2024 Asia Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)

Optimizing Multi-Speaker Speech Recognition with Online Decoding and Data Augmentation

Yizhou Peng*† and Eng Siong Chng*
* Nanyang Technological University, Singapore
E-mail: yizhou.peng@ntu.edu.sg
† National University of Singapore, Singapore

***Abstract*—This paper addresses missing transcription of short speeches from changed speakers in streaming ASR. This problem may be attributed to two factors: existing training data consists of only a single speaker in each training utterance, and the transformer memory of a single speaker during streaming decoding may impact the decoding of segments from the new speaker. To improve, we propose to leverage limited left-context streaming decoding and to include data augmentation of multi-speaker in training utterances. Our experimental results on Librispeech, Aishell-1, and SEAME corpus demonstrate 32%, 28%, and 9% relative improvements in word error rates (WER), character error rates (CER), and mixed error rates (MER) across English, Chinese, and code-switching datasets. These findings suggest robust solutions for real-time ASR applications in complex audio environments.**

I. Introduction

Since the advent of large-scale models like GPT [1] in natural language processing (NLP) and Whisper [2] in speech recognition as well as speech translation, there has been a significant surge in research aimed at leveraging Large Models to address a variety of tasks [3], [4]. In the meantime, to utilize abundant information in speech signals, SpeechGen [5] explores the application of prompt tuning to enhance the generative capabilities of Speech Language Models for various tasks. Also, the Qwen-Audio [6] model creates a universal audio-language model capable of processing diverse audio types, from human speech to natural sounds and music. These models have shown remarkable capabilities in understanding and generating human language. The final goal is to develop a single model that can seamlessly handle various tasks while achieving state-of-the-art performance for each mission.

However, despite the impressive advancements of these large-scale models, they often fall short of achieving the best performance on specific tasks when compared to specialized models with a similar number of parameters. This highlights the reality that specialized systems are still critical for most industry applications.

In the area of ASR, the capability to handle as many scenarios as possible, as well as computational efficiency, are two crucial factors when it comes to real applications. Specifically, traditional reading-style single-speaker speech recognition systems are no longer sufficient to meet the demands of modern applications because of the rapid development of the Internet, and the increasing prevalence of virtual meetings that cause continuous conversation with multiple speakers has become increasingly common. These scenarios require ASR systems to accurately transcribe conversations involving multiple participants, sometimes speaking simultaneously, which makes it more complex than that in single-speaker contexts [7]. Studies like those on streaming speaker-attributed ASR [7] and Qwen-Audio-Chat, which integrates audio and text inputs for multi-turn dialogues, emphasize the importance of these developments [6].

In addition to handling conversational and multi-speaker scenarios, there is another type of ASR systems that are designed to process and transcribe speech in real-time, making it indispensable for applications that require immediate feedback, such as live captioning, virtual assistants, and real-time communication platforms, namely streaming ASR. The need for low-latency, high-accuracy ASR solutions has driven significant research and development efforts in this area, such as RNN-Transducer [8], [9], U2++ Conformer [10], Zipformer-Transducer [11], even zero-shot streaming implementation for Whisper model [12].

In this paper, we focus on dealing with scenarios containing multiple speakers in a single decoding segment in the streaming ASR task. Specifically, we found that for single-speaker ASR systems, when performing offline decoding on sentences containing rapid speaker changes, the transcription for the changed speakers could be partially or totally missing. We propose that online streaming decoding can alleviate these issues to different extents with diverse configurations without requiring any model fine-tuning. Also, experimental results show that the multi-speaker data augmentation method [13] with our refinement strategy further improves the model performance on both single-speaker and multi-speaker testsets.

