Interspeech 2024
1-5 September 2024, Kos, Greece

Contextual Biasing Speech Recognition in Speech-enhanced Large Language Model

Xun Gong[1], *Anqi Lv*[2], *Zhiming Wang*[2], *Yanmin Qian*[1*]

[1]Auditory Cognition and Computational Acoustics Lab
MoE Key Laboratory of Artificial Intelligence, AI Institute
Department of Computer Science and Engineering, Shanghai Jiao Tong University, Shanghai, China
[2]AntGroup, Shanghai, China

gongxun@sjtu.edu.cn

Abstract

Recently, the rapid advancements in audio- and speech-enhanced large language models (SpeechLLMs), such as Qwen-Audio and SALMONN, have significantly propelled automatic speech recognition (ASR) forward. However, despite the improvements in universal recognition capabilities, bias word recognition persists as a prominent challenge for SpeechLLM, and is not extensively studied. In this study, we introduce two contextual biasing strategies aimed at improving the bias word recognition of SpeechLLM. Firstly, we explored two types of biasing prompts for SpeechLLMs, achieving 10% relative reduction in bias word error rate (WER). However, as the size of the bias list increased, performance significantly declined due to hallucination. Subsequently, we built the biasing fusion network for SpeechLLM that integrates high-level bias embeddings with the SpeechLLM framework. Our experiments conducted on the LibriSpeech test-clean/-other datasets demonstrate that our method achieves up to 10%/35% relative reduction in overall/bias WER compared to our baseline.

Index Terms: contextual biasing, speech recognition, speech-enhanced large language model

1. Introduction

Leveraging the capabilities of text-only large language models (LLMs), researchers have sought to integrate audio and speech within a unified framework. This integration has achieved performance on par with or surpassing that of traditional supervised models across various tasks, such as audio captioning, speech understanding, automatic speech recognition (ASR), and so on [1, 2, 3, 4, 5]. This advancement highlights the potential of LLMs in enhancing our interaction with and understanding of audio information, setting a new benchmark for speech recognition technologies. Qwen-Audio [1] exemplifies the capabilities of speech data fusion, leveraging the powerful generative and cognitive abilities of LLMs for downstream tasks to achieve performance levels unattainable by traditional end-to-end (E2E) ASR models, such as attention encoder-decoder (AED), recurrent neural network transducer (RNNT) and connectionist temporal classification (CTC) [6].

However, recognition of bias words (also referred to as named entities, such as proper names) present a significant challenge not only in the tranditional E2E ASR systems but also in the above speech-enhanced LLMs (SpeechLLM) [7, 8]. The recognition of these bias words is particularly problematic compared with conventional models. The issue is that the extensive textual knowledge embedded within LLMs, which, while beneficial in many contexts, can mislead the ASR process by introducing biases that skew recognition away from accurate interpretations of speech input. This complex interplay between the textual knowledge of LLMs and the speech input underscores the need for tailored approaches that can mitigate these biases and enhance ASR performance.

Early approaches borrowed from the DNN-HMM framework are to construct weighted finite state transducer (WFST) for bias words, and then incorporate into decoding [9]. Similarly, joint training and decoding with an additional contextual/personalized language model are explored by shallow fusion or other techniques [7, 10, 11, 12, 13, 14]. The main issue with constructing additional bias-specific LMs or FSTs is their lack of flexibility. With the evolution of LLMs, there has been an increasing interest in leveraging LLMs combined with prompts to correct ASR results. Personalized databases for working with LLMs are proposed to update the user bias list in text-only LLMs [15]. Re-scoring technique is explored by providing additional contextual information to a LLM [8] by utilizing the dynamic and zero-shot capabilities of LLMs. However, when it comes to correcting results, those approaches cannot utilize any speech information, highlighting a gap between leveraging contextual textual knowledge and integrating acoustic signals for more accurate ASR.

