Using Large Language Model for End-to-End Chinese ASR and NER

Yuang Li[1], Jiawei Yu[2*], Min Zhang[1], Mengxin Ren[1], Yanqing Zhao[1],*
Xiaofeng Zhao[1], Shimin Tao[1], Jinsong Su[2], Hao Yang[1]

[1]Huawei Translation Services Center, Beijing, China
[2]School of Informatics, Xiamen University, China

liyuang3@huawei.com, yujiawei@stu.xmu.edu.cn, jssu@xmu.edu.cn, yanghao30@huawei.com

Abstract

Mapping speech tokens to the same feature space as text tokens has become the paradigm for integrating speech modality into decoder-only large language models (LLMs). An alternative is to use an encoder-decoder architecture that incorporates speech features through cross-attention. In this work, we connect the Whisper encoder with ChatGLM3 and provide in-depth comparisons of these two approaches using Chinese automatic speech recognition (ASR) and named entity recognition (NER) tasks. We evaluate their performance using the F1 score and a fine-grained taxonomy of ASR-NER errors. Our experiments reveal that the encoder-decoder model outperforms the decoder-only model if the context is short, while the decoder-only model benefits from a long context as it fully exploits all layers of the LLM. Additionally, we obtain a state-of-the-art F1 score of 0.805 on the AISHELL-NER test set by using chain-of-thought NER which first infers long-form ASR transcriptions and then predicts NER labels.

Index Terms: speech recognition, named entity recognition, large language model

1. Introduction

Large language models (LLMs) have been shown to perform remarkably on natural language processing tasks, such as question answering, summarization, and machine translation [1]. Various approaches have been proposed to leverage the power of LLMs to multi-modalities. Early works focus on visual understanding tasks. MiniGPT-4 [2] directly feeds visual features into the LLM through a projection layer. LLaMA-Adapter [3] adopts fixed-length trainable vectors as layer-wise prompts which can include visual information. MiniGPT-4 and LLaMA-Adapter are decoder-only models, whereas Flamingo [4] utilizes an encoder-decoder framework where visual representations are merged into the LLM through cross-attention.

Recently, combining speech encoders with LLMs has gained momentum. Among various applications, the automatic speech recognition (ASR) task has received the most attention [5, 6, 7, 8, 9, 10]. Most existing works concentrate on the Adapter layer, which is used for reducing the dimensionality of the speech features and mapping them to the text embedding space. Different types of Adapter layers have been proposed, such as the Attention layer [7], the adaptive CTC downampler [6], and the Convolutional layers [9]. Moreover, a variety of speech encoders (e.g., Whisper encoder [11] and HuBERT [12]) and LLMs (e.g., LLaMA [13] and Vicuna [14]) have been explored in this context. Beyond ASR, the potential of LLMs has been further unleashed by applying them to more

challenging tasks, such as speech translation [15, 16], ASR error correction [17], and multitask speech and audio event understanding [18, 19, 20, 21]. However, these methods adopt a decoder-only architecture that takes speech or audio features as the input to LLMs (similar to miniGPT4), which differs from the standard encoder-decoder architecture of ASR [22]. The only exception is [5], where the HuBERT speech encoder is integrated with the LLaMA model via cross-attention for ASR domain adaptation. In this study, we conducted a thorough comparison of the two types of architectures on Chinese ASR and named entity recognition (NER) tasks, which have received less attention in prior research.

NER from speech is a fundamental task in spoken language understanding, which aims to identify and classify named entities into predefined categories, such as person (PER), location (LOC), and organization (ORG). This task can be performed by either a pipeline system [23] or an end-to-end (E2E) system [24, 25, 26, 27, 28]. A pipeline system consists of an ASR module and a text-based NER model, where the input audio is first transcribed by the ASR module, and then the resulting ASR output is processed by the NER model. In contrast, an E2E system directly extracts entities from speech without depending on the intermediate ASR output, therefore avoiding error propagation. We chose the NER task for our experiments because it requires the LLM to not only learn the mapping from speech features to text tokens, but also comprehend the semantic meaning of the ASR transcription. To analyze the ASR-NER results in detail, we applied the taxonomy of ASR-NER errors proposed in [29] for the pipeline system to our LLM systems.

