

CONNECTING SPEECH ENCODER AND LARGE LANGUAGE MODEL FOR ASR

Wenyi Yu¹, Changli Tang¹, Guangzhi Sun¹, Xianzhao Chen²,
Tian Tan², Wei Li², Lu Lu², Zejun Ma², Chao Zhang^{1,*}

¹ Department of Electronic Engineering, Tsinghua University, ² ByteDance

ywy22@mails.tsinghua.edu.cn; cz277@tsinghua.edu.cn

ABSTRACT

The impressive capability and versatility of large language models (LLMs) have aroused increasing attention in automatic speech recognition (ASR), with several pioneering studies attempting to build integrated ASR models by connecting a speech encoder with an LLM. This paper presents a comparative study of three commonly used structures as connectors, including fully connected layers, multi-head cross-attention, and Q-Former. Speech encoders from the Whisper model series as well as LLMs from the Vicuna model series with different model sizes were studied. Experiments were performed on the commonly used LibriSpeech, Common Voice, and GigaSpeech datasets, where the LLMs with Q-Formers demonstrated consistent and considerable word error rate (WER) reductions over LLMs with other connector structures. Q-Former-based LLMs can generalise well to out-of-domain datasets, where 12% relative WER reductions over the Whisper baseline ASR model were achieved on the Eval2000 test set without using any in-domain training data from Switchboard. Moreover, a novel segment-level Q-Former is proposed to enable LLMs to recognise speech segments with a duration exceeding the limitation of the encoders, which results in 17% relative WER reductions over other connector structures on 90-second-long speech data.

Index Terms— Large language model, automatic speech recognition, Q-Former, long-form speech

1. INTRODUCTION

Large language models (LLMs) [1–6] with rich knowledge and the ability to solve novel and complex tasks have revolutionised the field of natural language processing. More recently, significant attention has been drawn to enable LLMs to handle speech inputs [7–17]. In addition to pipeline-based methods in which the LLM serve as a controller to manage a set of functional models [7, 8], two categories of approaches have been developed. The first category of approaches discretises speech inputs and embeds the derived speech tokens into a vector space shared with the text tokens, then the LLMs are fine-tuned to fit into this new token space [9, 10]. The other category of approaches directly connects a speech encoder with an LLM using a connector that aligns the speech encoders with the LLMs [11–15]. This paper focuses on the second category of approaches.

When aligning the speech encoder output and LLM input spaces, the choice of the connector is of vital importance, and should meet the following requirements. First, the connector should be able to retain as much information from the speech inputs as possible, since it determines the amount of information that LLMs can receive from the speech. Second, as the computation and storage costs of LLMs

increase considerably when processing long input sequences, the connector should be able to achieve efficient and effective information compression to reduce the lengths of the LLM input sequences.

Focusing on automatic speech recognition (ASR) to show the ability of LLMs to recognise speech, this paper studies different structures of connectors that integrate LLMs with speech encoders. Specifically, an end-to-end ASR system was constructed by connecting a Whisper model encoder [18] with a Vicuna LLM [5]. Three types of connectors were compared in this paper, including the fully connected layers, multi-head cross-attention [19] and Q-Former [20]. To bypass the input length limitation of the pre-trained speech encoders and enable the LLM to process long speech inputs, a novel segment-level Q-Former connector is proposed. Experiments were conducted based on a training set with $\sim 4,000$ hours data, and the LLMs with Q-Former connectors consistently outperform strong Whisper baseline ASR systems on the in-domain datasets, and can achieve over 12% relative word error rate (WER) reductions on the out-of-domain Eval2000 test set from the Switchboard corpus. The influence of the number of connector output tokens and model size are studied. Moreover, the proposed segment-level Q-Former structure achieved obvious WER reductions on long speech inputs, when compared with other connectors.

The rest paper is organised as follows. Sec. 2 summarises the work related to multimodal LLMs. Secs. 3 and 4 introduce the three connectors to compare and the proposed segment-level Q-Former. The experimental setup and results are presented in Secs. 5 and 6, followed by conclusions.

