B	everybody will say well this should have been done or that should have been done
386.63	391.05	A	well and i i think it'll be known…

Baseline Hypothesis

<spk1> everybody will say well this should have been done or that should have been <spk2> **done** well and i i think it'll be known…

Corrected Hypothesis

<spk1> everybody will say well this should have been done or that should have been **done** <spk2> well and i i think it'll be known…

Figure 3: *Correction Examples: a) Errors due to overlapping speech, b) Errors around speaker turns.*

these labels are mapped to recognized words using the associated word boundaries from the ASR system. When the speaker turn boundary falls in the middle of a word, we assign the word to the speaker with the largest overlap with the word similar to the baseline system in [12]. We use a neural-network based Speech Activity detector (SAD) similar to [33] as a front end for both SD-ASR systems above.

3.3. SEC System

For the SEC model, we use a pre-trained Roberta-base model [16] as the backbone LM and a Transformer Encoder of size 128 hidden states for the Front-end model. The curriculum for P_{ASR} starts at 1 at the 1st epoch and decreases to 0.08 in the 10th and subsequent epochs in uniform steps. The curriculum for P_{Spk} starts at 0 in the 1st epoch and increase to 0.14 in the 10th and subsequent epochs in uniform steps. The model is trained with Adam Optimizer with a batch size of 32 and an average sequence length of 30 words per batch. We use a learning rate of 1e-4 and train the model for 30 epochs on a machine with 8 GPUs.

We use the SEC as a 2nd pass post-processing step to the baseline SD-ASR system in Section 3.2. In order to determine the number of simulated errors needed to effectively train the lexical SEC to correct the speaker errors, we follow the error curricula mentioned in Section 2.4 and pick the checkpoint with P_{ASR}, P_{Spk} that achieves the lowest WDER on the Fisher validation set. The values that achieve the best validation WDER are 0.1 for both P_{ASR} and P_{Spk}. In addition to the training parameters, we also tune an inference parameter, the sliding window size as mentioned in Section 2.1 also on the Fisher validation set.

3.4. Results

We tune for the sliding window size on our Fisher validation subset and plot the WDER with the corresponding values as shown in Figure 2. From the plot, we see that WDER decreases as the window size increases up to 30 due to increased lexical context for the backbone LM as well as the corrector model. The WDER further increases beyond the window size of 30 likely due to the corrector model being trained with an average

Table 1: *WDER of different models on Fisher test, RT03-CTS and CHAE dev, test sets. CHAE-109 is evaluated with and without fixing the number of speakers in 1st pass to 2. SimSEC: SEC model trained using simulated transcript errors, RealSEC: SEC trained/tuned on real paired data*

Model Type	Fisher Test	RT03-CTS		CHAE		CH-109	
		Validation	Test	Validation	Test	Known Spkrs	Unknown Spkrs
Baseline (No Correction)	2.26	2.30	2.18	4.23	2.82	3.69	4.28
SimSEC_v1 (Base Roberta)	1.72	2.18	1.98	4.16	2.68	3.41	4.29
SimSEC_v2 (Tuned Roberta)	1.63	1.90	1.67	3.52	2.49	3.16	3.74
RealSEC (flat-start Training)	1.53	1.73	1.58	3.31	2.30	2.98	3.57
SimSEC_v2 init + RealSEC Tuning	1.53	1.73	1.59	3.28	2.26	2.97	3.56

Table 2: *WDER of models with different amounts of Synthetic/Real Data on Fisher test set.*

Model Type	Fraction of Train Data	WDER
Baseline (No Correction)		2.26
SimSEC_v2	0.2	1.73
	0.4	1.68
	0.8	1.65
	1.0	1.63
SimSEC_v2 init + RealSEC Tuned	0.2	1.58
	0.4	1.57
	0.8	1.54
	1.0	1.54

seq length of 30 and more sliding windows with greater than 2 speakers being bypassed with a larger window size. We have also tried training with larger average sequence lengths but that did not show any additional gains compared to the sequence length of 30 words. So, we use the sliding window size as 30 words for the remainder of the experiments. We also show some qualitative examples of the correction performance on the Fisher test set using the SEC model with the best sliding window size in Figure 3. Figure 3a shows that the correction model is able to effectively correct errors due to overlapping speech when the SD hypothesizes one of the overlapping speakers and ASR hypothesizes the words of the other speaker. The model is also effective in correcting the lexically implausible errors around speaker turns which is one of the major error-prone scenario [34] for SD systems as seen in Figure 3b.

The quantitative WDER improvements of the correction models on the held out validation and test sets are outlined in Table 1. We call the model trained on ground truth transcripts with simulated speaker, ASR errors as the "SimSEC" model. SimSEC_v2 is the "SimSEC" model with a Fisher tuned backbone Roberta and trained with a custom curriculum as mentioned in Section 3.3 SimSEC_v1 is similar to SimSEC_v2 but with the Roberta-base as a backbone without any further fine-tuning. We evaluate SimSEC_v1 to quantify the gains attributed to fine-tuning of the backbone LM on conversational datasets. "SimSEC init + RealSEC Tuning" model is the paired data tuned model initialized with SimSEC_v2 and tuned using the 1st pass acoustic SD labels instead of the simulated speaker errors. RealSEC model is similar to "SimSEC_v2 init + RealSEC Tuning" but is only trained by flat-starting the model on real paired data.