In this paper, we combined the Whisper encoder [11] with the ChatGLM3-6B [30]. The Whisper is an ASR model trained on a large-scale speech corpus of 680k hours of data and ChatGLM-6B is a bilingual LLM that has outstanding performance in Chinese. We only fine-tuned the LoRA [31] adapters and the connectors between the Whisper encoder and ChatGLM3-6B on AISHELL datasets [32, 33]. Through extensive experiments and visualizations of gate values and attention scores, we uncovered that the encoder-decoder architecture with cross-attention leverages the deeper layers of the LLM and achieves superior performance on the short-form ASR task, while the decoder-only architecture with self-attention exploits all the LLM layers and excels on the long-context ASR and NER tasks. Additionally, we employed chain-of-thought (CoT) NER, which first generates long-form ASR transcriptions and then predicts NER labels. CoT NER attained a state-of-the-art (SOTA) F1 score of 0.805 on the AISHELL-NER test set [33]. According to the taxonomy of ASR-NER results, CoT NER achieved an absolute reduction in omission errors by 7% and an absolute improvement of 9% in correct entities compared to the baseline Conformer model.
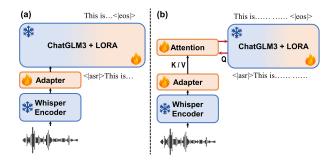
---

* Equal contribution.

Figure 1: *The speech modality is incorporated into the LLM through (a) an adapter (decoder-only), and (b) cross-attention layers (encoder-decoder).*

2. Methodology

2.1. Decoder-only model

One simple way to incorporate the speech modality into the LLM is to use an Adapter layer that bridges the gap between the speech features and the text embeddings. Figure 1 (a) illustrates how the ChatGLM3 model takes the speech tokens from the Whisper encoders, which are transformed by an Adapter, as the input and generates ASR transcriptions in an autoregressive way. In this setting, the speech tokens act as prompts. Since speech features usually have much longer lengths than text features, we downsample the speech features by stacking every five adjacent frames. Afterwards, the speech features are fed into two linear layers as shown in Equation 1.

$$\mathbf{S} = \text{Linear}(\text{ReLU}(\text{Linear}(\mathbf{H}_{whisper}))) \qquad (1)$$

where $\mathbf{H}_{whisper}$ is the downsampled speech features and $\mathbf{S}$ is the speech tokens, the output of the Adapter layer.

2.2. Encoder-decoder model

Figure 1 (b) shows the integration of the Whisper encoder into ChatGLM3, which follows the traditional Transformer [34] architecture where the encoder and the decoder are connected through cross-attention. After each self-attention layer of ChatGLM3, a cross-attention layer is added where the hidden states of text tokens serve as queries and the speech features serve as keys and values. Similar to the approach in [4, 5], we adopt gated cross-attention with learnable gates to control the amount of influence that the speech modality has on the final output. The gate values are initialized to zeros to stabilize training. Unlike previous works, we swap the order of the feedforward and cross-attention layers and apply the gates only to the cross-attention layers (Equation 2, 3, and 4). In our initial experiments, this improves the training stability and the performance of our model, as the speech features are processed by more layers before being fed into the LLM.

$$\mathbf{S} = \text{ReLU}(\text{Linear}(\mathbf{H}_{whisper})) \qquad (2)$$

$$\mathbf{S}^{(i)} = \text{ReLU}(\text{Linear}^{(i)}(\mathbf{S})) \qquad (3)$$

$$\mathbf{H}^{(i)} = \mathbf{H}^{(i)} + \text{Tanh}(g^{(i)}) \odot \text{XATT}^{(i)}(\mathbf{H}^{(i)}, \mathbf{S}^{(i)}, \mathbf{S}^{(i)}) \qquad (4)$$

where the downsampled speech features $\mathbf{H}_{whisper}$ are fed into a single linear layer, resulting in new features $\mathbf{S}$. Then at the i_{th} layer, $\mathbf{S}$ is processed by a Linear layer followed by a cross-attention (XATT) layer scaled by a Tanh gate.

