ENHANCING SPEAKER DIARIZATION WITH LARGE LANGUAGE MODELS: A CONTEXTUAL BEAM SEARCH APPROACH

Tae Jin Park, Kunal Dhawan, Nithin Koluguri, Jagadeesh Balam

NVIDIA, Santa Clara, USA

ABSTRACT

Large language models (LLMs) have shown great promise for capturing contextual information in natural language processing tasks. We propose a novel approach to speaker diarization that incorporates the prowess of LLMs to exploit contextual cues in human dialogues. Our method builds upon an acoustic-based speaker diarization system by adding lexical information from an LLM in the inference stage. We model the multi-modal decoding process probabilistically and perform joint acoustic and lexical beam search to incorporate cues from both modalities: audio and text. Our experiments demonstrate that infusing lexical knowledge from the LLM into an acoustics-only diarization system improves overall speaker-attributed word error rate (SA-WER). The experimental results show that LLMs can provide complementary information to acoustic models for the speaker diarization task via proposed beam search decoding approach showing up to 39.8% relative delta-SA-WER improvement from the baseline system. Thus, we substantiate that the proposed technique is able to exploit contextual information that is inaccessible to acoustics-only systems which is represented by speaker embeddings. In addition, these findings point to the potential of using LLMs to improve speaker diarization and other speech processing tasks by capturing semantic and contextual cues.

Index Terms— Speaker Diarization, Multi-speaker Speech Recognition, Large Language Model, Beam Search Decoding

1. INTRODUCTION

Multi-speaker speech recognition has been often approached by applying speaker diarization system on the input audio and feeding the speaker homogeneous segments [1] to automatic speech recognition (ASR) systems [2, 3] or speaker diarization and speech recognition are done simultaneously [4, 5]. As we can observe from the most popular previous studies [1], lexical information is only infused to improve single speaker ASR based on beam search decoding [6, 7] which reduces word error rate (WER). In general, the probability of the next word or next token is calculated by n-gram or neural language model (LM) trained on ample amount of data and the probability of the next token or word is added to the probability of the token from the acoustic model to integrate the lexical cue from the trained language model and acoustic model. This type of beam search technique can be applied to ASR models trained using connectionist temporal classification (CTC) loss [8] or recurrent neural network transducer (RNN-T) [9].

Despite the efficiency of the end-to-end ASR models which utilize an RNN-T architecture that integrates an internal LM, there are still significant benefits to be gained by these ASR models from incorporating an external language model. This improvement is mainly attributable to the disparities in the scale of training data available for acoustic and lexical modalities. Specifically, datasets for training end-to-end ASR models are limited to those containing both audio and its corresponding transcript while language models
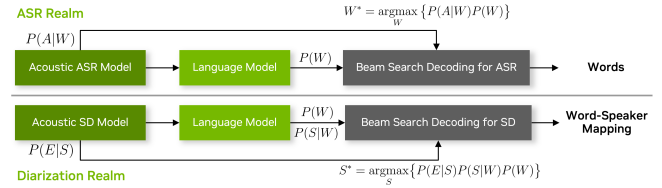

Fig. 1: The concepts of beam search decoding in the context of ASR and Speaker Diarization (SD).

can leverage text-only datasets, which are considerably larger and more diverse. In a similar vein, when it comes to speaker diarization, the volume of text data available is orders of magnitude greater than the volume of speaker-annotated audio data, especially when measured in terms of word count. Consequently, there is a potential for improvement by integrating the language models to enhance the performance of the speaker diarization task.

The aim of this paper is to introduce an application of language models to the realm of speaker diarization, demonstrating the benefit of language model trained on large amount of text-only datasets. Fig. 1 provides a comparative overview of LM applications in ASR realm and speaker diarization realm. We refer to our proposed technique as *contextual beam search*, as it seeks the most probable word-speaker mapping S by considering context from both modalities.