2. RELATED WORK

To enable LLMs to perform both speech perception and generation, SpeechGPT [9] and AudioPaLM [10] augment the vocabularies of LLaMA [4] and PaLM [3] LLMs with discrete speech tokens extracted by HuBERT [21] and W2v-BERT [22] or USM [23] speech encoders respectively. Regarding the approaches to connect multimodal encoders to LLMs, X-LLM [11] interfaced ChatGLM with audio and visual encoders. [14] and [13] connect speech encoders with reduced frame rates to LLMs, and achieve integrated multilingual ASR and speech translation respectively. Moreover, LLMs can also be prompt for domain adaptation [15] and uncertainty estimation [17] of the ASR results.

Several works studied visual LLMs [19, 20, 24–30]. Following BLIP-2 [20], InstructBLIP [26] and Video-LLaMA [27] introduced Q-Former as the module connector. Alternative connectors were also investigated in [19, 24, 28, 29]. Regarding audio and music LLMs, the reasoning ability based on audio is studied [31]. MU-LLaMA [32] showed outstanding music understanding abilities.

* Corresponding author

3. MODULE CONNECTOR

As shown in Fig. 1, the proposed ASR model consists of three modules: a frozen speech encoder, a trainable module connector and a frozen LLM. This section introduces three connectors, including fully connected layers, multi-head cross-attention and Q-Former.

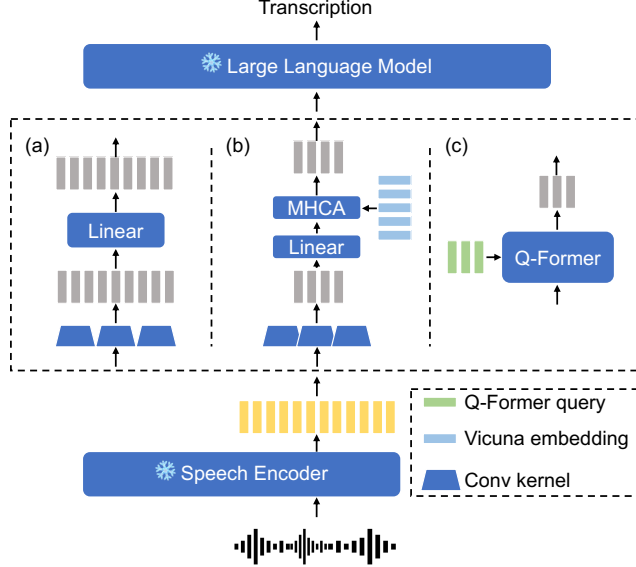


Fig. 1. Illustration of integrating a speech encoder and an LLM into an ASR system with a module connector of: (a) fully connected layers, (b) multi-head cross-attention, and (c) Q-Former.

For clarity, some basic notations are defined as follows: $\mathbf{X} \in \mathbb{R}^{n_x \times d_x}$ denotes the speech features obtained from the speech encoder, and the module connector compresses \mathbf{X} into $\mathbf{T}_{\text{speech}} \in \mathbb{R}^{n_t \times d_t}$ which are input to the LLM to produce ASR transcriptions. $\mathbf{H} \in \mathbb{R}^{n_h \times d_h}$ denotes the hidden states in connectors while n and d are the numbers of vectors and hidden dimensions respectively.

3.1. Fully connected layers

To compress the length of speech features, m adjacent frames $\mathbf{x}_i, \mathbf{x}_{i+1}, \dots, \mathbf{x}_{i+m-1}$ are stacked into $\mathbf{h}_i \in \mathbb{R}^{m \times d_x}$. Then two Linear(\cdot) layers with ReLU(\cdot) in between are introduced as follows:

$$\mathbf{T}_{\text{speech}} = \text{Linear}(\text{ReLU}(\text{Linear}(\mathbf{H}))), \quad (1)$$

where \mathbf{H} consists of \mathbf{h}_i of a batch of samples. Actually, the vector stacking operation together with the first linear layer works the same as a 1-dimensional (-d) convolutional layer, Conv1d(\cdot).

