Generating Data with Text-to-Speech and Large-Language Models for Conversational Speech Recognition

Samuele Cornell[*1], *Jordan Darefsky*[*2], *Zhiyao Duan*[2], *Shinji Watanabe*[1]

[1]Carnegie Mellon University, USA
[2]University of Rochester, USA

samuele.cornell@ieee.org, jdarefsk@u.rochester.edu

Abstract

Currently, a common approach in many speech processing tasks is to leverage large scale pre-trained models by fine-tuning them on in-domain data for a particular application. Yet obtaining even a small amount of such data could be problematic especially for sensitive domains and conversational speech scenarios, due both to privacy issues and annotation costs. To address this, synthetic data generation using single speaker datasets has been employed. Yet, for multi-speaker cases, such approach often requires extensive manual effort and is prone to domain mismatches. In this work, we propose a synthetic data generation pipeline for multi-speaker conversational ASR, leveraging a large language model (LLM) for content creation and a conversational multi-speaker text-to-speech (TTS) model for speech synthesis. We conduct evaluation by fine-tuning the Whisper ASR model for telephone and distant conversational speech settings, using both in-domain data and generated synthetic data. Results show that the proposed method is able to significantly outperform classical multi-speaker generation approaches that use external non-conversational speech datasets.
Index Terms: generative synthetic data, multi-talker speech recognition, text-to-speech, conversational speech processing

1. Introduction

Current robust speech processing methods are incredibly data hungry. For example, state-of-the-art automatic speech recognition (ASR) systems require tens or even hundreds of thousands of hours of training data in order to achieve enough robustness in different domains [1–3]. Such amount of training data is leveraged either explicitly by training from scratch on a large amount of data or implicitly, by fine-tuning/adapting a pre-trained "foundation" model which in its turn was trained, in a supervised or unsupervised manner [1,4–6], on a large dataset.

Nevertheless, for some domains, even obtaining a small portion of in-domain supervised data for fine-tuning is problematic as it could raise privacy concerns and/or be significantly expensive. This is especially true for sensitive application scenarios: medical applications, government, law enforcement et cetera. Moreover, as regulations get stricter in many countries, scaling in-domain training data is becoming more difficult also in other domains.

To be fair, aside from privacy issues, application scenarios that require recordings with multiple speakers are also inherently difficult, time-consuming and, crucially costly to annotate and thus to obtain in scale. Prominent examples are meeting scenarios [7,8] including doctor-patient recordings, speech captioning, speech analytics and so on.

[*]These authors contributed equally to this work.

Instead, at the same time, there are speech processing approaches that need such multi-speaker conversational data for training. Crucially these have also been proven to be effective on such data as demonstrated in recent speech processing challenges [7–9]. Prominent examples are end-to-end neural diarization (EEND) and most target speaker voice activity detection (TS-VAD) approaches [10–14] as well as multi-speaker ASR [15–19]. Lack of annotated in-domain conversational data at scale is a significant issue for these techniques which is only partly mitigated by leveraging foundation models [17–19]. As such, many of these approaches have to rely on synthetic data to increase the training material. This is usually obtained by artificially overlapping clips from existing datasets and adding noise and reverberation.

While several toolkits to ease the workload have been proposed [20, 21], creating synthetic datasets is still more an art than a science as it often needs lots of hand-tuning, domain knowledge, heuristics and significant trial and error. This process is also highly prone to the introduction of unwanted biases in the resulting dataset, leading to a performance drop due to domain mismatch [12].

As such it is desirable to have more automated, machine learning based approaches for such synthetic data creation. And in fact, several methods have explored such research direction, mainly focusing on improving ASR performance by leveraging synthetic data created with text-to-speech (TTS) models [22–27, 27–31] or leveraging ASR and TTS cycle-consistency during training [32, 33] for semi-supervised training. However, all these approaches focused on single-speaker scenarios, making them unsuitable for the aforementioned application domains where multi-talker conversational ASR is required. In parallel, recent works [34, 35] on respectively speech summarization and audio captioning, have shown how large-language models (LLM) can be leveraged effectively for synthetic data audio augmentation.