The utilization of lexical cues to speaker diarization, speaker turn detection and segmentation has been investigated for a long time yet remains less popular than acoustic-only speaker diarization research. Some of the earliest studies on this topic include the systems presented in [10, 11], which leveraged linguistic patterns to identify speakers during the diarization process. Numerous studies have improved speaker segmentation or clustering accuracy by integrating ASR output to leverage lexical cues [12–14]. Additionally, by merging speaker turn probabilities based on audio and text during the clustering phase [15], lexical cues are further infused into speaker diarization results. Conversely, ASR and speaker diarization results have been jointly optimized to harness the lexical cues from ASR word outputs [4, 5]. More recently, the study presented in [16] introduced semantic information through neural embeddings generated by a spoken language processing (SLP) unit. Subsequently, a multi-modal (audio-text) speaker change detector was proposed [17], along with the introduction of a speaker error correction (SEC) system [18] based on a pre-trained language model.

Our proposed method has the following distinction against the previous studies. Firstly, our approach leverages a general-purpose LLM that is trainable on text-only datasets. This effectively addresses the data-sparsity challenge commonly associated with speaker diarization. In contrast, the systems proposed in the aforementioned studies [4, 13–18] employ neural layers, such as RNN-T or transformer architectures to produce the final speaker logits and thus necessitate training on paired audio-text datasets, which our method circumvents. Secondly, our approach is not constrained by
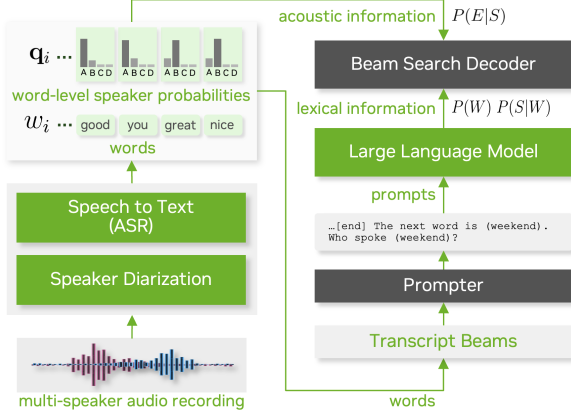
Fig. 2: Dataflow diagram of the proposed system.

the number of speakers in a speaker diarization module as the decoding process does not rely on logits from a neural network layer. This is another advantage that systems in [4, 14] cannot offer, as they are limited to a fixed number of speakers. Lastly, our approach functions similarly to general-purpose LMs work for end-to-end ASR models where an arbitrary LLM can be plugged in to improve the performance of the acoustic-only speaker diarization model. This offers significant advantages in various practical scenarios, especially when there is a need for modifying only the ASR model or the LM. For instance, when deploying the model for a different language, we can simply substitute ASR models and LLMs while using the same acoustic-only diarization model.

2. PROBABLISTIC MODELING FOR BEAM SEARCH

2.1. Probablistic Formulations of ASR and Speaker Diarization

In ASR frameworks, the task of converting speech-to-text (STT) revolves around building a model that translates a sequence of acoustic observations into a corresponding sequence of words. Formally, if we let S denote the speaker identity, W the word token, and A the acoustic observation, the STT task can be mathematically represented as estimating the most likely word sequence W given an acoustic observation A. This probability can be denoted as $P(W|A)$. Using Bayes' theorem, this can be represented as:

$$W^* = \operatorname*{argmax}_{W} \big\{ P(W|A) \big\} \tag{1}$$

$$= \operatorname*{argmax}_{W} \left\{ \frac{P(A|W)P(W)}{P(A)} \right\} \tag{2}$$

$$= \operatorname*{argmax}_{W} \big\{ P(A|W)P(W) \big\}, \tag{3}$$

where W is a word or a token and A is an acoustic observation. Expanding this idea to the realm of speaker diarization, our goal is to estimate the speaker label S given both the acoustic observation E and word W. Formally, we can express this as:

$$S^* = \operatorname*{argmax}_{S} \{ P(S|E, W) \} \tag{4}$$

$$= \operatorname*{argmax}_{S} \left\{ \frac{P(E, W|S)P(S)}{P(E, W)} \right\} \tag{5}$$