The main differences between the Adapter and the cross-attention architectures can be summarized as follows:

- **Principle**: The Adapter layer enables the LLM to handle speech tokens and text tokens uniformly. The cross-attention layer considers the speech features as a distinct source sequence from the text tokens.
- **Implementation**: The Adapter architecture uses the speech tokens as inputs whereas the cross-attention architecture incorporates the speech features after each layer. Consequently, the Adapter architecture has the advantage of requiring less modification to the source code of the LLM.
- **Parameters and Computation**: Compared to the naive Adapter method, the cross-attention approach introduces a much larger number of parameters, since it involves cross-attention at each layer. Nevertheless, we found that the cross-attention architecture achieves faster training and inference because the cross-attention operates at a lower dimension than the self-attention in the LLM.

2.3. Long-form ASR and CoT NER

We propose a three-phase training schedule to adapt our Whisper-ChatGLM3 models to Chinese ASR and NER. The training involves three tasks including short-form ASR, long-form ASR, and CoT NER. Table 1 illustrates the input formats of these tasks, and the details are provided as follows:

- **Short-form ASR**: We select a variable number of utterances at random and concatenate their features and their corresponding transcriptions. A special token, denoted by $|\textbf{asr}|$, is used to indicate the ASR task. Random concatenation was shown to be effective for the Whisper model [7], as it uses 30-second inputs that are longer than the typical segments in the ASR corpus.
- **Long-form ASR**: The use of historical context from both speech and text modalities was shown to enhance ASR performance [35, 36]. Motivated by this, we investigated the effect of incorporating both historical speech and text information in the recognition of the current utterance. We construct an input sequence by concatenating historical speech tokens S_{i-1} and current speech tokens S_i, separated by a special token γ that marks the beginning of the current utterance. The model produces a long-form transcription with a special token $|\textbf{sep}|$ that indicates the start of the current transcription.
- **CoT NER**: We instruct the LLM to produce long-form transcriptions first and then assign NER labels to the current utterance. This approach can be regarded as a CoT process. It can also be viewed as the combination of pipeline and E2E NER systems as the model accesses both ASR transcriptions and speech features.

Table 1: *Different training tasks. S and T denote speech and text tokens for an utterance respectively. $|\cdot|$ indicates special text tokens. γ is a special audio token that separates the current and historical speech tokens.*

Short-form ASR	$S_i, S_j \|\textbf{asr}\| T_i, T_j$
Long-form ASR	$S_{i-1}, \gamma, S_i \|\textbf{asr}\| T_{i-1} \|\textbf{sep}\| T_i$
CoT NER	$S_{i-1}, \gamma, S_i \|\textbf{asr}\| T_{i-1} \|\textbf{sep}\| T_i \|\textbf{ner}\| \hat{T}_i$

In the first two phases of training, the models are optimized on short-form and long-form ASR tasks respectively. In the third phase, we perform multitask training of long-form ASR

and CoT NER jointly.

2.4. Categories of NER predictions

Referring to the method in [29], we conducted a fine-grained analysis of NER predictions. This allows us to gain a deeper insight into the sources of errors, the benefits of LLMs, and the impact of historical information on NER performance. The categories of NER predictions include:

- **Correct Span**: Entities that match the gold entity tags, meaning the location of the entity is accurately predicted in the ASR transcription.
- **Correct Entity**: A subset of correct span. All the tokens within the entity are predicted accurately. This is the only category that is considered correct by the standard F1 score.
- **Error Span**: Entities that deviate from the gold entity tags.
- **Replacement**: A subset of error span. The entity type is predicted incorrectly.
- **Omission**: A subset of error span. The entity in the gold transcript is missing.

3. Experimental Setups

3.1. Dataset and metrics

In our experiments, we used the AISHELL-NER dataset [33], an annotated version of the AISHELL-1 [32] dataset, which consists of 170 hours of Chinese speech data. The training set of AISHELL-NER contains about 120,000 sentences, and the test set has 7176 sentences. The dataset annotates three types of named entities: person (PER), location (LOC), and organization (ORG), using special symbols: "[·]" for PER, "(·)" for LOC, and "< · >" for ORG. We formulated NER as a sequence generation task, where the model predicts the entity symbols along with the ASR transcriptions. We measured the performance of our model on the test set, which has 900 PER, 1330 LOC, and 1165 ORG entities, using character error rate (CER) for ASR and F1 score for NER.