3.2. Multi-head cross-attention

To bridge the gap between the multi-modal encoder output features \mathbf{X} and LLM input textual features $\mathbf{T}_{\text{speech}}$, a multi-head attention layer [33] denoted as MultiHead(Query, Key, Value) is used in the multi-head cross-attention approach to align the two feature spaces [19]. First, a Conv1d(\cdot) layer reduces the length of the speech input by a rate of s . Then the hidden states \mathbf{H} are converted to $\mathbf{T}_{\text{speech}}$ based on the textual embeddings \mathbf{E} using MultiHead(\cdot). That is,

$$\mathbf{H} = \text{Linear}(\text{Conv1d}(\mathbf{X})) \quad (2)$$

$$\mathbf{T}_{\text{speech}} = \text{MultiHead}(\mathbf{H}, \mathbf{E}, \mathbf{E}). \quad (3)$$

3.3. Q-Former

Q-Former [20] is a Transformer-based module converting variable-length input sequences into fixed-length output query representations. It was initially proposed for visual-text modality alignment, and here is applied to audio-text alignment. In each Q-Former block, trainable query embeddings $\mathbf{Q} \in \mathbb{R}^{n_q \times d_q}$ interact with the input features \mathbf{X} through multi-head self-attention and cross-attention layers. Specifically, Q-Former, denoted as $\text{QF}(\mathbf{Q}, \mathbf{X})$, in this work consists of two Transformer decoder blocks [33] with the causal attention masks removed. Here \mathbf{Q} is used as the decoder inputs and \mathbf{X} as the encoder outputs in the standard Transformer.

4. SEGMENT-LEVEL Q-FORMER

Transformer-based speech encoders can have limitations on the input sequence duration [18]. To enable LLMs to process with longer speech inputs, the whole sequence can be split into several shorter segments to transform by the speech encoder separately. Such segments can be concatenated to reform a single sequence at either the input or output end of Q-Former. In this paper, the structure of segment-level Q-Former (seg-QF) shown in Fig. 2 is proposed, which uses a Q-Former to transform each encoder output segment simultaneously and concatenates their fixed-length output token sequences before feeding into the LLM. Compared to performing the concatenation at the Q-Former input end and producing a fixed number of n_q output tokens, seg-QF allows varying the number of output tokens $N \times n_q$ according to the number of segments N , which is more suitable for speech inputs with variable lengths in a wide range. Note the trainable query embeddings \mathbf{Q} and Q-Former layers are shared among all the segments, and seg-QF can be initialised with a pre-trained standard Q-Former.

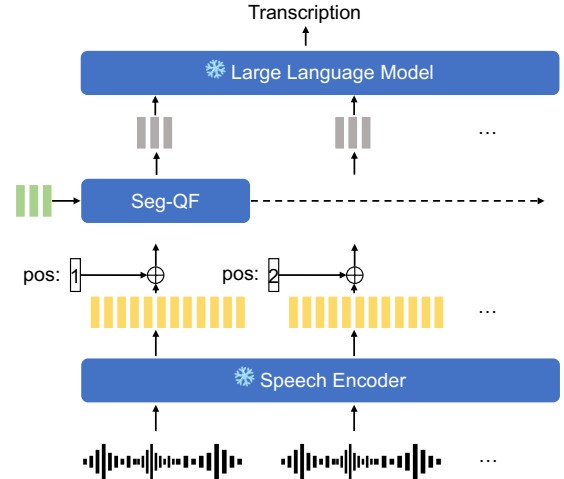


Fig. 2. The model structure of segment-level Q-Former (seg-QF). The integers in rectangles are segment-level positional encodings.

Despite that relative positions of the frames are provided by the speech encoder within each segment $\mathbf{S}_i \in \mathbb{R}^{n_x \times d_x}$, Seg-QF is not aware of their absolute positions in the whole input sequence. To inform Seg-QF with such information, segment-level position embeddings $\mathbf{p}_i \in \mathbb{R}^{d_x}$ are added to \mathbf{X} , as shown in Fig. 2. Specifically,

$$\mathbf{T}_{\text{speech}} = [\text{QF}(\mathbf{Q}, \mathbf{S}_i \oplus \mathbf{p}_i)]_{i=1}^N, \quad (4)$$

where $\mathbf{S}_i \oplus \mathbf{q}_i$ means adding \mathbf{q}_i to each row of \mathbf{S}_i .