$$= \operatorname*{argmax}_{S} \{ P(E, W|S)P(S) \}. \tag{6}$$

For the sake of simplifying our computations and model, we make an assumption of conditional independence between the acoustic observation E and word W, given the speaker identity S. This assumption

is mathematically represented as:

$$P(E, W|S) \stackrel{\text{C.I.}}{=} P(E|S)P(W|S). \tag{7}$$

From this conditional independence assumption, we can restructure Eq. (4) to remove the unrelated term $P(W)$, leading to the essential expressions that require computation. The derivation is presented as:

$$S^* \stackrel{\text{C.I.}}{=} \operatorname*{argmax}_{S} \big\{ P(E|S)P(W|S)P(S) \big\} \tag{8}$$

$$= \operatorname*{argmax}_{S} \big\{ P(E|S)P(S|W)P(W) \big\}. \tag{9}$$

In alignment with the ASR framework illustrated in Fig. 1, where $P(A|W)$ is represented as acoustic model, we utilize the acoustic-only diarization model to represent $P(E|S)$. Additionally, we derive $P(S|W)$ using pre-trained general-purpose language models, such as n-gram language models and LLMs. It is crucial to differentiate between $P(W)$ in Eq. (6) and $P(W)$ in Eq. (3) because the value of $P(W)$ in Eq. (6) is contingent upon the condition that speakers are assigned to each word in the preceding word sequence. Let w_i denote i-th word: each i index corresponds to a specific word and its assigned speaker identity k, as illustrated in Fig. 2. The given relationship can be expressed as:

$$\mathbf{W_S}^{C-1} = \{ (w_1, \mathbf{q}_1), (w_2, \mathbf{q}_2), \cdots, (w_{C-1}, \mathbf{q}_{C-1}) \}, \tag{10}$$

where C denotes the word sequence length (*i.e.*, context length) and q_i is the i-th speaker probability vector generated by the speaker diarization model. The matrix $\mathbf{S}$ encompasses a sequence of speaker probability vectors $\mathbf{q}$, represented as $\mathbf{S} = [\mathbf{q}_1, \mathbf{q}_2, \ldots, \mathbf{q}_{C-1}]$. With the notation $\mathbf{W_S}^{C-1}$, the probability of a word given the past transcription $P(W)$ is expressed as:

$$P(W) = P(w_i | \mathbf{W_S}^{C-1}). \tag{11}$$

2.2. Acoustic Inference: Speaker Diarization

We employ an improved version of Multi-scale Diarization Decoder (MSDD) model [19], which is introduced in [20]. Given a maximum speaker limit N_S, the diarization output is manifested by N_S-dimensional floating point numbers, each representing the probability associated with each frame of 0.05 seconds in length. We utilize time-stamps derived from an ASR model to sample the diarization these diarization logit values. The corresponding speaker probability for k-th speaker can be described as

$$q_k = P(E|S)|_{S=k} = \frac{\sum_{t=1}^{T} p(S = k|E, t)}{\sum_{k}^{N_S} \sum_{t=1}^{T} p(S = k|E, t)}, \tag{12}$$

where $P(S|E, t)$ denotes the sigmoid logit value at time t and T denotes the number of diarization logit frames within the word. Consequently, we obtain an N_S-dimensional floating point probability vector $\mathbf{q}$ that sums up to one. As a result, we convert sigmoid values to probability values as we treat $P(E|S)$ as a probability measure. Note that any type of diarization or speaker recognition system can be employed to determine the speaker probability $P(E|S)$ for the sequence of $\mathbf{q}_i$ values, where i denotes word index. After the diarization and ASR processes, or any other transcription and speaker recognition methods, we obtain a word sequence, $\mathbf{w}$ and corresponding speaker probability values per word denoted as $\mathbf{q}_i = [q_1, q_2, \ldots, q_k]_i$, as illustrated in Fig. 2.