3.2. Model architecture

We employed the Conformer-based [33] E2E system as a baseline model that did not incorporate the LLM. Our systems utilized the encoder of Whisper-large-v2 [11] as the speech encoder and ChatGLM3-6B [30] as the base LLM. We froze the Whisper encoder and applied efficient fine-tuning of ChatGLM3 with LoRA [31], enabling the LLM to adapt to the domain of the AISHELL dataset and comprehend ASR and NER tasks. The fine-tuning of the LLM avoided reliance on the parameter in the Adapter to accomplish the task that the language model should handle. For LoRA, we set the rank to 32 and applied it to both the attention and feedforward layers, resulting in 15 million parameters. For the decoder-only architecture with the Adapter layer, the Adapter had 43 million parameters with two linear layers that mapped the feature from the dimension of 6400 (1280 × 5) to 4096. For the encoder-decoder architecture with the gated cross-attention, the speech features were first projected to the dimension of 4096 and then reduced to 1024 at each layer. The cross-attention layer had eight heads, and the multi-head dot-product attention was computed at a low dimension of 1024 and then projected back to 4096 after the attention layer. The linear layers and the cross-attention layers had 437 million parameters in total.

3.3. Training schedule

The model underwent three phases of training. In the first phase, the model was trained on the short-form ASR task for 40 epochs, using a learning rate of 5e-5 and a batch size of 64. In the second phase, the model was fine-tuned on the long-form ASR task for 20 epochs, using a learning rate of 3e-5. The number of historical utterances was randomly sampled for each training example. The concatenated audio was either padded or cropped to 30 seconds, depending on whether it was shorter or longer than that duration. In the third phase, the model was further trained for 20 epochs, using a learning rate of 3e-5. To activate the model's NER capability without compromising its ASR capability, we optimized the model for the long-form ASR task with a probability of 30% and for the NER task with a probability of 70%.

4. Experimental Results

4.1. ASR results

As demonstrated in Table 2, our systems achieved a substantial improvement over the Conformer baseline, with a relative CER reduction of 19.7%. The encoder-decoder architecture outperformed the decoder-only architecture when no historical context was available. Incorporating contextual information led to consistent ASR accuracy enhancement for both architectures. Nevertheless, the decoder-only architecture benefited more from historical information than the encoder-decoder architecture, as evidenced by the relative CER reduction of 14.4% versus 7.0%, respectively. This suggests that the decoder-only architecture can leverage historical information more effectively.

Table 2: *The ASR performance of two systems with or without history context measured by **CER (%)**.*

	Short-form	Long-form
Conformer	4.83	/
Decoder-only	4.02	**3.44**
Encoder-decoder	**3.88**	3.61

4.2. NER results

As shown in Table 3, the decoder-only and encoder-decoder models achieved the highest F1 scores of 0.805 and 0.789 respectively when using historical context, which were significantly higher than the Conformer model's score of 0.743. Regarding the types of named entities, all models performed

Table 3: *The NER performance of two systems with different historical context lengths (His.) measured by **F1 score**.*

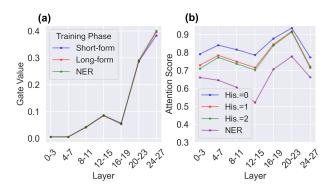| Method	His.	PER	LOC	ORG	Total
Conformer	0	0.561	0.832	0.778	0.743
Decoder-only	0	0.601	0.868	0.812	0.778
	1	0.631	0.881	0.837	0.799
	2	0.635	0.878	0.853	**0.805**
Encoder-decoder	0	0.596	0.879	0.813	0.782
	1	0.594	0.885	0.832	**0.789**
	2	0.600	0.878	0.831	0.788

Figure 2: *(a) Gate values of the cross-attention across different layers for the encoder-decoder architecture during different training phases. (b) The attention scores correspond to the speech tokens across different layers for the decoder-only architecture with different historical (His.) context lengths and tasks (i.e., ASR or NER).*

poorly on names. This is mainly because Chinese names have many variations and homophones that can create confusion and ambiguity for the ASR system. The trend of NER performance when using historical context was similar to that of ASR, in that the decoder-only model's F1 score increased steadily, while the encoder-decoder model's F1 score reached its peak with only one historical utterance.