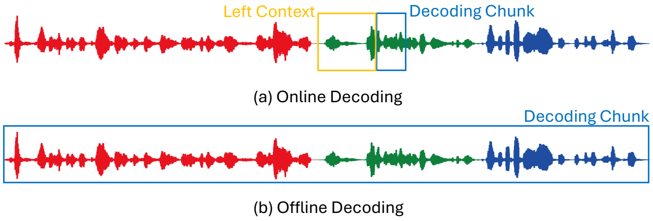
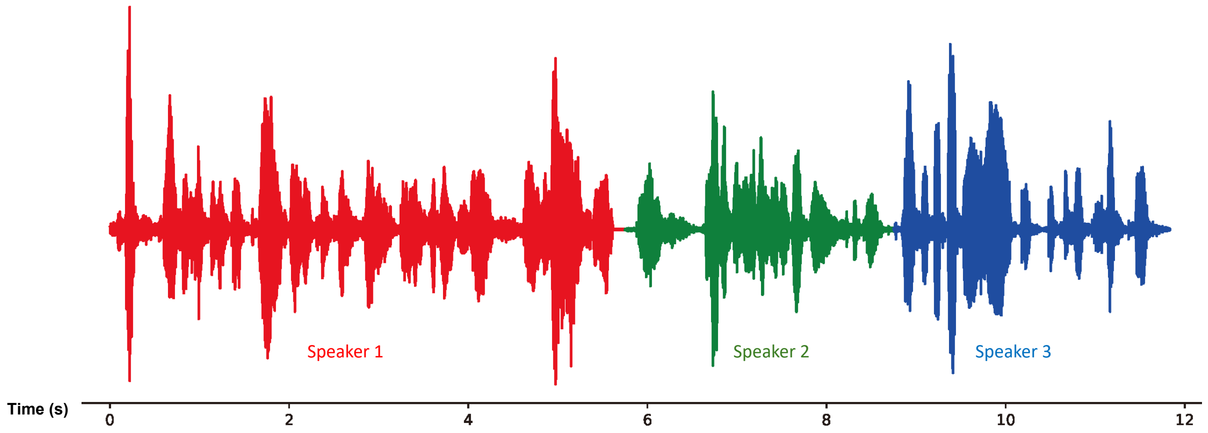


Fig. 1. Comparison between Offline and Online decoding for a sentence

This paper is organized as follows. Section I shows the background of this topic. Section II illustrates related works



Ref: Okay so you can actually apply for your preferred flat at any given town and how it actually. Ah I see but then I can't I can't leave it there for a certain period. I should say probably around eight eight hundred to one k.

Offline: Okay so you can actually apply for your preferred flat at any given **time** and how it actually. I should say probably around eight eight hundred to one k.

Online-1: Okey so you can actually apply for your preferred flat at any given **time** and how it actually. period. I should say probably around eight eight hundred to one k.

Online-2: Okay so you can actually apply for your preferred flat at any given **time** and how it actually. but then I can't I can't leave it there for a certain period. I should say probably around eight eight hundred to one k.

Fig. 2. An inference example of a multi-speaker speech segment using single-speaker ASR model, where the silence gap between each speaker is less than 0.5 seconds. **Ref** stands for reference text that is manually transcribed, **Offline** means the model sees the entire sentence when performing decoding. **Online-1** and **Online-2** perform online stream decoding whose decoding chunksize is 640ms, where **Online-1** uses 2.56s left context while **Online-2** sees only 1.28s.

to multi-speaker ASR and streaming ASR. Section III then includes methods, corpus information, model architectures, and training configurations. Succeedingly is section IV for experimental results and analysis. Finally, section V concludes the work.

II. Related Works

Multi-speaker speech recognition has become an essential focus area with the increasing prevalence of conversational and meeting scenarios where multiple participants may speak simultaneously. A model using token-level serialized output training (t-SOT) [14] to accurately identify and transcribe speech from multiple speakers in real-time is proposed [15], which significantly improves performance in overlapping speech scenarios. Additionally, a mixture encoder for joint speech separation and recognition [16] was proposed, leveraging explicit speech separation while incorporating cross-speaker context information to mitigate error propagation, achieving significant improvements.

Streaming ASR is critical for applications requiring real-time transcription, such as live captioning and virtual assistants. The introduction of cumulative attention mechanisms in streaming Transformers reduces latency while maintaining high accuracy by synchronizing attention heads within the model [17]. MiniStreamer framework [18], which enhances the Conformer model with chunked-context masking, optimizing it for edge devices with limited computational resources. These advancements illustrate the ongoing efforts to improve the efficiency and effectiveness of streaming ASR systems in various real-world applications.