2.3. Scoring for Beam Search Decoding

Our proposed beam search decoding approach is based on Eq. (9). Let β represent the mixing coefficient between acoustics-only speaker diarization and the language model, and let α represent
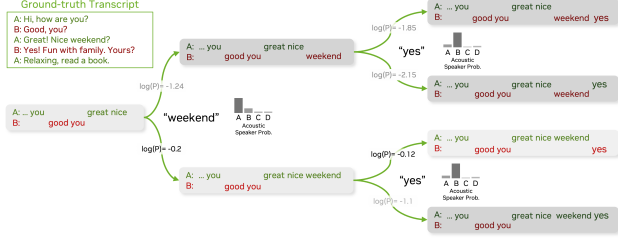
Fig. 3: Illustration of beam search for a two-speaker dialogue.

the scaling parameter for $P(W)$. The beam search score function can then be formulated as:

$$P_{BSD}(S) = \log\big(P(E|S)\big) + \beta \log\big(P(S|W)P(W)^\alpha\big). \quad (13)$$

We employ a modified version of the beam search decoder from [21] which is capable of applying Eq. (13) as a score for beam search. Fig. 3 provides a visual representation of the beam search decoding process. The calculations of $P(E|S)$ and $P(S|W)$ will be covered in section 3. The role of the term $P(W)$ is crucial. Since the sum of $P(S|W)$ for all speakers S equals 1, if all speakers have an equal probability in a lexical sense, $P(S|W)$ becomes $1/N_S$. For instance, if we consider cases where multiple speakers utter the same filler words, we are likely to see $P(S|W)$ approaches $1/N_S$ as there is not significant lexical context to distinguish one's speaker identity from another. The term $P(W)$ addresses these situations by assigning a relatively low probability, compensating for the uncertain lexical cue. Hence, the beam search predominantly relies on $P(E|S)$ to select the most probable speaker for the corresponding word.

While $P(W)$ can serve as a confidence parameter for lexical context, the degree of compensation can be controlled by the parameter α. From a mathematical perspective, if α is close to 0, we give less importance on the confidence of the language model and as α increases, we further suppress the lexical context, proportional to the word probability.

3. LEXICAL INFERENCE

We employ two types of language model for comparison: the KenLM [22] based n-gram language model and the GPT-based LLM from the Megatron-LM framework in the NeMo toolkit [23, 24].

3.1. Baseline n-gram language model

3.1.1. Speaker probability from n-gram

We use a 4-gram language model [25] that is publicly available and has approximately 8M probability value entries. The following example demonstrates speaker-wise transcript and the *next word* that should be appended to a speaker's last word.

```
[Speaker-0] <s> how are you doing these days</s><s> well tell
me more</s>
[Speaker-1] <s> things are going very well</s><s> there is a
project that i'm
[All-Speakers] <s> how are you doing these days</s><s> things
are going very well</s><s>well tell me more</s><s> there
is a project that i'm
[Next Word] working
```

For the n-gram language model, we use the start-of-sentence (SOS) token, `<s>` and the end-of-sentence (EOS) token, `</s>`. These tokens can significantly influence the probability of the n-gram score for a given word sequence. The following equation describes the probability of having w_{next} given that all C words assigned with the particular speakers in S:

$$P_S(w_i = w_{\text{next}}) = P(w_{S,i}|\mathbf{W}_S^{C-1}) \quad (14)$$
$$= P(S, W), \quad (15)$$

where $\mathbf{W}_S^{C-1}$ is a word sequence $w_1, \ldots, w_n$ for the given speaker

S, and $C-1$ represents the length of the context window for speaker k. Therefore, the probability for the k-th speaker among N_S speakers is denoted by the following equation:

$$P(S|W)|_{S=k} = \frac{P(S, W)}{P(W)}\bigg|_{S=k} = \frac{P(w_{k,i}|\mathbf{W}_k^{C-1})}{\sum_{k=1}^{N} P(w_{k,i}|\mathbf{W}_k^{C-1})}. \quad (16)$$