4.3. Visualizations

Figure 2 provides visualizations to better understand the results in the previous sections and the differences between the two architectures. Figure 2 (a) illustrates the gate values across different layers for the cross-attention layer. The gate values remained roughly unchanged across different training phases and deep layers were assigned significantly larger gate values than shallow layers.

For the decoder-only architecture, the influence of speech modality is controlled by self-attention. We calculated the average attention scores for the speech tokens on the AISHELL test set (Table 2 (b)). The attention score on the speech tokens decreased with longer context as more attention was given to the text tokens. For the NER task, we first generated ASR transcriptions that provided rich semantic information for NER, resulting in the lowest attention scores for speech tokens. The attention scores are consistently above 0.5 across different layers, indicating that the speech features are important for ASR and NER. Moreover, the attention scores for the deep and shallow layers are similar, implying that all LLM layers were fully utilized.

Based on the previous observations, we can gain insights into why the decoder-only model performed better with a longer context. First, the decoder-only model better utilized the pretrained parameters inside the LLM by incorporating speech features at shallow layers. Additionally, the decoder-only model can dynamically adjust the importance of text tokens according to the context length and the nature of the task, while the encoder-decoder architecture treated the speech modality in a static manner with similar weights under different scenarios.

4.4. NER taxonomy

Table 4 presents the taxonomy of NER predictions. It is evident that all models have a low rate of replacement errors, whereas omission errors are more prevalent. These errors mainly stemmed from PER and ORG, which are often rare entities. One of the main advantages of using the LLM is that it reduced the omission error by almost 50% compared to the Conformer baseline. With the LLM, more than 90% of entities can be labeled accurately, but among them, 10% have erroneous ASR results which are mostly substitution errors for personal names. These results prove that with the aid of the LLM, the model can better identify rare named entities, but it remains difficult to precisely recognize the tokens within the Chinese names without prior knowledge.

The impact of historical context was also investigated, and it can be observed that for the decoder-only model, the addition of one historical context enhanced the percentage of the correct span by 1.3% and the percentage of the correct entity by 2.0%. This indicates that a longer context not only improved the model's ability to locate the entity but also promoted ASR accuracy as it ensured that the predicted entities were more coherent across utterances. For the encoder-decoder model, adding one historical context only increased the percentage of the correct span by 1.1% and the percentage of the correct entity by 0.8%.

Table 4: *The classification of the entity predictions into different categories: Correct Span (Cor. Span), Correct Entity (Cor. Ent.), Error Span (Err. Span), Replacement (Rep.), and Omission (Omi.). The numbers are **percentages (%)**.*

Method	His.	Cor. Span	Cor. Ent.	Err. Span	Rep.	Omi.
Conformer	0	82.3	71.0	17.7	1.5	11.5
Decoder-only	0	88.7	77.1	11.3	1.3	5.6
	1	90.0	79.1	10.0	1.1	5.2
	2	**90.7**	**79.9**	**9.3**	**1.0**	**4.5**
Encoder-decoder	0	88.4	77.4	11.6	1.3	5.8
	1	**89.5**	**78.2**	**10.5**	**1.2**	**5.2**
	2	88.9	78.0	11.1	1.4	5.5

5. Conclusion

In this paper, we explored combining a speech encoder with an LLM for Chinese ASR and NER tasks using two different architectures: decoder-only and encoder-decoder. Under utterance-level evaluation, both architectures achieved significant improvements over the baseline Conformer model. We further compared these two approaches in terms of their capability to utilize long-form historical information. The long-form evaluations indicate that the decoder-only model benefited more from incorporating historical context than the encoder-decoder model. To explain this phenomenon, we conducted a comprehensive analysis of the gate values and attention scores, which revealed the superior ability of the decoder-only model to adjust the importance of speech and text modalities dynamically. For future works, we intend to conduct larger-scale experiments and evaluate our systems on more tasks. We hypothesize that the encoder-decoder model may have an advantage over the decoder-only approach in tasks where audio features are crucial, such as audio event detection. Therefore, we plan to investigate the potential of combining the two approaches to achieve superior performance on a wide range of tasks that require both high-level semantic and fine-grained acoustic information.