Building upon these previous works, here we explore the intriguing possibility of using TTS models in conjunction with a LLM to generate multi-speaker conversational data. In this preliminary work we focus on 2 speakers multi-speaker ASR on real-world telephone (Fisher [36]) and distant speech recognition settings (Mixer 6 Speech [37]) by fine-tuning Whisper [1]. The contributions of this work are the following: 1) we propose a synthetic data generation pipeline for conversational ASR by using LLMs for content generation and a conversational multi-speaker TTS model for speech generation; 2) we perform a systematic investigation on the use of synthetic data for training multi-speaker ASR models obtained with different approaches: using "classical" LibriSpeech based multi-speaker simulation, using a conventional state-of-the-art (SotA) TTS model and finally using a recently proposed conversational TTS model [38].

2. Method under study

Our approach is summarized in Figure 1. We consider the use of a pre-trained chat-optimized LLM for creating short conversation transcripts between two participants from scratch for when in-domain conversational transcriptions are not available or would be costly to obtain. In detail, in this work, we use the recently released Llama 3 Instruct model and few-shot prompt it with 8 prompt examples randomly selected from a 1000 examples subset of Spotify Podcasts dataset [39] text data (same data used for training the Parakeet TTS model see following Sec. 2.1). That is, for each new example we want to generate, we randomly select a subset of eight text examples from our Parakeet subset to use as the few-shot prompt. Such procedure could also be used to augment existing in-domain text-only data in the same way and/or by fine-tuning the LLM on some of the in-domain data.

These LLM obtained transcripts are then used to generate synthesized speech through a multi-speaker TTS model. The resulting data, consisting of ground truth multi-speaker transcripts and the synthesized multi-speaker mixture can then be used for training or fine-tuning purposes, i.e. in Sec. 4 for adapting Whisper to perform multi-speaker ASR.

2.1. Conversational TTS generation

The effectiveness of such an approach will heavily depend on the capability of the TTS model used. While we expect LLMs will be proficient in generating conversational transcripts as shown in previous work on summarization [34], most TTS models are not capable of synthesizing multi-speaker conversational data. In fact, one could naively generate each speaker's utterances independently and then stitch them together, however such an approach would fail to capture real conversational speech turn taking dynamics and para-linguistic subtleties such as changes of intonation, etc., and would therefore potentially introduce a domain mismatch in the generated audio.

Recently, in [38] a conversational TTS model, Parakeet, has been proposed. Parakeet's training dataset includes 60,000 hours of Spotify Podcasts data, much of which is multi-speaker Therefore it is able to directly generate two-speaker short conversations of up to 30 seconds when given a text in the style of the one in Figure 1 i.e. with speaker-id related tags [S1] and [S2]. We use a diffusion version of Parakeet that, similarly to [40] autoregressively generate blocks of continuous latents using latent diffusion on each block. Each block consists of 128 latents. We use an autoencoder that maps 44,100 Hz audio to 16-channel dimensional latents, with a time downsampling factor of 1024.

LLM-generated transcripts and speech examples are available online[1].

3. Experimental setup

3.1. Evaluation data

As said, in this preliminary work we focus on 2 speakers multi-speaker conversational ASR. This is primarily because the Parakeet TTS model only supports 2 speakers utterances due to the training data that was used. In addition, we also consider scenarios with relative high signal-to-noise ratio (SNR). In fact, tackling more complex settings such as CHiME-6 [7] requires to also model the background noise and dynamic acoustic conditions (as the participants move, reverberation can change significantly). We thus perform our experiments using two conversational speech datasets with these characteristics: Fisher Corpus (both Part 1 and Part 2) and Mixer 6 Speech.

[1] popcornell.github.io/SynthConvASRDemo

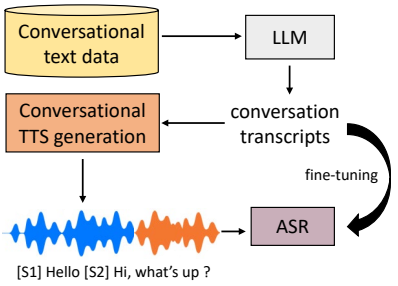

Figure 1: *Block diagram of the proposed approach.*

3.1.1. *Fisher*

Fisher comprises of 11699 telephone conversations between two English speakers sampled at 8 kHz. Each conversation is around 10 minutes long. We use the train, validation and test split from [41] (11577, 61 and 61 conversations of respectively 1960 h, 7 h and 7 h). Due to being telephone speech, it features separate channels for each of the two speakers. However, since our focus here is more general single-channel conversational speech processing, we mixdown the two to mono. We also resample the signal to 16 kHz as we use Whisper which was trained on 16 kHz data (see Sec. 3.3).