3.1.2. Word Probability

As we discussed in the previous section, we calculate the $P(W)$ term separately. Note that this probability differs from $P(W)$ in Eq. (16) since we use the entire word sequence, which includes all speakers, to calculate the $P(W)$ value using the following equation:

$$P(W) = \sum_{k=1}^{N_S} P(w_{k,i}|\mathbf{W}_k^{L-1})|_{w_{k,i}=w_{k,\text{next}}} \quad (17)$$
$$= P(w_{\text{next}}|\mathbf{W}_S^{L-1}), \quad (18)$$

where L is the length of the context that includes all N-speakers.

3.2. Speaker Diarization Prompt for LLM

While the baseline n-gram language model works for the purpose of estimating the most probable speaker, a neural language model trained on large amount of data can parse more context in the given text-based conversation. This allows it to estimate a more accurate speaker probability based on lexical cues.

3.2.1. Speaker probability from LLM

In the LLM-based calculation of $P(S|W)$, we rely on a prompt that asks the expected speaker the expected speaker to provide the subsequent word. An illustrative example of such a prompt for the subsequent word `working` is provided below.

```
[Speaker0]:how are you doing these days
[Speaker1]:things are going very well
[Speaker0]:well tell me more
[Speaker1]:there is a project that i'm
[end]
Question: The next word is (working). Who spoke (working)?
Answer:[Speaker1]
```

Using the provided prompt template, we simulate the probability $P(S, W)$ by sampling the probability values from the token indicating the speaker index that follows the token labeled as `speaker`. The prompt includes the text leading up to `speaker`. From this, we can derive $P(S|W)$ using the subsequent equation:

$$P(S|W)|_{S=k} = \frac{p_k}{\sum_{k=1}^{N_S} p_k}, \quad (19)$$

where p_k denotes the simulated probability calculated from the logit values in the LLM output for speaker index k. It is worth noting that the n-gram approach inherently cannot account for interactions between two speakers when determining speaker probability. In contrast, the LLM approach, the LLM approach allows for the consideration of interplay and combined information between speakers when estimating the speaker probability $P(S|W)$.

3.2.2. Word Probability

In case of calculating $P(W)$, we use a specific input to sample the probability of the last word. We employ the same dialogue prompt as used for the speaker probability. However, we insert the term *next word* and remove any text following the `[end]` token. With this prompt, the word probability can be expressed as:

$$P(W)|_{W=w_{\text{next}}} = \frac{p_d}{\sum_{d=1}^{D} p_d}, \quad (20)$$

where d and D represent the token index and the total number of tokens, respectively.

Language Model (LM)	(LM for $P(S\|W)$, LM for $P(W)$)									ASR	Speaker Diarization			
	-		All n-gram		(LLM, n-gram)		(n-gram, LLM)		All LLM						
Matching	TS-match		BSD		BSD		BSD		BSD						
Metric	ΔSA	Δcp	ΔSA	Δcp	ΔSA	Δcp	ΔSA	Δcp	ΔSA	Δcp	WER	Miss	FA	CER	DER
CH-others	8.45	8.28	8.07	7.86	8.36	8.18	8.61	8.41	**7.76**	**7.58**	25.15	5.97	9.79	6.04	21.80
CH-109	5.05	4.97	3.70	3.64	3.70	3.63	4.42	4.30	**3.24**	**3.16**	23.07	5.55	4.39	1.44	11.38
AMI-MH-dev	8.22	8.20	3.47	3.45	3.49	3.47	3.52	3.50	**3.45**	**3.42**	23.68	15.89	4.02	1.02	15.89
AMI-MH-test	6.38	6.41	3.92	3.88	**3.74**	**3.72**	3.98	3.94	3.84	3.80	23.25	12.96	4.98	1.36	19.03

Table 1: Performance evaluation of multi-speaker ASR based on time-stamp (TS) matching and beam search decoding (BSD).