6. References

[1] OpenAI, "GPT-4 technical report," *arXiv preprint arXiv:2303.08774*, 2023.

[2] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.

[3] R. Zhang, J. Han, A. Zhou, X. Hu, S. Yan, P. Lu, H. Li, P. Gao, and Y. Qiao, "LLaMA-adapter: Efficient fine-tuning of language models with zero-init attention," *arXiv preprint arXiv:2303.16199*, 2023.

[4] J.-B. Alayrac, J. Donahue, P. Luc, A. Miech, I. Barr, Y. Hasson, K. Lenc, A. Mensch, K. Millican, M. Reynolds *et al.*, "Flamingo: a visual language model for few-shot learning," *Proc. NeurIPS*, vol. 35, pp. 23 716–23 736, 2022.

[5] Y. Li, Y. Wu, J. Li, and S. Liu, "Prompting large language models for zero-shot domain adaptation in speech recognition," in *Proc. ASRU*, 2023.

[6] S. Ling, Y. Hu, S. Qian, G. Ye, Y. Qian, Y. Gong, E. Lin, and M. Zeng, "Adapting large language model with speech for fully formatted end-to-end speech recognition," *arXiv preprint arXiv:2307.08234*, 2023.

[7] W. Yu, C. Tang, G. Sun, X. Chen, T. Tan, W. Li, L. Lu, Z. Ma, and C. Zhang, "Connecting speech encoder and large language model for asr," *arXiv preprint arXiv:2309.13963*, 2023.

[8] J. Wu, Y. Gaur, Z. Chen, L. Zhou, Y. Zhu, T. Wang, J. Li, S. Liu, B. Ren, L. Liu, and Y. Wu, "On decoder-only architecture for speech-to-text and large language model integration," *arXiv preprint arXiv:2307.03917*, 2023.

[9] Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli, C. Fuegen, and M. Seltzer, "Prompting large language models with speech recognition abilities," *arXiv preprint arXiv:2307.11795*, 2023.

[10] Y. Hono, K. Mitsuda, T. Zhao, K. Mitsui, T. Wakatsuki, and K. Sawada, "An integration of pre-trained speech and language models for end-to-end speech recognition," *arXiv preprint arXiv:2312.03668*, 2023.

[11] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," *arXiv preprint arXiv:2212.04356*, 2022.

[12] W.-N. Hsu, B. Bolte, Y.-H. H. Tsai, K. Lakhotia, R. Salakhutdinov, and A. Mohamed, "HuBERT: Self-supervised speech representation learning by masked prediction of hidden units," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, vol. 29, pp. 3451–3460, 2021.

[13] 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.

[14] W.-L. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, J. E. Gonzalez, I. Stoica, and E. P. Xing, "Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality," March 2023. [Online]. Available: https://lmsys.org/blog/2023-03-30-vicuna/

[15] Z. Chen, H. Huang, A. Andrusenko, O. Hrinchuk, K. C. Puvvada, J. Li, S. Ghosh, J. Balam, and B. Ginsburg, "Salm: Speech-augmented language model with in-context learning for speech recognition and translation," *arXiv preprint arXiv:2310.09424*, 2023.

[16] Z. Huang, R. Ye, T. Ko, Q. Dong, S. Cheng, M. Wang, and H. Li, "Speech translation with large language models: An industrial practice," *arXiv preprint arXiv:2312.13585*, 2023.

[17] S. Radhakrishnan, C.-H. Yang, S. Khan, R. Kumar, N. Kiani, D. Gomez-Cabrero, and J. Tegnér, "Whispering llama: A cross-modal generative error correction framework for speech recognition," in *Proc. EMNLP*, 2023.