3.1.2. *Mixer 6 Speech*

As an additional scenario we consider Mixer 6 Speech and, in detail, the version re-annotated for the CHiME-7 challenge [8]. It consists of 2-speakers interviews of approximately 15 minutes (sampled at 16 kHz) recorded by 14 different far-field recording devices. For our purposes here we use only the table-top microphone device (CH04). We use the splitting from [8], where full long-form annotation is only available for the development (59 interviews, 15 h) and evaluation sets (23 interviews, 13 h). Here we further split the development set into an adaptation portion and a validation portion of respectively 2:30 h and 4 h after discarding utterance groups longer than 30 s as done in [19]. This further split allows to compare the use of synthetic data versus in-domain data for fine-tuning.

3.2. Baseline Methods

3.2.1. *NeMo multi-speaker simulation tool*

In this work we consider two baseline methods. First, a "classical" synthetic speech generation method i.e. where single speaker speech from one high quality speech dataset (e.g. LibriSpeech [42]) is used to construct conversation-style fake recordings by artificially overlapping single speaker utterances and contaminating them by adding noise, artificial room impulse response (RIR) or other transforms (e.g. clipping, microphone transfer function etc.). We make use here of the SotA NeMo multi-speaker simulation tool [21] (NeMo MSS in the following). In detail, we use LibriSpeech train-clean 360 and 100 portions and generate 100 h of short conversations between two speakers of up to 30 seconds in length. For Mixer 6 Speech experiments, we use additionally the built-in RIR simulation in order to generate simulated far-field speech.

3.2.2. xTTS-v2

The second baseline method we consider is the approach outlined in Section 2, where a standard TTS model is used to generate the training data. For the TTS model we consider Coqui xTTS-v2 model [43] (denoted simply as xTTS in Sec. 4) In detail, for each utterance group in the training dataset (either LLM-generated or taken from a text-only corpus) we sample two speaker ids from LibriSpeech train-clean 360 and 100 and then two corresponding LibriSpeech enrollment utterances to condition xTTS-v2 for the generated TTS id. Then we generate each utterance in the utterance group independently via xTTS-v2 and truncate excessive leading and trailing silence regions using Silero VAD [44]. These are then resampled to 16 kHz and mixed together by randomly adding start time offsets based on the order of the sentences in the utterance group transcript, ensuring that utterances from the same speaker do not overlap.

3.3. ASR System

As said, in this work, we focus on conversational speech recognition (CSR). For our experiments, which consider 2 speakers conversational speech, we use the method proposed in [19] where Whisper [1] was adapted to perform multi-speaker ASR by fine-tuning it with a serialized output training (SOT) [15] objective on utterance groups. This approach aligns with common practices in the field where often a model pre-trained on a large amount of data (i.e. a foundation model) is fine-tuned/adapted for a particular domain or application of interest.

Compared to [19], in our experiments we focus only on standard SOT without considering timestamps and use only Whisper medium. We use low-rank adapters (LoRA) [45] while the rest of the model is kept frozen. During each fine-tuning experiment a linear warm-up schedule is employed for the first N epoch, then the learning rate is linearly decayed till a maximum of 20 epochs. The L^2 norm of the gradients is clipped to 5. One LoRA adapter for each linear layer in the model (i.e. for each query, key, value and feed-forward network layer) is employed. For each adapter we set parameters rank to 64, alpha to 128 and dropout to 0.1. In our preliminary experiments with the full Fisher training set, we found that this configuration yielded the best results, even when compared to fine-tuning the entire model. If validation loss does not improve for 2 consecutive epochs the training is halted. We tune the batch size, amount of warm-up epochs (N) and the value of the maximum learning rate for each set of experiments. Parakeet synthesized audio is resampled to 16 kHz in our experiments and during training, for Fisher experiments, for all synthetic data, we use on-the-fly resampling to simulate telephone 3400 Hz band-limiting. Instead, for Mixer 6 experiments, only for xTTS and Parakeet, we contaminate the data with reverberation using random RIRs obtained from [46]. This of course is less realistic than the RIR simulation used in NeMo MSS as the RIR is the same for both speakers. We make our fine-tuning code publicly available[2].