4. EXPERIMENTAL RESULTS

4.1. Experiment Settings

4.1.1. Pre-trained Models

- **ASR:** We employ a Conformer-CTC model [26] that is implemented using the NeMo Toolkit [24]. The ASR model has approximately 122M parameters and 1024 tokens.
- **Speaker Diarization:** We employ an improved version [20] of the MSDD model [19], which has 32M parameters. It builds upon Titanet [27], Transformer Encoder [28], and Clustering [29].
- **LLM**[1]**:** We use an LLM based on [23] which is a scaled version of GPT [31] model of 2B parameter size. Trained on a 1.1T token subset of the dataset curated by NVIDIA Data Collector (NDC) [30], containing 70% of English dataset, 15% of Stack v1.1 dataset [32] and 15% of multi-lingual datasets from Common Crawl [33].

4.1.2. Evaluation Metric

- **WER**: The WER is determined using a hypothesis script generated from a single-channel mixed audio clip and a transcript that contains words in onset order. Note that this WER should be differentiated from the channel-specific or speaker-specific WER in other studies. We use text normalization tool from [34] for evaluation.
- **SA-WER**: The Speaker-Attributed WER (SA-WER) [3] is a WER metric grounded in speaker mapping, as established by speaker diarization. SA-WER measures the WER by comparing hypothesis and reference transcripts for each specific speaker mapping.
- **cpWER**: The concatenated minimum-permutation word error rate (cpWER) [35] is a metric designed to capture both ASR and diarization accuracy. cpWER is calculated by taking the minimum WER from concatenated transcripts of multiple speakers across all potential permutations.
- **Δcp** and **ΔSA**: delta-cpWER and delta-SA-WER follows the following relationship.

$$\Delta cp = cpWER - WER$$
$$\Delta SA = SA\text{-}WER - WER$$

Our evaluation is based on the assumption that Δcp and ΔSA reflect the diarization error in cpWER and SA-WER, respectively.

4.2. Datasets and Systems Overview

Table 1 shows the performance for each setup and dataset. For AMI-MH (Mixed Headset) [36] dataset, we use only-word version [37] of annotation. The Call Home American English Speech (CHAES, LDC97S42) is a corpus composed solely of English telephonic speech data. CH-109 is a subset that includes two speakers per session, whereas CH-others signifies the other sessions found within CHAES. We optimize α, β in Eq. (13) and context window length C in Eq. (10), beam-width for beam search decoding on CH-others and AMI-MH-dev then use the parameters for CH-109 and AMI-MH-test, respectively. For this parameter optimization, we employ Optuna [38]. The Diarization Error Rate (DER) is computed using a collar of 0.25 seconds while including overlaps.

In terms of systems, *TS-match* is a system that relies on time-stamps (TS) to match speaker diarization time-stamps with the decoded word time-stamps from ASR system. *All n-gram* and *All LLM* are systems where speaker probability and word probability values are calculated from the n-gram LM and the LLM, respectively. Conversely, *(LLM, n-gram)* and *(n-gram, LLM)* employ one model for speaker probability and another for word probability, in accordance with the provided notation. All experiments involving the LLM were conducted using the NVIDIA TESLA V100 GPU.

4.3. Analysis of Experimental Results

In Table 1, the performance of TS-matching is compared with four combinartory setups that integrate both n-gram and LLM. Regarding Δcp and ΔSA, it is crucial to highlight that speaker confusion resulting from speaker diarization results in a double error count because it manifests as an insertion for one speaker and a deletion for another. The proposed method improves the baseline system's delta-SA-WER by up to 39.8%. We can deduce couple of findings: Firstly, the trend shows that applying same type of language model leads to a lower error rate. This is likely because the $P(W)$ and $P(S|W)$ are less prone to discrepancies. Secondly, LLM seems more effective in estimating $P(S|W)$, given that the average performance of applying LLM for $P(S|W)$ demonstrates superior performance over n-gram. We speculate that the performance difference stems from the fact that the n-gram model considers only a single speaker, while LLM processes the entire transcription, taking into account all speakers within the context window. This contextual understanding is a distinct advantage LLM holds over n-gram LM, providing a more nuanced estimation of the speaker. However, using LLM demands roughly 15 times more computational time during inference when compared to the n-gram LM. This discrepancy underscores the potential need for a trade-off, suggesting a hybrid *(LLM, n-gram)* configuration.