III. Experiments

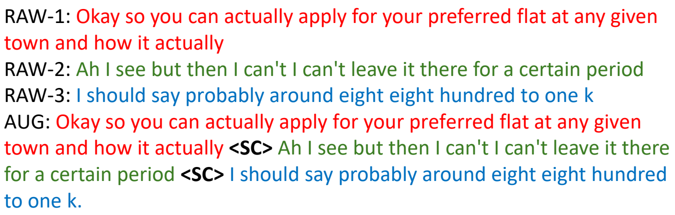
A. Methodologies

1) Limited Left-Context Streaming Decoding: In streaming ASR, chunk-size and left-context are critical parameters for balancing latency and accuracy during decoding. Fig. 1 (a) illustrates an example of online stream decoding where the Decoding Chunk is audio frames to form a specific chunk-size of the chunk, and the Left Context involves the amount of previous audio data considered when processing the current Decoding Chunk. Chunk-size refers to the length of audio data processed at one time, with smaller chunks reducing latency but potentially decreasing accuracy due to less context. In comparison, larger chunks increase accuracy at the cost of higher latency. To simplify, Fig. 1 (b) indicates that Offline Decoding includes the whole speech segment into one Decoding Chunk.

However, we find that in multi-speaker scenarios, accuracy is not necessarily improved when given more left context. On the contrary, when the model sees less left context, it produces more accurate recognition results, even compared with the Offline decoding method, which is generally believed to generate the best annotations. As shown in Fig. 2, for a single-speaker ASR model using Zipformer architecture [11] with a causal input layer, the Offline and Online-1 decoding methods produce results that almost lose all content from **Speaker 2** while the Online-2 decoding method only misses a few words. This suggests that for a trained single-speaker causal model, decoding with limited left context in streaming style can resolve the multi-speaker issues in some respect.

2) Multi-Speaker Data Augmentation: In [13], simply and randomly combining several speakers' speech segments together significantly improves the model performance on multi-speaker speech recognition while maintaining unharmed single-speaker recognition accuracy. Technically, this augmentation method works on-the-fly after speech features are extracted for batches during training, combining speech features of segments in one batch from two or three different speakers to form an augmented new batch. For transcriptions in the augmented training batch, it would simply add <SC> label, which stands for Speaker-Change, in between the transcription of combined segments while remaining unchanged for those not chosen for augmentation. Fig. 3 shows how the augmented annotation formed by given segments from three different speakers. However, to perform an on-the-fly style of augmentation, the combination can only occur in each batch during training, which might significantly reduce the diversity of combinations between different speakers.

To solve the above issue, we simplify the augment process by generating the augmented dataset before training, randomly combining speakers from the whole training set for each epoch, which maximizes the diversity of combinations of speakers.



RAW-1: Okay so you can actually apply for your preferred flat at any given town and how it actually
RAW-2: Ah I see but then I can't I can't leave it there for a certain period
RAW-3: I should say probably around eight eight hundred to one k
AUG: Okay so you can actually apply for your preferred flat at any given town and how it actually **<SC>** Ah I see but then I can't I can't leave it there for a certain period **<SC>** I should say probably around eight eight hundred to one k.

Fig. 3. An augmentation example of a multi-speaker speech segment. **RAW-{1,2,3}** stand for three speech segments from three different speakers. **AUG** means the resulting augmented segment that combines the three separate utterances with <SC> label in between where <SC> stands for Speaker-Change

B. Data

We perform our experiments on a diversity of datasets in languages including English, Chinese, and Code-switching scenarios to show the effectiveness of our methods. Correspondingly, we choose Librispeech [19], Aishell-1 [20], and SEAME [21] datasets for each language setting since these datasets are the most widely used for each language for research purposes in the last few years. Librispeech is a large-scale corpus of approximately 1,000 hours of English speech derived from audiobook recordings, containing read speech recorded by native speakers from the United States. Aishell-1 is an open-source Mandarin Chinese speech corpus consisting of about 178 hours of speech data, recorded from 400 speakers with various accents across China, containing several topics such as news reports and conversational contents, while recorded in reading style. SEAME (South East Asia Mandarin-English) is a conversational speech corpus featuring code-switching between Mandarin and English, collected from naturally occurring conversations and interviews in Singapore and Malaysia, in both formal and informal settings. Table I shows the statistics information for each dataset.

For Librispeech, we only report results on **Test-C**lean and **Test-O**ther. Dev$_{\text{Man}}$ and Dev$_{\text{Sge}}$ are two official subsets for evaluation purposes of the SEAME corpus. All sets with **-M** label stand for simulated Speaker-Mixed sets, consisting of two or three speakers in one segment, keeping the exact same duration and number of words as the original sets, following the multi-speaker data augmentation method mentioned in section III-A2.