3.4. Evaluation Setup

For each dataset, we run our experiments using the same setup as in [19], where oracle voice activity detection (VAD) is considered and the dataset is divided into several utterance groups [3, 19]. Again, following [19], we then perform evaluation for each utterance group independently and accumulate word error rate (WER) statistics over the whole dataset (insertions, deletions etc.). This is because in this preliminary work, as said, we only focus on multi-speaker ASR, and an evaluation which considers the whole conversation (e.g. as in CHiME-6/7) would require a diarization component. Thus contaminating and complicate the results analysis.

As figure of merits we thus consider concatenated minimum permutation WER (cpWER) [7]. This is the same as WER figure in [19], with the best permutation evaluated for each utterance group independently. We also consider multi-input multi-output WER (MIMO-WER), which, contrary to cpWER, is more tolerant to speaker assignment errors. Meeteval toolkit [47] was used to compute both. Whisper text normalization is used both during training and scoring.

4. Experiments

4.1. Fisher

In Table 1 we report results obtained on the Fisher test set as defined in Sec. 3.1.1 with different data used for fine-tuning. As a baseline, in the first row, we report the results with no adaptation in the first row while in the second panel results with in-domain Fisher training data adaptation. We can observe that the difference between using the full training set or a 80 h data subset is modest, due to the fact we are leveraging a strong pre-trained model. In the third and bottom panel we instead report results obtained with the synthetic data approaches under study. In particular, for the two TTS approaches (xTTS and Parakeet), we consider two opposite situations: a best-case/oracle scenario where we use in-domain conversation transcriptions and another one where we suppose we have none and thus we use as input Llama-3 random generated utterance groups transcripts (LLM_{rnd}) as described in Sec. 2.

We can observe that xTTS-based generation is able to improve over NeMo MSS when Fisher only transcriptions (Fisher) are used. When LLM generated transcriptions are used (LLM_{rnd}) its performance is on par/slightly worse. Instead for Parakeet, the difference between using LLM generated transcripts and using directly Fisher training set transcriptions is modest and actually the generated ones afford the best performance. In general, while the performance gain compared to the baseline synthetic data approaches (xTTS and NeMo MSS) is significant there remains a substantial gap compared to using in-domain data (Fisher). It appears that this gap cannot be bridged solely by scaling the amount of synthetic data.

In Figure 2 we report cpWER on Fisher for different amounts of adaptation data, both from Fisher training set and from synthetic approaches. It can be seen that for modest amounts of data (less than 5 h) the proposed approach can be competitive to using in-domain data, however as adaptation data amount is scaled its performance saturates quickly: the improvement between 50 h and 5 h is marginal when compared to the one afforded by using in-domain data. This trend is also observed for the other synthetic data approaches and suggests that indeed there is some inherent mismatch in all synthetic data approaches that prevents effective scaling. Again, for Parakeet at least, results suggest that this mismatch seems to be more related to the signal/acoustical content rather than the semantical content as the gap between using Fisher transcriptions or LLM ones appears to be modest.

4.2. Mixer 6 Speech

In Table 2, we show results obtained on the Mixer 6 scenario. The trends observed on Fisher are largely the same also here,

[2] github.com/popcornell/ASRLightningFT

Table 1: *Multi-speaker ASR results on Fisher test set with different adaptation data.*

Adaptation Data	amount (hours)	cpWER (%)	MIMO-WER (%)
-	0	44.94	26.15
Fisher	1960	13.76	13.58
Fisher	80	15.43	14.94
NeMo MSS	80	34.37	26.51
xTTS (Fisher)	80	24.88	24.07
xTTS (LLM_{rnd})	80	34.65	28.31
Parakeet (Fisher)	80	21.44	21.00
Parakeet (LLM_{rnd})	80	20.41	19.48
Parakeet (LLM_{rnd})	160	19.93	19.45