5. CONCLUSIONS

In this paper, we introduce a beam search decoding-based approach for applying a language model to speaker diarization. The proposed method offers a key advantage: it uses individually optimized models for ASR, diarization, and LLM. By training each model independently, we can leverage large-scale data sources specific to each domain. Moreover, even without fine-tuning the ASR, diarization, and LLM models, the proposed method achieves significant improvement over conventional approaches. This methodology can be seamlessly adapted to accommodate multilingual contexts by integrating multilingual ASR and LLM models—areas that have seen significant advancements recently. Looking forward, our future research will focus on several areas: Firstly, we intend to integrate beam search decoding for ASR and diarization by applying a single LLM which can achieve a more efficient system with enhanced accuracy. Secondly, we plan to integrate the ASR and diarization decoders to obtain more accurate timestamps alongside speaker logits, aiming for a streamlined multi-speaker ASR. Finally, we will explore ways to improve the model by introducing more sophisticated context, either by fine-tuning or prompt-tuning the LLM using domain-specific data.

---

[1]Details regarding model training and datasets will be provided in [30]

6. REFERENCES

[1] Tae Jin Park et al., "A review of speaker diarization: Recent advances with deep learning," *Computer Speech and Language*, vol. 72, pp. 101317, 2022.

[2] Ivan Medennikov et al., "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario," in *Proc. Interspeech*. ISCA, 2020, pp. 274–278.

[3] Samuele Cornell et al., "The CHiME-7 DASR Challenge: Distant meeting transcription with multiple devices in diverse scenarios," *CHiME-7 Workshop*, 2023.

[4] Laurent El Shafey et al., "Joint speech recognition and speaker diarization via sequence transduction," in *Proc. Interspeech*. ISCA, 2019, pp. 396–400.

[5] Naoyuki Kanda et al., "Transcribe-to-diarize: Neural speaker diarization for unlimited number of speakers using end-to-end speaker-attributed asr," in *Proc. ICASSP*. IEEE, 2022, pp. 8082–8086.

[6] Hermann Ney et al., "A data-driven organization of the dynamic programming beam search for continuous speech recognition," in *Proc. ICASSP*. IEEE, 1987, vol. 12, pp. 833–836.

[7] Harald Scheidl et al., "Word beam search: A connectionist temporal classification decoding algorithm," in *Proc. ICFHR*. IEEE, 2018, pp. 253–258.

[8] Alex Graves et al., "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proc. ICML*, 2006, pp. 369–376.

[9] Alex Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[10] Leonardo Canseco-Rodriguez, Lori Lamel, and Jean-Luc Gauvain, "Speaker diarization from speech transcripts," in *Proc. ICSLP*, 2004, vol. 4, pp. 3–7.

[11] Leonardo Canseco, Lori Lamel, and J-L Gauvain, "A comparative study using manual and automatic transcriptions for diarization," in *Proc. ASRU*. IEEE, 2005, pp. 415–419.

[12] Tae Jin Park and Panayiotis Georgiou, "Multimodal speaker segmentation and diarization using lexical and acoustic cues via sequence to sequence neural networks," in *Proc. Interspeech*. ISCA, 2018, pp. 1373–1377.

[13] Wei Xia et al., "Turn-to-diarize: Online speaker diarization constrained by transformer transducer speaker turn detection," in *Proc. ICASSP*. IEEE, 2022, pp. 8077–8081.

[14] Aparna Khare et al., "Asr-aware end-to-end neural diarization," in *Proc. ICASSP*. IEEE, 2022, pp. 8092–8096.