On the other hand, attention-based contextual biasing methods have been explored such as CLAS [16], C-RNNT [17] and others [18, 19, 20]. PromptASR [21] introduces prompts into E2E ASR system using BERT, enabling contextualized speech recognition with adjustable transcription styles. Fine-grained phoneme information is utilized to boost bias recognition in [22, 23]. Phone-TCPGen [22] introduces a novel approach of incorporating subword-level phoneme-aware encodings into TCPGen, whereas Qiu et al. [23] enhances transducer by utilizing phonemic and text-only information for bias words. These advancements highlight a trend towards more adaptable, context-sensitive ASR models.

To address the challenge of bias word recognition in SpeechLLM, we propose two biasing strategies: the biasing prompts and the biasing fusion network. Initially, we present two types of biasing prompts designed to enhance SpeechLLM's input. One is to extends the token list with special tags that surrounds the bias words (referred to as 'special tagged' prompt), while the second method embeds bias words within naturally constructed human language (referred to as 'natural language' prompt). However, we observed a huge performance decrease with the expansion of the bias list, primarily attributed to the hallucination of LLM. To counteract this issue, especially in scenarios with extensive bias lists, we propose a biasing fusion network that builds upon SpeechLLM. This method starts with a lightweight text encoder dedicated to encode bias words

*Corresponding author

10.21437/Interspeech.2024-965

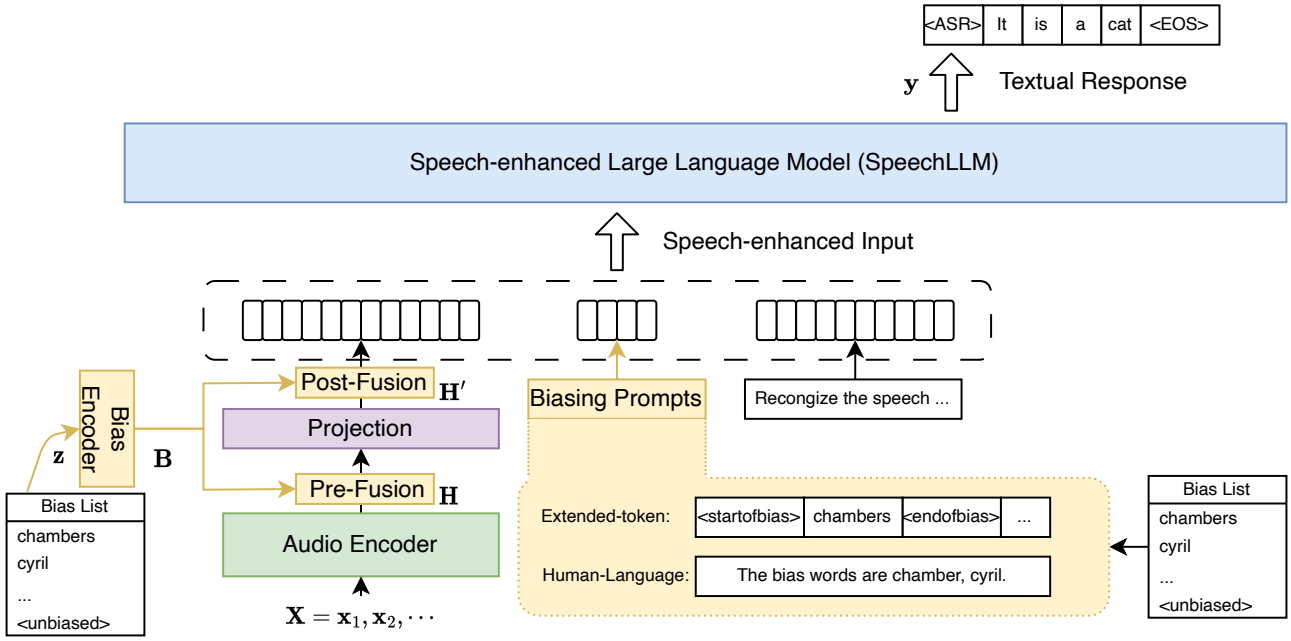

Figure 1: *Two different biasing strategies proposed for SpeechLLM. The biasing prompt strategy and the biasing fusion network is shown in yellow color.*