From Table 1, we can see that almost all of the corrector models produce considerable WDER gains over the Baseline SD-ASR reconciled system across all the datasets, except from SimSEC_v1 on CH-109 with unknown speakers. It can be observed that tuning the backbone Roberta LM in SimSEC_v2 can produce moderate WDER gains over the pretrained Roberta-base LM, especially on CHAE validation set and CH-109 with unknown speakers. The model trained on Paired data, either by tuning the SimSEC_v2 model or by flat-start training (RealSEC) produces further gains over the models train with errors simulated (SimSEC_v1 and SimSEC_v2). The performance improvement of the models on CH-109 without fixing the speakers to 2 in the Clustering phase is comparatively limited due to hypothesising more than 2 speakers on few of the audio files leading to smaller average WDER gains. With the best model "SimSEC_v2 init + RealSEC Tuning", we observe relative WDER gains in the range 15-30% across all the datasets.

To further analyze the importance of dataset sizes needed to train or tune the models, we perform an ablation by only using a fraction of the Fisher train data as shown in Table2. We evaluate The models SimSEC_v2 and "SimSEC init + RealSEC Tuning" with different fractions of ground truth text and paired data respectively. We see that the WDER of the SimSEC_v2 model and "SimSEC init + RealSEC Tuning" model only improves moderately and saturates at a point as the amount of text data and paired data increases respectively. This shows that the corrector model can be trained purely on small amounts of ground truth transcripts by simulating speaker, ASR errors and can also be fine-tuned on a small amount of paired data to achieve significant WDER gains.

4. Conclusion and Future Work

In this work, we propose a novel Speaker Error Corrector (SEC) to correct word-level speaker label errors from a conventional audio-only speaker diarization system. We achieve this using a language model over the ASR transcriptions to correct the speaker labels. The proposed lexical SEC can be trained effectively using only text data by simulating speaker errors without the need for any paired audio-text data. A small amount of paired data can further improve model performance, leading to overall relative reduction of WDER by over 15% across three telephony datasets. The proposed SEC framework is also lightweight and is easy to integrate as a post-processing module over existing systems.

One limitation of our current work is that it has been applied only to conversations in English. One future work can include training a multi-lingual SEC to make the system language-agnostic. To increase the robustness of this approach, in addition to the first-pass SD labels, we can leverage additional complementary acoustic cues to further improve the performance. Also, the current SEC model can only handle 2 speakers in a sliding window, which we plan to generalize to handle more number of speakers. We will also explore leveraging large generative models to synthesize conversational transcripts across multiple domains using curated prompts [35].

5. References

[1] 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," *Computer Speech & Language*, vol. 72, p. 101317, 2022.

[2] Q. Wang, C. Downey, *et al.*, "Speaker diarization with lstm," in *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5239–5243, IEEE, 2018.

[3] A. Zhang, Q. Wang, Z. Zhu, J. Paisley, and C. Wang, "Fully supervised speaker diarization," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6301–6305, IEEE, 2019.

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

[5] R. Yin, H. Bredin, and C. Barras, "Neural speech turn segmentation and affinity propagation for speaker diarization," in *Annual Conference of the International Speech Communication Association*, 2018.

[6] 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.

[7] W. Xia, H. Lu, Q. Wang, A. Tripathi, Y. Huang, I. L. Moreno, and H. Sak, "Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection," in *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 8077–8081, IEEE, 2022.

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

[9] D. Snyder, D. Garcia-Romero, G. Sell, D. Povey, and S. Khudanpur, "X-vectors: Robust dnn embeddings for speaker recognition," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pp. 5329–5333, IEEE, 2018.

[10] N. Dawalatabad, M. Ravanelli, F. Grondin, J. Thienpondt, B. Desplanques, and H. Na, "Ecapa-tdnn embeddings for speaker diarization," *arXiv preprint arXiv:2104.01466*, 2021.

[11] T. J. Park, K. J. Han, J. Huang, X. He, B. Zhou, P. Georgiou, and S. Narayanan, "Speaker diarization with lexical information," *arXiv preprint arXiv:2004.06756*, 2020.

[12] L. E. Shafey, H. Soltau, and I. Shafran, "Joint speech recognition and speaker diarization via sequence transduction," *arXiv preprint arXiv:1907.05337*, 2019.

[13] M. India, J. Hernando, and J. A. Fonollosa, "Language modelling for speaker diarization in telephonic interviews," *Computer Speech & Language*, vol. 78, p. 101441, 2023.