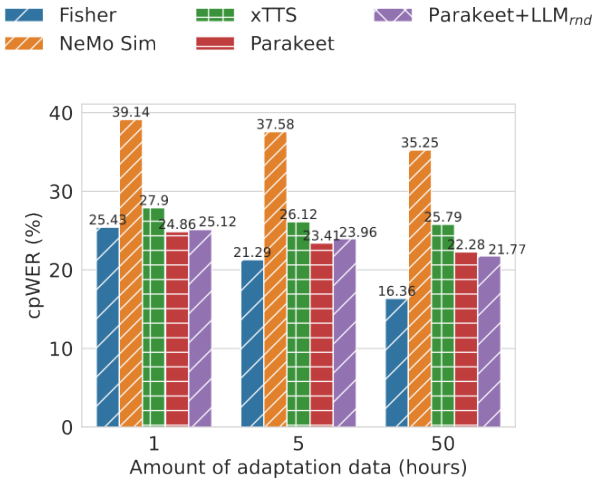

Figure 2: *Multi-speaker ASR results on Fisher test set for different adaptation data sources and quantity.*

despite the rather naive artificial reverberation strategy used for xTTS and Parakeet experiments. This confirms that the proposed approach can be also effective for far-field multi-speaker synthetic data, at least when compared to the classical approach (NeMo MSS results) and when available in-domain data is very scarce (here 2:30 h). Parakeet (LLM_{rnd}, 80 h) also compares favorably with the third and fourth rows, where we report the results of using instead the Fisher full 1960 h training set and a 80 h subset for adaptation. For these Fisher experiments, to reduce the mismatch due to the telephone lower sampling frequency, we applied telephone band-limiting to Mixer 6 in the inference phase. We also contaminated the Fisher 6 training data with reverberation as done for Parakeet and xTTS as described in Sec. 3.3.

4.3. Further discussion & remarks

Considering both Fisher and Mixer 6 experiments, the fact that Parakeet+LLM_{rnd} improves considerably over NeMo MSS while xTTS fails, suggest that turn-taking and para-linguistics may play a considerable role for multi-talker ASR.

Finally, for both Mixer 6 Speech and Fisher scenarios, we tried to use $50\,h$ of synthetic LLM_{rnd} data from the methods under study to augment a portion of in-domain data ($5\,h$ and $50\,h$) by mixing the two or by training on synthetic data and then fine-tune on in-domain data. However, in most instances we failed to improve significantly compared to using only the in-domain data, with xTTS and NeMo MSS actually degrading the performance. For example, by combining $50\,h$ of Parakeet (LLM_{rnd}) and $50\,h$ of original Fisher training data the model achieved a cpWER of 15.74% which is only marginally better than the 16.36% obtained with only $50\,h$ of Fisher (Figure 2). Interestingly, negligible or no improvement was also observed when the in-domain data was more modest (5 h). This result may be due to the fact that here we are trying to leverage an already strong pre-trained model and, as such, more than quantity, it is the quality of the adaptation data that matters most. As such future works may need to focus on few-shot adaptation of the TTS model to allow to match better the in-domain data.

Table 2: *Multi-speaker ASR results on Mixer 6 Speech eval set with different adaptation data.*

Adaptation Data	amount (hours)	cpWER (%)	MIMO-WER (%)
-	0	43.67	32.16
Mixer6	2.30	20.36	19.77
Fisher	1960	20.83	20.33
Fisher	80	22.12	21.36
NeMo MSS	80	36.71	28.21
xTTS (Mixer6)	2.30	25.99	24.47
xTTS (LLM_{rnd})	80	35.65	30.18
Parakeet (Mixer6)	2.30	23.52	22.82
Parakeet (LLM_{rnd})	2.30	23.70	22.12
Parakeet (LLM_{rnd})	80	21.25	20.17

5. Conclusions

In this work we study the use of synthetically generated data for multi-speaker ASR, focusing on the 2-speaker case. In detail, our goal is to compare different strategies to obtain such synthetic data i.e. by using artificially overlapped , a SotA conventional TTS model and, finally also a novel conversational TTS model capable of generating natively multi-speaker utterances. Results show that this approach is promising and significantly outperforms previous SotA multi-speaker simulation techniques. Furthermore, we show that, for the scenarios considered, it can achieve performance reasonably close to that of using in-domain data, but only when such in-domain data is limited to a few hours. For Mixer 6, our approach also obtained results comparable to using external real-world multi-speaker data (Fisher). In general experiments suggest that the LLM generated transcripts are reliable but that there is currently a performance gap with in-domain data (when this latter can be scaled). This gap is likely due to signal level mismatches and prevents effective scaling of this approach.