into a sequence of bias embeddings. Bias embeddings, combined with intermediate features, are then merged to adjust the hidden states of SpeechLLM. This strategy is specifically engineered to enhance SpeechLLM's ability to accurately recognize targeted words. For these methods, we adopt low-rank adaptation (LoRA) [24] for fine-tuning, which effectively reduces trainable parameters. Our approaches surpass our baseline in recognizing bias words, and, compared to prior studies, our experiments demonstrate competitive or even superior performance on both overall and bias word accuracy.

2. Speech-enhanced Large Language Model (SpeechLLM) for ASR

Basically, a traditional E2E ASR model processes acoustic features $\boldsymbol{X} = [\boldsymbol{x}_1, \cdots, \boldsymbol{x}_T]$ as its input and infers the output tokens $\boldsymbol{y} = [\boldsymbol{y}_1, \cdots, \boldsymbol{y}_L]$ using ASR models.

Recently, the rapid development of large language models (LLMs) has significantly increased interest in speech-enhanced LLMs (SpeechLLMs), which demonstrate impressive performance in ASR tasks [1, 2]. As depicted in Figure 1, the typical architecture of a SpeechLLM consists of three components: the audio encoder, the projection module, and the foundational text-based LLM. Initially, acoustic features $\boldsymbol{X}$ are input into the audio encoder, generating $\boldsymbol{H} = \text{Audio-Encoder}(\boldsymbol{X})$. Then, the projection module maps $\boldsymbol{H}$ into $\boldsymbol{H}'$ via $\boldsymbol{H}' = \text{Projection}(\boldsymbol{H})$, aligning it with the dimensional level of the LLM's textual embeddings. $\boldsymbol{H}'$ is then fed into LLM along with various textual prompts, which are designed to direct the LLM towards specific downstream tasks, such as using 'recognize the speech' with <ASR> for the ASR task.

$$y_i \overset{casual}{\longleftarrow} \text{LLM}(\boldsymbol{H}', <\text{ASR}>, \boldsymbol{y}_{<i-1}). \quad (1)$$

Multi-task training is used to facilitate knowledge sharing and collaborative learning across similar speech-based tasks.

To reduce the training cost for SpeechLLM, researchers use pre-trained modules for most components and employ low-rank adaptation [24] to speed up the training process. The audio encoder is initialized with Whisper-large-v2 [25], while foundational models like Qwen [26] or LLaMA [27] are chosen for initializing the LLM. The training objective is formulated as a multi-class cross-entropy loss for each predicted token.

3. Contextual Biasing SpeechLLM

To bias the SpeechLLM, we introduce two strategies, the biasing prompt method and the biasing fusion method in Figure 1.

3.1. Biasing Prompts for SpeechLLM

Harnessing the capabilities of LLMs, we straightforwardly incorporate a bias word list into the input of SpeechLLM as pre-retrieved knowledge [28]. Two biasing prompt templates are pre-defined for bias words:

- **Special tagged prompt** We extend the LLM's token list to utilize the bias words with <startofbias> and <endofbias>. During recognition, LLM is trained to learn bias words such as '<startofbias>Cuthbert<endofbias>' and '<startofbias>Marilla<endofbias>' before <ASR>. The <unbiased> token is also added and used for training to specify biased and unbiased situation, where bias words are inaccessible for unbiased situation.
- **Natural language prompt** To avoid the complexities of extending the token list, we also develop a more naturally template, such as 'The bias words are Cuthbert and Marilla'. Similarly, we randomly drop the bias prompt above not only for the unbiased situation but also to make SpeechLLM more robust to the prompt sentence.

Moreover, to deal the scenarios where the bias list is totally irrelevant, we add negative examples during training.

3.2. Biasing Fusion for SpeechLLM

Although the biasing prompts seems elegant to contextual bias the SpeechLLM, their effectiveness significantly drop as the bias list enlarges To address this issue, we introduce a biasing fusion network for SpeechLLM, shown in Figure 1. This approach enables SpeechLLM to more effectively handle bias words that may occur in the decoded sentences.