TABLE I
OVERALL SPEECH DATA DISTRIBUTION FOR BOTH ASR MODEL TRAINING AND TESTING. **EN** STANDS FOR ENGLISH WHILE **CN** MEANS CHINESE. **EN-CN CS** CORRESPONDS TO ENGLISH CHINESE CODE-SWITCHING. **TRAIN-S** MEANS SINGLE-SPEAKER TRAINING SET WHILE **TRAIN-M** STANDS FOR MULTI-SPEAKER TRAINING SET.

Corpus	Language	Subset	Duration(Hrs)
Librispeech	EN	Train-S / Train-M	961.1
		Test-C / Test-C-M	5.4
		Test-O / Test-O-M	5.3
Aishell-1	CN	Train-S / Train-M	150.9
		Dev / Dev-M	18.1
		Test / Test-M	10.0
SEAME	EN-CN CS	Train-S / Train-M	93.6
		Dev$_{\text{Man}}$ / Dev$_{\text{Man}}$-M	7.5
		Dev$_{\text{Sge}}$ / Dev$_{\text{Sge}}$-M	3.9

C. Model

We perform all of our experiments using zipformer-transducer architecture, with icefall[1] toolkit that is supported by K2[2] project. The model architecture involves a zipformer [11] encoder, a stateless transducer [22] decoder, and a simple joiner network. The encoder is configured with layers of 2, 2, 3, 4, 3, and 2, each having subsampling factors of 1, 2, 4, 8, 4, and 2, respectively, following the official configuration of zipformer recipe. The input feature is 80-dim Mel frequency bins, computed on 25-ms windows with a stride of 10 ms. SpecAugment [23] is enabled for all experiments, while speed-perturb is disabled, as well as mix precision training is also disabled. We use 4000 secs, 2000 secs, and 800 secs as model batch sizes for Librispeech, Aishell-1, and SEAME experiments, respectively. We use BPE [24] as modeling units for English while using characters for Chinese, which results in 500 BPEs for Librispeech, 4338 units for Aishell-1, and 3136 units for SEAME corpus. The context size is set to one for Aishell-1 and SEAME, while it remains 2 for Librispeech. The learning rate scheduler and optimizer use the same configuration proposed in zipformer [11], named Eden LRScheduler and ScaledAdam optimizer, which achieve faster convergence and better performance than Adam and significantly reduce warmup steps during training. All of our models are trained with a maximum of 8x A100-40GB GPUs according to different batch size configurations for each corpus.

The models for single-speaker were trained for 40 epochs, 60 epochs, and 60 epochs for the Librispeech, Aishell-1, and SEAME corpus, respectively. For multi-speaker, the models

[1] https://github.com/k2-fsa/icefall
[2] https://github.com/k2-fsa/k2

TABLE II
WER/CER/MER RESULTS FOR MODELS TRAINED WITH **TRAIN-S** SETS. **OFFLINE-S** AND **ONLINE-S** STAND FOR OFFLINE AND ONLINE DECODING FOR SINGLE-SPEAKER MODELS. **INF** SPECIFIED IN CHUNK-SIZE AND LEFT-CONTEXT MEANS UNLIMITED DECODING CHUNK. **BOLD** NUMBERS ARE THE BEST RESULTS FOR EACH TESTSET.

Decoding Configs			Librispeech				Aishell-1				SEAME			
Methods	Chunk-size	Left-Context	Test-C	Test-O	Test-C-M	Test-O-M	Dev	Test	Dev-M	Test-M	Dev_{Man}	Dev_{Sge}	Dev_{Man}-M	Dev_{Sge}-M
Offline-S	inf	inf	**2.74**	**6.44**	**3.27**	9.97	**5.08**	**5.39**	6.64	7.24	**16.55**	**23.68**	18.78	**25.74**
Online-S	320 ms	1280 ms	3.66	9.57	3.74	10.23	5.90	6.46	6.03	6.67	19.09	26.97	20.04	28.17
	320 ms	2560 ms	3.55	9.17	3.84	11.14	5.85	6.31	6.30	7.31	18.89	26.83	20.32	28.58
	320 ms	5120 ms	3.52	9.07	4.11	12.24	5.85	6.30	6.70	8.33	18.79	26.83	20.74	28.93
	640 ms	1280 ms	3.40	8.59	3.44	**9.20**	5.57	6.11	**5.72**	**6.34**	17.94	25.37	**18.69**	26.54
	640 ms	2560 ms	3.29	8.29	3.47	9.91	5.5	6.04	5.92	6.82	17.84	25.26	18.95	26.81
	640 ms	5120 ms	3.28	8.18	3.69	11.08	5.5	6.01	6.27	7.63	17.82	25.19	19.47	27.13

were trained with only half of the epochs for single-speaker because we combined the Train-S and Train-M to form the final training dataset, keeping similar updating steps for the single and multi-speaker models.