This work, as said, has several limitations. We only considered two-speaker conversational speech, short 30-second conversations, and relatively high SNR scenarios. These constraints were primarily imposed by the current limitations of the Parakeet TTS model. Further research is needed to overcome these limitations. For example to tackle more complex noisy/reverberant scenarios the TTS model needs to incorporate acoustic scenario modeling.

6. Acknowledgments

S. Cornell was supported by IC Postdoctoral Research Fellowship Program at CMU via ORISE through an agreement between U.S. DoE and ODNI. We'd also like to thank Google's TPU Research Cloud (TRC), which provided compute for generating synthetic Parakeet samples and Llama synthetic text utterances. Our work would not have been possible without their support.

7. References

[1] A. Radford *et al.*, "Robust speech recognition via large-scale weak supervision," in *ICML*. PMLR, 2023.

[2] Y. Peng *et al.*, "Reproducing whisper-style training using an open-source toolkit and publicly available data," in *Proc. of ASRU*. IEEE, 2023.

[3] N. Kanda *et al.*, "Large-scale pre-training of end-to-end multi-talker asr for meeting transcription with single distant microphone," *arXiv preprint arXiv:2103.16776*, 2021.

[4] A. Baevski *et al.*, "wav2vec 2.0: A framework for self-supervised learning of speech representations," *Advances in neural information processing systems*, vol. 33, 2020.

[5] W.-N. Hsu *et al.*, "Hubert: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM TASLP*, vol. 29, 2021.

[6] S. Chen *et al.*, "WavLM: Large-scale self-supervised pre-training for full stack speech processing," *IEEE Journal of Selected Topics in Signal Processing*, vol. 16, no. 6, 2022.

[7] S. Watanabe *et al.*, "CHiME-6 challenge: Tackling multispeaker speech recognition for unsegmented recordings," in *CHiME Workshop*, 2020.

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

[9] N. Ryant *et al.*, "The third dihard diarization challenge," *Proc. of Interspeech*, 2021.

[10] Y. Fujita *et al.*, "End-to-end neural speaker diarization with self-attention," in *Proc. of ASRU*. IEEE, 2019.

[11] K. Kinoshita, M. Delcroix, and N. Tawara, "Integrating end-to-end neural and clustering-based diarization: Getting the best of both worlds," in *Proc. of ICASSP*. IEEE, 2021.

[12] F. Landini *et al.*, "From simulated mixtures to simulated conversations as training data for end-to-end neural diarization," 2022.

[13] I. Medennikov *et al.*, "Target-speaker voice activity detection: a novel approach for multi-speaker diarization in a dinner party scenario," *Proc. of Interspeech*, 2020.

[14] N. Tawara *et al.*, "Ntt speaker diarization system for chime-7: multi-domain, multi-microphone end-to-end and vector clustering diarization," *CHiME Workshop*, 2023.

[15] N. Kanda *et al.*, "Serialized output training for end-to-end overlapped speech recognition," *Proc. of Interspeech*, 2020.

[16] ——, "Investigation of end-to-end speaker-attributed asr for continuous multi-talker recordings," in *Proc. of SLT*. IEEE, 2021.

[17] Z. Huang *et al.*, "Adapting self-supervised models to multi-talker speech recognition using speaker embeddings," in *Proc. of ICASSP*. IEEE, 2023.

[18] S. Cornell *et al.*, "One model to rule them all? towards end-to-end joint speaker diarization and speech recognition," in *Proc. of ICASSP*. IEEE, 2024.

[19] C. Li *et al.*, "Adapting multi-lingual asr models for handling multiple talkers," *Proc. of Interspeech*, 2023.

[20] T. Cord-Landwehr *et al.*, "Mms-msg: A multi-purpose multi-speaker mixture signal generator," in *Proc. of IWAENC*. IEEE, 2022.

[21] T. J. Park *et al.*, "Property-aware multi-speaker data simulation: A probabilistic modelling technique for synthetic data generation," *Proc. of Interspeech*, 2023.