Bias Encoder

The first step in biasing SpeechLLM is encoding the bias words using a bias encoder. <unbiased> is added, to aid in distinguishing unbiased words from biased ones. The bias encoder processes the augmented bias list $[z_1, \cdots, z_N, <\text{unbiased}>]$, producing a sequence of bias embeddings $\boldsymbol{B}$. With the variable length of bias text post-tokenization, A pooling operation is added to unify the variable length of bias word. It captures the statistical properties of z_i by incorporating both the mean and standard deviation.

$$\boldsymbol{b}_i = \text{Pool}(\text{Bias-Encoder}(z_i)), \tag{2}$$

where $\boldsymbol{b}_i \in \mathbb{R}^D$, $\boldsymbol{B} = [\boldsymbol{b}_1, \cdots, \boldsymbol{b}_{N+1}]$. To simplify the network and maintain the compatibility between bias encoder and SpeechLLM, we use the same tokenizer as the SpeechLLM.

Biasing Fusion We fuse the bias embedding sequence with intermediate features $\boldsymbol{M}$ inside SpeechLLM. Two different intermediate features $\boldsymbol{M} = \boldsymbol{H}$ (pre-projection fusion) and $\boldsymbol{M} = \boldsymbol{H}'$ (post-projection fusion) are explored during the fusion. As the projection module in SpeechLLM make an important connection between audio and text, the intermediate features $\boldsymbol{H}$ and $\boldsymbol{H}'$ can performs a different role and have a different effect when fusing with biasing embeddings.

To transition from unbiased intermediate features $\boldsymbol{M}$ into biased features $\boldsymbol{M}^b$, we introduce an auxiliary multi-head attention (MHA) module, where the key and value comes from $\boldsymbol{B}$, the query is $\boldsymbol{M}$ and the dimension of $\boldsymbol{M}$ is also D. The whole biasing fusion process is:

$$\boldsymbol{M}^b = \boldsymbol{M}^{+}\text{MHA}(\text{query}=\boldsymbol{M}, \text{key}=\boldsymbol{B}, \text{value}=\boldsymbol{B}), \tag{3}$$

where a residual connection is added to maintain the stability.

4. Experimental Setup

Qwen-Audio [1] is adopted as our pre-trained SpeechLLM model. The audio encoder is based on the Whisper-large-v2 model [25], which has 640M parameters. It is a 32-layer transformer encoder with attention dimension equal to 1280, takes 80-dim fbank features as input with two convolution downsampling layers of downsampling rate 4. The fundamental LLM is based on a 32-layer transformer decoder with attention dimension equal to 4096, Qwen-7B [26] with totally 7.7B parameters. The projection layer is a Linear layer mapping the 1280-dim audio features to 4096. The vocabulary size is 155947. Qwen-Audio [1] is then trained with multi-task targets where ASR task is included.

For the biasing prompt method, we use supervised fine-tuning (SFT) technique with LoRA [24] applied on the LLM weights with 70M tunable parameters. As for the biasing fusion method, the bias encoder have different architectures. Firstly, train-from-scratch transformer encoder architecture that has 512 dimension with 8 heads and 12 layers is explored. Then, when taking LLM as the bias encoder, the same LLM is used as SpeechLLM's LLM part, and the first layer's latent feature is used as bias embeddings $\boldsymbol{B}$. Meanwhile, tokenized embeddings is also used to simplify the training. To deal with the dimension mismatch between bias embeddings $\boldsymbol{B}$ and the intermediate features $\boldsymbol{M}$, the projection has changed to match D in MHA, which also has 8 heads. Besides the trainable MHA, we also applying LoRA [24] on different parts of SpeechLLM, where the rank is 8 and alpha is 32 with dropout equal to 0.05.