IV. Experimental Results

The experimental results mainly contain two parts: single-speaker and multi-speaker. Each model is tested under both single and multi-speaker scenarios with both offline and online streaming decoding configurations. Word Error Rate (WER) is reported for Librispeech, Character Error Rate (CER) is reported for Aishell-1, and Mix Error Rate (MER) where CER is calculated for Chinese parts and WER is calculated for English parts, is reported for SEAME.

A. Single-Speaker Models

In this section, all of our models are trained with **Train-S** set of each corpus. Table II shows the results of these models on the corresponding testsets. In Table II, we can conclude that for a single-speaker ASR model, testing on single-speaker testsets would always obtain the best recognition results, either decoding with offline or online settings, compared with multi-speaker testsets. It also shows significant performance gaps in the offline decoding config between corresponding multi-speaker and single-speaker testsets, especially observed from Test-Other of Librispeech, Test of Aishell-1, and Dev_{Man} of SEAME, where the gaps reach the range of (13%, 55%) in relative, respectively.

From online decoding results, it is observed that a larger decoding chunk-size gives better WER results regardless of left-context size given for all situations in our experiments. However, left-context size shows a reversed influence on single-speaker versus multi-speaker testsets. Specifically, when changing left-context configurations for single-speaker testsets, it shows reasonably better performance each time scaling up. In contrast, in multi-speaker scenarios, the models produce much worse recognition results, which is around 10% relative performance degradation for each time left-context size scaling up. We even find that most of the best results for multi-speaker testsets come from chunk-size of 640 ms and left-context size of 1280 ms, such as Test-O-M of Librispeech, Dev-M and Test-M from Aishell-1, and Dev_{Man}-M from SEAME. In this configuration, the performance gap between single-speaker and multi-speaker testsets is significantly reduced to a range of (4%, 8%). The results show consistency with the situation we present in Fig. 2 that indicates single-speaker ASR systems produce better recognition results when given less left-context for multi-speaker speech data.

TABLE III
WER RESULTS FOR MODELS TRAINED WITH THE COMBINATION OF **TRAIN-S** AND **TRAIN-M** SETS FOR LIBRISPEECH. **SINGLE** AUGMENT CONFIG STANDS FOR THE MODELS TRAINED WITH THE SAME COPY OF **TRAIN-M** FOR EACH EPOCH, WHILE **MULTIPLE** MEANS THAT IN EACH EPOCH OF TRAINING, THE MODELS USE DIFFERENT **TRAIN-M** GENERATED SEPARATELY. **OFFLINE-M** AND **ONLINE-M** MEAN OFFLINE AND ONLINE DECODING FOR MULTI-SPEAKER MODELS. **CHUNK-SIZE** FOR ONLINE DECODING IS FIXED TO 640 MS HERE.

Configs			Librispeech			
Augment	Methods	Left-Context	Test-C	Test-O	Test-C-M	Test-O-M
Single	Offline-M	inf	2.72	6.60	2.80	6.76
	Online-M	1280 ms	3.37	8.53	3.39	8.68
		2560 ms	3.27	8.32	3.27	8.47
		5120 ms	3.28	8.20	3.28	8.44
Multiple	Offline-M	inf	2.73	6.59	2.79	6.74
	Online-M	1280 ms	3.32	8.47	3.34	8.49
		2560 ms	3.23	8.30	3.24	8.40
		5120 ms	3.17	8.24	3.20	8.28

B. Multi-Speaker Models

In this section, all of our models are trained with the combination of **Train-S** and **Train-M** sets. To begin with, we propose two augmentation strategies for **Train-M** generation for Librispeech corpus, **Single** augment means that we only generate one set of **Train-M** for all epochs, while **Multiple** means that we re-generate **Train-M** for each epoch, aiming to obtain more diverse speaker combination cases. Table III shows the performance of the resulting models. Comparing the offline decoding performance between the **Single** and **Multiple** augmentation configurations, no obvious difference is observed. However, when it comes to online streaming decoding, **Multiple** augmentation shows a bit of improvement among most of the testsets and left-context configurations. This surely suggests that a diversity of combinations of speakers

TABLE IV
WER/CER/MER results for models trained with the combination of **Train-S** and **Train-M** sets, using **Multiple Train-M** Augment config. **Offline-S** stands for offline decoding for single-speaker models, while **Offline-M** and **Online-M** correspond to offline and online decoding for multi-speaker models. **inf** specified in Chunk-size and Left-Context means unlimited decoding chunk.