[14] N. Flemotomos, P. Georgiou, and S. Narayanan, "Linguistically aided speaker diarization using speaker role information," *arXiv preprint arXiv:1911.07994*, 2019.

[15] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "Bert: Pre-training of deep bidirectional transformers for language understanding," *arXiv preprint arXiv:1810.04805*, 2018.

[16] Y. Liu, M. Ott, N. Goyal, J. Du, M. Joshi, D. Chen, O. Levy, M. Lewis, L. Zettlemoyer, and V. Stoyanov, "Roberta: A robustly optimized bert pretraining approach," *arXiv preprint arXiv:1907.11692*, 2019.

[17] Z. Yang, Z. Dai, Y. Yang, J. Carbonell, R. R. Salakhutdinov, and Q. V. Le, "Xlnet: Generalized autoregressive pretraining for language understanding," *Advances in neural information processing systems*, vol. 32, 2019.

[18] T. Brown, B. Mann, N. Ryder, M. Subbiah, J. D. Kaplan, P. Dhariwal, A. Neelakantan, P. Shyam, G. Sastry, A. Askell, *et al.*, "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[19] Y. Bengio, J. Louradour, R. Collobert, and J. Weston, "Curriculum learning," in *Proceedings of the 26th annual international conference on machine learning*, pp. 41–48, 2009.

[20] C. Cieri *et al.*, "Fisher english training speech part 1 speech ldc2004s13," *Web Download. Philadelphia: Linguistic Data Consortium*, 2004.

[21] C. Cieri *et al.*, "Fisher english training part 2, speech ldc2005s13," *Web Download. Philadelphia: Linguistic Data Consortium*, 2005.

[22] Q. Wang, Y. Huang, H. Lu, G. Zhao, and I. L. Moreno, "Highly efficient real-time streaming and fully on-device speaker diarization with multi-stage clustering," *arXiv preprint arXiv:2210.13690*, 2022.

[23] D. G. Canavan, Alexandra and G. Zipperlen, "Callhome american english speech ldc97s42," *Web Download. Philadelphia: Linguistic Data Consortium*, 1997.

[24] J. G. Fiscus *et al.*, "2003 nist rich transcription evaluation data ldc2007s10," *Web Download. Philadelphia: Linguistic Data Consortium*, 2007.

[25] P. Cyrta, T. Trzciński, and W. Stokowiec, "Speaker diarization using deep recurrent convolutional neural networks for speaker embeddings," in *Information Systems Architecture and Technology: Proceedings of 38th International Conference on Information Systems Architecture and Technology–ISAT 2017: Part I*, pp. 107–117, Springer, 2017.

[26] J. G. Fiscus, J. Ajot, N. Radde, C. Laprun, *et al.*, "Multiple dimension levenshtein edit distance calculations for evaluating automatic speech recognition systems during simultaneous speech.," in *LREC*, pp. 803–808, Citeseer, 2006.

[27] J. S. Chung, J. Huh, S. Mun, M. Lee, H. S. Heo, S. Choe, C. Ham, S. Jung, B.-J. Lee, and I. Han, "In defence of metric learning for speaker recognition," *arXiv preprint arXiv:2003.11982*, 2020.

[28] Y. Higuchi, M. Suzuki, and G. Kurata, "Speaker embeddings incorporating acoustic conditions for diarization," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7129–7133, IEEE, 2020.

[29] W. Zhou, W. Michel, K. Irie, M. Kitza, R. Schlüter, and H. Ney, "The rwth asr system for ted-lium release 2: Improving hybrid hmm with specaugment," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 7839–7843, IEEE, 2020.

[30] Y. Wang, A. Mohamed, D. Le, C. Liu, A. Xiao, J. Mahadeokar, H. Huang, A. Tjandra, X. Zhang, F. Zhang, C. Fuegen, G. Zweig, and M. L. Seltzer, "Transformer-based acoustic modeling for hybrid speech recognition," in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 6874–6878, IEEE, 2020.

[31] Y. Yang, P. Wang, and D. Wang, "A conformer based acoustic model for robust automatic speech recognition," *arXiv preprint arXiv:2203.00725*, 2022.

[32] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *arXiv preprint arXiv:2005.08100*, 2020.

[33] S. Majumdar and B. Ginsburg, "Matchboxnet: 1d time-channel separable convolutional neural network architecture for speech commands recognition," *arXiv preprint arXiv:2004.08531*, 2020.

[34] M. T. Knox, N. Mirghafori, and G. Friedland, "Where did i go wrong?: Identifying troublesome segments for speaker diarization systems," in *Thirteenth Annual Conference of the International Speech Communication Association*, 2012.

[35] M. Chen, A. Papangelis, C. Tao, S. Kim, A. Rosenbaum, Y. Liu, Z. Yu, and D. Hakkani-Tur, "Places: Prompting language models for social conversation synthesis," in *Findings of the Association for Computational Linguistics: EACL 2023*, p. to appear, 2023.