The experiments are conducted on LibriSpeech [29], where 100 hours train clean set is used by default, and 960 hours whole training set is used to provide better performance. As for the division of rare words and common words, we follow Rare5k [10], where the 209.2K bias words comes out beside the 5,000 most common words in the 960h training set. Word error rate (WER) (%), unbiased WER (U-WER) and biased WER (B-WER) are evaluated over all test sets. Unbiased/biased WER are counted over common/bias words, and insertion error is counted to biased error if the insertion word is in the bias list.

5. Experimental Results

5.1. Biasing Prompts for SpeechLLM

Table 1: *The performance WER (U-WER/B-WER) (%) on LibriSpeech test sets results with different prompt types and different bias number N for biasing-prompted SpeechLLM. <UB> means the special token for <unbiased> situation.*

Bias Prompt Type	N	test-clean	test-other
-	-	2.0 (1.3/8.4)	4.2 (2.6/18.4)
special tagged	10	26.9 (24.1/49.7)	34.1 (31.4/58.1)
special tagged	5	6.4 (3.5/30.1)	12.8 (9.5/42.1)
special tagged	3	2.0 (1.3/7.5)	4.0 (2.6/16.4)
w/o <UB>	3	2.1 (1.3/7.8)	4.1 (2.7/16.8)
w/o negative	3	2.1 (1.3/7.6)	4.1 (2.7/16.5)
natural language	5	8.0 (5.0/33.1)	15.6 (9.8/47.1)
natural language	3	2.2 (1.5/7.8)	4.2 (2.7/16.6)
w/o empty	3	2.2 (1.5/8.0)	4.5 (3.0/17.3)
w/o negative	3	2.2 (1.5/7.9)	4.2 (2.8/16.8)

We explore two bias prompt types mentioned in Section 3.1, the special tagged prompt and the natural language prompt. For the 'special tagged' prompt, at $N = 3$, the performance is optimized, maintaining a stable U-WER while improving the B-WER by approximately 10%. We also observe that adding <unbiased> token and negative examples do help to the bias word recognition. As for the 'natural language' prompt, it achieves similar optimal result as 'special tagged', but get silightly worse compared to 'special tagged', which degrade the performance of U-WER by 3%. The reason is that the SpeechLLM (Qwen-Audio-Chat) model we choose has many tasks, and then wrongly specify the ASR task to 'keyword spotting' by our natural language prompt. This also corroborates that we can see a great performance drop on "w/o empty" prompts, because such examples effectively correct the misallocation of tasks.

Both approaches show a clear trend where performance degrades with an increase in the number of bias words (N), especially when $N > 3$, highlighting a threshold beyond which additional bias words adversely affect model accuracy. The performance drop, particularly observed with the increase in the number of bias words beyond a certain threshold in the biasing-prompted SpeechLLM, can be attributed to the hallucination phenomenon commonly seen in LLMs [30]. This phenomenon

Table 3: *Performance (WER (U-WER/B-WER)) (%) comparsions of different deep biasing methods and our proposed bias-fusion SpeechLLM method.* ***Bold*** *numbers denotes the best performance of B-WER for our proposed method in each block.* *Underline* *number denotes the best performance of WER compared with all methods with comparable supervised training data. The 'proposed' method is with our biasing fusion network and post-projection fusion with LoRA on LLM and different bias encoding strategies.*