Decoding Configs			Librispeech				Aishell-1				SEAME			
Methods	Chunk-size	Left-Context	Test-C	Test-O	Test-C-M	Test-O-M	Dev	Test	Dev-M	Test-M	Dev$_{\text{Man}}$	Dev$_{\text{Sge}}$	Dev$_{\text{Man}}$-M	Dev$_{\text{Sge}}$-M
Offline-S	inf	inf	2.74	6.44	3.27	9.97	5.08	5.39	6.64	7.24	16.55	23.68	18.78	25.74
Offline-M	inf	inf	2.73	6.59	2.79	6.74	4.80	5.17	4.81	5.24	16.63	23.53	16.78	23.52
Online-M	320 ms	1280 ms	3.58	9.23	3.65	9.45	5.54	5.96	5.48	5.96	18.55	25.96	18.60	26.15
	320 ms	2560 ms	3.52	9.05	3.53	9.20	5.40	5.89	5.40	5.87	18.14	25.84	18.56	25.98
	320 ms	5120 ms	3.49	8.91	3.51	9.08	5.38	5.87	5.38	5.91	18.34	25.80	18.45	25.95
	640 ms	1280 ms	3.32	8.47	3.34	8.49	5.18	5.68	5.19	5.69	17.70	24.79	17.73	24.74
	640 ms	2560 ms	3.23	8.30	3.24	8.40	5.13	5.60	5.15	5.62	17.57	24.68	17.57	24.67
	640 ms	5120 ms	3.17	8.24	3.20	8.28	5.13	5.58	5.16	5.65	17.55	24.68	17.57	24.68

generalizes the models more effectively. Consequently, **Multiple** augmentation is applied in our following experiments.

Table IV shows the performance for our multi-speaker ASR models for each corpus using **Multiple** data augmentation configuration. First of all, when comparing offline decoding results between single and multi-speaker models, significantly improved results for multi-speaker testsets are observed. In the meantime, improvements are also obtained regards to some of the single-speaker testsets. Specifically, Test-Clean of Librispeech, Dev, and Test of Aishell-1, and Dev$_{\text{Sge}}$ of SEAME show improved results. In contrast, the results of Test-Other of Librispeech and Dev$_{\text{Man}}$ of SEAME become a bit worse but still acceptable. Another interesting finding is that when it comes to online streaming decoding, consistent performance is found for all test sets and decoding configurations; that is, when decoding chunk size and left-context size are expanded wider, better recognition accuracy is obtained for not only single-speaker testsets but also multi-speaker counterparts. Finally, the results of multi-speaker testsets compared with single-speaker counterparts are now similar among all decoding configurations using the multi-speaker ASR models compared against single-speaker models shown in table II.

The results suggest that when models are exposed to varying conditions and combinations of different speakers, the effects of streaming decoding configurations on multi-speaker testsets are consistent with those observed for single-speaker testsets. This consistency indicates that the improvements in recognition accuracy with larger decoding chunk sizes and expanded left-context sizes apply similarly to both single and multi-speaker scenarios, highlighting the robustness of the model in handling diverse and complex audio.

V. Conclusion

This study demonstrates effective strategies to improve single-speaker ASR systems in multi-speaker scenarios through online streaming decoding and multi-speaker data augmentation. By optimizing left-context configurations and employing diverse speaker combinations, we achieved significant improvements in speech recognition accuracy. Our methods offer practical solutions for real-time ASR applications, enhancing their ability to handle complex and dynamic audio environments. The results underscore the importance of specialized configurations and data augmentation in advancing ASR technology to meet the demands of modern communication systems. Future work will focus on further refining these techniques for tasks such as online speaker diarization and voice activity detection, together with speech recognition function.

VI. Acknowledgements

The computational work for this article was partially performed on resources of the National SuperComputing Centre (NSCC), Singapore.

References

[1] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving language understanding by generative pre-training," *OpenAI*, 2018. [Online]. Available: https://cdn.openai.com/research-covers/language-unsupervised/language_understanding_paper.pdf.