[22] A. Rosenberg *et al.*, "Speech recognition with augmented synthesized speech," in *Proc. of ASRU*. IEEE, 2019.

[23] Z. Chen *et al.*, "Improving speech recognition using gan-based speech synthesis and contrastive unspoken text selection." in *Proc. of Interspeech*, 2020.

[24] N. Rossenbach *et al.*, "Generating synthetic audio data for attention-based speech recognition systems," in *Proc. of ICASSP*. IEEE, 2020.

[25] A. Tjandra, S. Sakti, and S. Nakamura, "Machine speech chain," *IEEE/ACM TASLP*, vol. 28, 2020.

[26] A. Fazel *et al.*, "SynthASR: Unlocking synthetic data for speech recognition," *Proc. of Interspeech*, 2021.

[27] X. Zheng *et al.*, "Using synthetic audio to improve the recognition of out-of-vocabulary words in end-to-end asr systems," in *Proc. of ICASSP*. IEEE, 2021.

[28] S. Ueno *et al.*, "Data augmentation for asr using tts via a discrete representation," in *Proc. of ASRU*. IEEE, 2021.

[29] T.-Y. Hu *et al.*, "Synt++: Utilizing imperfect synthetic data to improve speech recognition," in *Proc. of ICASSP*. IEEE, 2022.

[30] M. Soleymanpour *et al.*, "Synthesizing dysarthric speech using multi-speaker tts for dysarthric speech recognition," in *Proc. of ICASSP*. IEEE, 2022.

[31] E. Casanova *et al.*, "Asr data augmentation in low-resource settings using cross-lingual multi-speaker tts and cross-lingual voice conversion," in *Proc. of Interspeech*, 2023.

[32] T. Hori *et al.*, "Cycle-consistency training for end-to-end speech recognition," in *Proc. of ICASSP*. IEEE, 2019.

[33] M. K. Baskar *et al.*, "Eat: Enhanced asr-tts for self-supervised speech recognition," in *Proc. of ICASSP*. IEEE, 2021.

[34] J.-w. Jung *et al.*, "Augsumm: towards generalizable speech summarization using synthetic labels from large language model," *Proc. of ICASSP*, 2024.

[35] S.-L. Wu *et al.*, "Improving audio captioning models with fine-grained audio features, text embedding supervision, and llm mix-up augmentation," in *Proc. of ICASSP*. IEEE, 2024.

[36] C. Cieri, D. Miller, and K. Walker, "The Fisher corpus: A resource for the next generations of speech-to-text." in *LREC*, 2004.

[37] L. Brandschain *et al.*, "The Mixer 6 corpus: Resources for cross-channel and text independent speaker recognition," in *LREC*, 2010.

[38] J. Darefsky, G. Zhu, and Z. Duan, "Parakeet," 2024. [Online]. Available: https://jordandarefsky.com/blog/2024/parakeet/

[39] A. Clifton *et al.*, "100,000 podcasts: A spoken english document corpus," in *Proceedings of the 28th International Conference on Computational Linguistics*, 2020.

[40] Z. Liu *et al.*, "Autoregressive diffusion transformer for text-to-speech synthesis," *arXiv preprint arXiv:2406.05551*, 2024.

[41] G. Morrone *et al.*, "End-to-end integration of speech separation and voice activity detection for low-latency diarization of telephone conversations," *Speech Communication*, 2024.

[42] V. Panayotov *et al.*, "Librispeech: an asr corpus based on public domain audio books," in *Proc. of ICASSP*, 2015.

[43] E. Casanova *et al.*, "Xtts: a massively multilingual zero-shot text-to-speech model," *arXiv e-prints*, 2024.

[44] S. Team, "Silero vad: pre-trained enterprise-grade voice activity detector (vad), number detector and language classifier," https://github.com/snakers4/silero-vad, 2021.

[45] E. J. Hu *et al.*, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[46] T. Ko *et al.*, "A study on data augmentation of reverberant speech for robust speech recognition," in *Proc. of ICASSP*. IEEE, 2017.

[47] T. von Neumann *et al.*, "MeetEval: A toolkit for computation of word error rates for meeting transcription systems," *CHiME Workshop*, 2023.