ASR Model	Train Set	$N = 100$		$N = 500$	
		test-clean	test-other	test-clean	test-other
Qwen-Audio (SpeechLLM) [1]	-	2.0 (1.3/8.4)	4.2 (2.6/18.4)	2.0 (1.3/8.4)	4.2 (2.6/18.4)
Phone-TCPGen [22]	100h	-	-	4.9 (-/12.1)	15.6 (-/31.5)
proposed (Encoder=Scratch Encoder)	100h	1.9 (1.3/6.9)	3.9 (2.6/15.3)	2.0 (1.4/7.3)	4.2 (2.8/15.8)
proposed (Encoder=LLM)	100h	1.9 (1.3/**6.8**)	3.8 (2.6/**15.0**)	2.0 (1.4/**7.1**)	4.1 (2.8/**15.4**)
DB-RNNT+DB-LM [10]	960h	2.0 (1.5/5.7)	5.8 (4.9/14.1)	2.1 (1.6/6.2)	6.1 (5.1/15.1)
Phone-TCPGen [22]	960h	-	-	2.2 (-/4.6)	6.0 (-/12.3)
PromptASR+history [21]	7000h	1.7 (-/-)	4.1 (-/-)	2.0 (-/-)	4.5 (-/-)
proposed (Encoder=Scratch Encoder)	960h	1.6 (1.3/**5.5**)	3.8 (2.6/**13.5**)	1.9 (1.4/**6.0**)	3.9 (2.7/**14.2**)
proposed (Encoder=LLM)	960h	1.8 (1.3/6.1)	3.8 (2.6/14.9)	1.9 (1.4/6.5)	4.1 (2.7/15.4)

Table 2: *Performance (WER (U-WER/B-WER)) (%) comparsions of different deep biasing methods and our proposed bias-fusion SpeechLLM method on librispeech test-other set. The training set is 100h and the bias list size N is 100 for both training and evaluation. N/A denotes no parts of SpeechLLM is used by LoRA and training.*

Bias Encoder	Fusion	LoRA Part	WER (%)	
			Tot	U/B
-	-	-	4.2	2.6/18.4
Scratch Encoder	Pre	Audio Encoder	4.8	3.4/16.8
	Pre	Projection	4.4	3.0/16.2
	Pre	LLM	4.2	2.8/16.0
	Pre	N/A	4.2	2.8/16.3
	Post	Audio Encoder	4.4	3.1/16.8
	Post	Projection	4.1	2.8/15.6
	Post	LLM	3.9	2.6/15.3
	Post	N/A	4.1	2.7/16.0
LLM (Qwen) Plain	Post	LLM	3.8	2.6/15.0
	Post	LLM	3.8	2.6/15.2

occurs when the input prompts are too long. The first few words are recognized accurately, but the LLM then proceeds to generate text that is significantly deviate from the intended audio content.

5.2. Biasing Fusion for SpeechLLM

Following the setup in Section 3.2, we explore different types of bias encoders, different fusion positions (pre/post of the projection module) for bias embeddings and intermediate and different LoRA modules to be tuned during biased supervised fine-tuning in Table 2. Firstly, we explore a train-from-scratch bias encoder (denoted as Scratch Encoder) and evaluate the effectiveness of different LoRA components. Results show that applying LoRA on LLM obtains the best performance with relative B-WER reduction 13% for pre-fusion method and 17% for post-fusion method, respectively. However, LoRA on the audio encoder or the projection module hurts the recognition of unbiased words, evidenced by an increase in U-WER for both pre-/post- fusion. This is because fine-tuning these components may destroy the robustness of the strong audio part in SpeechLLM. Experiments also shows that if no LoRA is applied (N/A) to SpeechLLM, the biased part can still work fine (18.4% $\rightarrow$ 16.3%), however, not as effectively as LLM with LoRA.

Compared the first and the second sub-block for scratch encoder, we can conclude that pre fusion is worse than post fusion by an average of 4% accross different LoRA parts. This indicate us that post-fusion, which leverages bias embeddings $\boldsymbol{B}$ derived from textual data, aligns better with the latent space of $\boldsymbol{H}'$, thus proving more effective. Besides, we also tried different bias encoders, the first-layer output of LLM (the same as the LLM part of SpeechLLM) and the plain embeddings are evaluated and the LLM's hidden features seems to be effective as it is pre-trained with large text and can capture high-level textual information more effectively.

Shown in Table 3, we conclude our bias fusion network with two different architectures, the scratch encoder and the pre-trained LLM encoder, with post fusion and LoRA on LLM part. Results show that for the 100 hours train set, the pre-trained LLM encoder outperforms and achieve 19%/16% for $N = 100/500$ relative B-WER reduction. As for the whole 960 hours set, our proposed sctrach encoder performs best by 35%/25% relative B-WER reduction, for $N = 100/500$. Meanwhile, it also outperforms the prior works and reaches 1.8%/3.8% WER for librispeech test sets.