[2] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *OpenAI*, 2022. [Online]. Available: https://cdn.openai.com/papers/whisper.pdf.

[3] J. Wu, X. Hu, Y. Wang, B. Pang, and R. Soricut, "Omni-smola: Boosting generalist multimodal models with soft mixture of low-rank experts," *arXiv preprint arXiv: 2312.00968*, 2023.

[4] P. K. Rubenstein, C. Asawaroengchai, D. Nguyen, *et al.*, "Audiopalm: A large language model that can speak and listen," *arXiv preprint arXiv: 2306.12925*, 2023.

[5] H. Wu, K.-W. Chang, Y.-K. Wu, and H.-y. Lee, "Speechgen: Unlocking the generative power of speech language models with prompts," *arXiv preprint arXiv: 2306.02207*, 2023.

[6] Y. Chu, J. Xu, X. Zhou, *et al.*, "Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models," *arXiv preprint arXiv: 2311.07919*, 2023.

[7] N. Kanda, J. Wu, Y. Wu, *et al.*, “Streaming Speaker-Attributed ASR with Token-Level Speaker Embeddings,” in *Proc. INTERSPEECH 2022*.

[8] K. Rao, H. Sak, and R. Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with rnn-transducer,” in *Proc. 2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*.

[9] S. Wang, P. Zhou, W. Chen, J. Jia, and L. Xie, “Exploring rnn-transducer for chinese speech recognition,” in *Proc. 2019 Asia-Pacific Signal and Information Processing Association Annual Summit and Conference (APSIPA ASC)*.

[10] D. Wu, B. Zhang, C. Yang, *et al.*, *U2++: Unified two-pass bidirectional end-to-end model for speech recognition*, 2021. arXiv: 2106.05642.

[11] Z. Yao, L. Guo, X. Yang, *et al.*, “Zipformer: A faster and better encoder for automatic speech recognition,” in *Proc. The Twelfth International Conference on Learning Representations*, 2023.

[12] D. Macháček, R. Dabre, and O. Bojar, “Turning whisper into real-time transcription system,” in *Proceedings of the 13th International Joint Conference on Natural Language Processing and the 3rd Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics: System Demonstrations*, Nov. 2023.

[13] S. Thomas, H. Kuo, G. Saon, and B. Kingsbury, “Multi-speaker data augmentation for improved end-to-end automatic speech recognition,” *Proc. ICASSP*, 2023.

[14] N. Kanda, Y. Gaur, X. Wang, Z. Meng, and T. Yoshioka, “Serialized Output Training for End-to-End Overlapped Speech Recognition,” in *Proc. Interspeech 2020*.

[15] N. Kanda, J. Wu, Y. Wu, *et al.*, “Streaming Multi-Talker ASR with Token-Level Serialized Output Training,” in *Proc. Interspeech 2022*.

[16] S. Berger, P. Vieting, C. Boeddeker, R. Schlüter, and R. Haeb-Umbach, “Mixture Encoder for Joint Speech Separation and Recognition,” in *Proc. INTERSPEECH 2023*.

[17] M. Li, S. Zhang, C. Zorila, and R. Doddipatla, “Transformer-based streaming asr with cumulative attention,” *Proc. ICASSP*, 2022.

[18] H. Gulzar, M. R. Busto, T. Eda, K. Itoyama, and K. Nakadai, “Ministreamer: Enhancing small conformer with chunked-context masking for streaming asr applications on the edge,” *INTERSPEECH 2023*, 2023. DOI: 10.21437/interspeech.2023-1162.

[19] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An asr corpus based on public domain audio books,” *Proc. ICASSP*, 2015.

[20] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, “Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline,” *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, 2017.

[21] D.-C. Lyu, T.-P. Tan, E.-S. Chng, and H. Li, “Mandarin–english code-switching speech corpus in south-east asia: Seame,” *Language Resources and Evaluation*, 2015.

[22] M. Ghodsi, X. Liu, J. Apfel, R. Cabrera, and E. Weinstein, “Rnn-transducer with stateless prediction network,” in *Proc. ICASSP*, 2020.

[23] D. S. Park, W. Chan, Y. Zhang, *et al.*, “SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition,” in *Proc. INTERSPEECH 2019*, 2019.

[24] R. Sennrich, B. Haddow, and A. Birch, “Neural machine translation of rare words with subword units,” *arXiv preprint arXiv:1508.07909*, 2015.