[15] Tae Jin Park et al., "Speaker diarization with lexical information," in *Proc. Interspeech*. ISCA, 2019, pp. 391–395.

[16] Luyao Cheng et al., "Exploring speaker-related information in spoken language understanding for better speaker diarization," in *arXiv preprint arXiv:2305.12927*. IEEE, 2023.

[17] Jee-weon Jung et al., "Encoder-decoder multimodal speaker change detection," in *Proc. ICASSP*. IEEE, 2023.

[18] Rohit Paturi et al., "Lexical speaker error correction: Leveraging language models for speaker diarization error correction," *arXiv preprint arXiv:2306.09313*, 2023.

[19] Tae Jin Park et al., "Multi-scale speaker diarization with dynamic scale weighting," in *Proc. Interspeech*. ISCA, 2022.

[20] Tae Jin Park et al., "The CHiME-7 Challenge: System description and performance of nemo team's DASR system," *CHiME-7 Workshop*, 2023.

[21] Kensho Technologies, "PyCTCDecode: A fast and lightweight python-based CTC beam search decoder for speech recognition," 2023, [Accessed 09-06-2023].

[22] Kenneth Heafield et al., "KenLM: Faster and smaller language model queries," in *Proc. SMT Workshop*, 2011, pp. 187–197.

[23] Mohammad Shoeybi et al., "Megatron-LM: Training multi-billion parameter language models using model parallelism," *arXiv preprint arXiv:1909.08053*, 2019.

[24] Oleksii Kuchaiev et al., "NeMo: a toolkit for building ai applications using neural modules," *arXiv preprint arXiv:1909.09577*, 2019.

[25] Vassil Panayotov et al., "LibriSpeech: an asr corpus based on public domain audio books," 2015, [Accessed 09-06-2023].

[26] Anmol Gulati et al., "Conformer: Convolution-augmented transformer for speech recognition," in *Proc. Interspeech*. ISCA, 2020, pp. 5036–5040.

[27] Nithin Rao Koluguri et al., "TitaNet: Neural model for speaker representation with 1d depth-wise separable convolutions and global context," in *Proc. ICASSP*. IEEE, 2022, pp. 8102–8106.

[28] Ashish Vaswani et al., "Attention is all you need," *Proc. NIPS*, vol. 30, 2017.

[29] Tae Jin Park et al., "Auto-tuning spectral clustering for speaker diarization using normalized maximum eigengap," *IEEE SPL*, vol. 27, pp. 381–385, 2019.

[30] NVIDIA, "NeMo Megatron Launcher," https://github.com/NVIDIA/NeMo-Megatron-Launcher, 2023, [Accessed 09-06-2023].

[31] Alec Radford et al., "Language models are unsupervised multitask learners," *OpenAI blog*, vol. 1, no. 8, pp. 9, 2019.

[32] Hugging Face, "The-Stack-Dedup:Datasets at Hugging Face," https://huggingface.co/datasets/bigcode/the-stack-dedup, 2023, [Accessed 09-06-2023].

[33] Alexandra Sasha Luccioni and Joseph D Viviano, "What's in the box? a preliminary analysis of undesirable content in the common crawl corpus," *arXiv preprint arXiv:2105.02732*, 2021.

[34] OpenAI, "Whisper: Robust speech recognition via large-scale weak supervision," https://github.com/openai/whisper, 2023, [Accessed 09-06-2023].

[35] Shinji Watanabe et al., "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," *arXiv preprint arXiv:2004.09249*, 2020.

[36] Wessel Kraaij et al., "The AMI meeting corpus," in *Proc. International Conference on Methods and Techniques in Behavioral Research*, 2005.

[37] Federico Landini et al., "Bayesian HMM clustering of x-vector sequences (VBx) in speaker diarization: theory, implementation and analysis on standard tasks," *CSL*, vol. 71, pp. 101254, 2022.

[38] Takuya Akiba et al., "Optuna: A next-generation hyperparameter optimization framework," in *Proc. the 25th ACM SIGKDD*, 2019, pp. 2623–2631.