6. Conclusion

In this study, we proposed two innovative approaches to address bias word recognition in speech-enhanced LLM-based ASR, the biasing prompt method and the biasing fusion method. The biasing prompt method employs supervised fine-tuning with two pre-defined prompts, which is evaluted its effectiveness by 10% relative bias word error rate (B-WER) reduction. As the bias list grows, the above method drop performances a lot due to LLM's hallucination. Therefore, we propose the biasing fusion method to fuse embedded bias words with intermediate features of SpeechLLM. Different bias encoders are explored to find the most effective architecture, different fusion positions are explored to enhance performance. Our method reaches 19% relative B-WER reduction for 100h training set with $N = 100$ and 35% for 960h training set, respectively.

7. Acknowledgements

This work was supported in part by China NSFC projects under Grants 62122050 and 62071288, in part by Shanghai Municipal Science and Technology Commission Project under Grant 2021SHZDZX0102, and in part by Ant Group and Ant Group Research Intern Program.

8. References

[1] Y. Chu, J. Xu, X. Zhou, Q. Yang, S. Zhang, Z. Yan, C. Zhou, and J. Zhou. Qwen-audio: Advancing universal audio understanding via unified large-scale audio-language models.

[2] C. Tang, W. Yu, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang. Salmonn: Towards generic hearing abilities for large language models.

[3] J. Wang, Z. Du, Q. Chen, Y. Chu, Z. Gao, Z. Li, K. Hu, X. Zhou, J. Xu, Z. Ma, W. Wang, S. Zheng, C. Zhou, Z. Yan, and S. Zhang. Lauragpt: Listen, attend, understand, and regenerate audio with gpt.

[4] M. Wang, W. Han, I. Shafran, Z. Wu, C.-C. Chiu, Y. Cao, Y. Wang, N. Chen, Y. Zhang, H. Soltau, P. Rubenstein, L. Zilka, D. Yu, Z. Meng, G. Pundak, N. Siddhartha, J. Schalkwyk, and Y. Wu. Slm: Bridge the thin gap between speech and text foundation models.

[5] S. Deshmukh, B. Elizalde, R. Singh, and H. Wang. Pengi: An audio language model for audio tasks.

[6] J. Li, "Recent advances in end-to-end automatic speech recognition," *APSIPA Transactions on Signal and Information Processing*, vol. 11, no. 1, 2022.

[7] D. Zhao, T. N. Sainath, D. Rybach, P. Rondon, D. Bhatia, B. Li, and R. Pang, "Shallow-Fusion End-to-End Contextual Biasing," in *Proc. Interspeech 2019*, 2019, pp. 1418–1422.

[8] C. Sun, Z. Ahmed, Y. Ma, Z. Liu, L. Kabela, Y. Pang, and O. Kalinli. (2023) Contextual biasing of named-entities with large language models.

[9] A. Gourav, L. Liu, A. Gandhe, Y. Gu, G. Lan, X. Huang, S. Kalmane, G. Tiwari, D. Filimonov, A. Rastrow *et al.*, "Personalization strategies for end-to-end speech recognition systems," in *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2021, pp. 7348–7352.

[10] D. Le, M. Jain, G. Keren, S. Kim, Y. Shi, J. Mahadeokar, J. Chan, Y. Shangguan, C. Fuegen, O. Kalinli, Y. Saraf, and M. L. Seltzer, "Contextualized streaming end-to-end speech recognition with trie-based deep biasing and shallow fusion," in *Interspeech 2021*. ISCA, 2021, pp. 1772–1776.

[11] G. Sun, X. Zheng, C. Zhang, and P. C. Woodland, "Can contextual biasing remain effective with whisper and gpt-2?" *arXiv preprint arXiv:2306.01942*, 2023.