[18] Y. Gong, A. H. Liu, H. Luo, L. Karlinsky, and J. Glass, "Joint audio and speech understanding," in *Proc. ASRU*, 2023.

[19] J. Liang, X. Liu, W. Wang, M. D. Plumbley, H. Phan, and E. Benetos, "Acoustic prompt tuning: Empowering large language models with audition capabilities," *arXiv preprint arXiv:2312.00249*, 2023.

[20] 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," *arXiv preprint arXiv:2310.00230*, 2023.

[21] 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," *arXiv preprint arXiv:2311.07919*, 2023.

[22] J. Chorowski, D. Bahdanau, D. Serdyuk, K. Cho, and Y. Bengio, "Attention-based models for speech recognition," in *Proc. NeurIPS*, Dec. 2015, pp. 577–585.

[23] M. A. B. Jannet, O. Galibert, M. Adda-Decker, and S. Rosset, "Investigating the effect of ASR tuning on named entity recognition," in *Proc. Interspeech*, F. Lacerda, Ed., 2017.

[24] M. Gaido, S. Papi, M. Negri, and M. Turchi, "Joint speech translation and named entity recognition," *CoRR*, 2022.

[25] D. Serdyuk, Y. Wang, C. Fuegen, A. Kumar, B. Liu, and Y. Bengio, "Towards end-to-end spoken language understanding," in *Proc. ICASSP*, 2018.

[26] H. Yadav, S. Ghosh, Y. Yu, and R. R. Shah, "End-to-end named entity recognition from english speech," in *Proc. Interspeech*, H. Meng, B. Xu, and T. F. Zheng, Eds., 2020.

[27] S. Mdhaffar, J. Duret, T. Parcollet, and Y. Estève, "End-to-end model for named entity recognition from speech without paired training data," in *Proc. Interspeech*, H. Ko and J. H. L. Hansen, Eds., 2022.

[28] S. Arora, S. Dalmia, B. Yan, F. Metze, A. W. Black, and S. Watanabe, "Token-level sequence labeling for spoken language understanding using compositional end-to-end models," in *Proc. EMNLP Findings*, Y. Goldberg, Z. Kozareva, and Y. Zhang, Eds., 2022.

[29] P. Szymanski, L. Augustyniak, M. Morzy, A. Szymczak, K. Surdyk, and P. Zelasko, "Why aren't we NER yet? artifacts of ASR errors in named entity recognition in spontaneous speech transcripts," in *Proc. ACL*, A. Rogers, J. L. Boyd-Graber, and N. Okazaki, Eds., 2023.

[30] A. Zeng, X. Liu, Z. Du, Z. Wang, H. Lai, M. Ding, Z. Yang, Y. Xu, W. Zheng, X. Xia, W. L. Tam, Z. Ma, Y. Xue, J. Zhai, W. Chen, P. Zhang, Y. Dong, and J. Tang, "Glm-130b: An open bilingual pre-trained model," *Proc. ICLR*, 2023.

[31] E. J. Hu, yelong shen, P. Wallis, Z. Allen-Zhu, Y. Li, S. Wang, L. Wang, and W. Chen, "LoRA: Low-rank adaptation of large language models," in *Proc. ICLR*, 2022.

[32] H. Bu, J. Du, X. Na, B. Wu, and H. Zheng, "Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline," in *Proc. O-COCOSDA*, 2017, pp. 1–5.

[33] B. Chen, G. Xu, X. Wang, P. Xie, M. Zhang, and F. Huang, "AISHELL-NER: Named entity recognition from chinese speech," in *Proc. ICASSP*. IEEE, 2022.

[34] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proc. NeurIPS*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30, 2017.

[35] A. Schwarz, I. Sklyar, and S. Wiesler, "Improving RNN-T ASR accuracy using context audio," in *Proc. Interspeech*, Brno, Czech Republic, Sep. 2021.

[36] T. Hori, N. Moritz, C. Hori, and J. L. Roux, "Advanced long-context end-to-end speech recognition using context-expanded transformers," in *Proc. Interspeech*, Brno, Czech Republic, Sep. 2021.