[12] X. Gong, Z. Zhou, and Y. Qian, "Knowledge Transfer and Distillation from Autoregressive to Non-Autoregessive Speech Recognition," in *Proc. Interspeech 2022*, 2022, pp. 2618–2622.

[13] X. Gong, Y. Lu, Z. Zhou, and Y. Qian, "Layer-wise fast adaptation for end-to-end multi-accent speech recognition," in *Interspeech 2021*. ISCA, 2021, pp. 1274–1278.

[14] X. Gong, Y. Wu, J. Li, S. Liu, R. Zhao, X. Chen, and Y. Qian, "Longfnt: Long-form speech recognition with factorized neural transducer," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[15] A. Salemi, S. Mysore, M. Bendersky, and H. Zamani, "Lamp: When large language models meet personalization," *arXiv preprint arXiv:2304.11406*, 2023.

[16] G. Pundak, T. N. Sainath, R. Prabhavalkar, A. Kannan, and D. Zhao, "Deep context: End-to-end contextual speech recognition," in *2018 IEEE Spoken Language Technology Workshop (SLT)*. IEEE, 2018, pp. 418–425.

[17] F.-J. Chang, J. Liu, M. Radfar, A. Mouchtaris, M. Omologo, A. Rastrow, and S. Kunzmann, "Context-aware transformer transducer for speech recognition," in *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2021, pp. 503–510.

[18] K. M. Sathyendra, T. Muniyappa, F.-J. Chang, J. Liu, J. Su, G. P. Strimel, A. Mouchtaris, and S. Kunzmann, "Contextual Adapters for Personalized Speech Recognition in Neural Transducers," in *ICASSP 2022 - 2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, May 2022, pp. 8537–8541.

[19] X. Gong, W. Wang, H. Shao, X. Chen, and Y. Qian, "Factorized aed: Factorized attention-based encoder-decoder for text-only domain adaptive asr," in *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2023, pp. 1–5.

[20] X. Gong, Y. Wu, J. Li *et al.*, "Advanced long-content speech recognition with factorized neural transducer," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, pp. 1–14, 2024.

[21] X. Yang, W. Kang, Z. Yao, Y. Yang, L. Guo, F. Kuang, L. Lin, and D. Povey, "Promptasr for contextualized asr with controllable style," *arXiv preprint arXiv:2309.07414*, 2023.

[22] H. Futami, E. Tsunoo, Y. Kashiwagi, H. Ogawa, S. Arora, and S. Watanabe, "Phoneme-aware encoding for prefix-tree-based contextual asr," *arXiv preprint arXiv:2312.09582*, 2023.

[23] J. Qiu, L. Huang, B. Li, J. Zhang, L. Lu, and Z. Ma, "Improving large-scale deep biasing with phoneme features and text-only data in streaming transducer," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[24] E. J. Hu, Y. Shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[25] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28 492–28 518.

[26] J. Bai, S. Bai, Y. Chu, Z. Cui, K. Dang, X. Deng, Y. Fan, W. Ge, Y. Han, F. Huang *et al.*, "Qwen technical report," *arXiv preprint arXiv:2309.16609*, 2023.

[27] H. Touvron, T. Lavril, G. Izacard, X. Martinet, M.-A. Lachaux, T. Lacroix, B. Rozière, N. Goyal, E. Hambro, F. Azhar *et al.*, "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[28] P. Lewis, E. Perez, A. Piktus, F. Petroni, V. Karpukhin, N. Goyal, H. Küttler, M. Lewis, W.-t. Yih, T. Rocktäschel *et al.*, "Retrieval-augmented generation for knowledge-intensive nlp tasks," *Advances in Neural Information Processing Systems*, vol. 33, pp. 9459–9474, 2020.

[29] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: an asr corpus based on public domain audio books," in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.

[30] J.-Y. Yao, K.-P. Ning, Z.-H. Liu, M.-N. Ning, and L. Yuan, "Llm lies: Hallucinations are not bugs, but features as adversarial examples," *arXiv preprint arXiv:2310.01469*, 2023.