5. EXPERIMENTAL SETUP

5.1. Data specifications

Experiments were conducted on three setups. Specifically, models in Sections 6.1-6.4 were trained on LibriSpeech [34] 960h dataset. Models in Section 6.5 were trained on around 4,000 hours of data including LibriSpeech 960h, Common Voice 7.0 (English) [35] and GigaSpeech [36] subset M. Models in Section 6.6 were finetuned on LibriSpeech train-clean-100 subset. While the test sets of the aforementioned data were used for in-domain evaluation, the Eval2000 set was used for out-of-domain evaluation.

5.2. Model specifications

The Vicuna LLMs [5] and the Speech encoders from the Whisper models were used as the decoders and encoders [18], and were both frozen in training. In Section 6.4, speech encoders with different sizes were compared, including those from Whisper base, medium and large-v2. LLMs including Vicuna 7B and Vicuna 13B were compared. Based on Table 2, the best-performing setup with Whisper large-v2 encoder and Vicuna 13B were used in other sections.

All the models in Sections 6.1–6.5 were trained for 90k steps with a batch size of 24 with NVidia A100 80GB GPUs. Models in Section 6.6 were initialised with pre-trained Q-Former models or fully connected layers and trained for 10k steps with a batch size of 8. Sinusoid encoding matrix [33] was used as the segment-level position embeddings. Checkpoints with the highest validation set accuracy obtained with teacher forcing were selected as the final models.

6. EXPERIMENTAL RESULTS

6.1. Random audio concatenation in training

The Whisper encoder is built to have a fixed input window of 30 seconds, and zero vectors are padded to the input to increase its sequence length to match the window size. When connecting the Whisper encoder to LLMs using Q-Former and this padding strategy, high deletion errors are often produced for long inputs since the attention mechanisms of Q-Former are trained to ignore the ending elements in the sequence, which are often padded zeros in the training samples. To resolve this issue, a random concatenation strategy is applied, which is similar to that used in [37]. Each input utterance in a training mini-batch is concatenated with a number of utterances randomly selected from the whole training set. The random concatenation continues as long as the total length of the concatenated utterance does not exceed a pre-set upper limit of T seconds. The values of T for the training samples are set to follow a uniform distribution of $\mathcal{U}[0, 30]$ seconds. From the results in Table 1, it is demonstrated the strategy can considerably reduce deletion errors and improve the WERs, and therefore it is always used in the rest of the experiments.

6.2. Comparisons with different connectors

A desirable connector should be able to extract all useful information from the input without causing obvious increases in computation and storage. Regarding ASR, WERs can be used as an indicator to reflect the quality of the information extracted by the connector. Besides having a reasonable amount of model parameters, the connector is also expected to produce a reduced number of output tokens when required since the number of LLM input tokens is a key factor influencing both LLM computation cost and memory usage.

Random Concat.	LibriSpeech			
	test-clean %WER	%Del	test-other %WER	%Del
✗	3.72	1.38	6.47	1.32
✓	2.28	0.30	5.20	0.50

Table 1. Results with or without using the random concatenation training strategy. Both %WERs and deletion error rates (%Del) are shown. The ASR is a Vicuna 13B LLM with a Q-Former connector.

Model	#Tokens	#Params	LibriSpeech	
			test-clean	test-other
FC	75	24.6M	3.00	6.70
FC	300	23.6M	2.29	5.44
CA	75	133.4M	3.22	7.54
QF	60	24.5M	2.33	5.43
QF	80	24.5M	2.28	5.20

Table 2. %WERs of LLMs with different connectors. FC, CA, and QF refer to fully connected layers, multi-head cross-attention, and Q-Former connectors respectively. #Tokens and #Params are the numbers of output tokens and model parameters of the connectors.

In Table 2, three connectors, including fully connected layers, multi-head cross-attention and Q-Former, are compared. Q-Former results in the lowest WERs on both test sets by producing only 80 output tokens. Although fully connected layers with 300 output tokens (with $m = 5$ in Sec. 3.1) can achieve similar WERs to Q-Former, it requires much more calculations and memory usage. The WERs produced by the fully connected layers connector with 75 output tokens (with $m = 20$) are obviously worse. The multi-head cross-attention connector [19] has ~ 133.4 million (M) model parameters, which are ~ 6 times more than the others, but still produced the worst WERs. As a result, Q-Former is used in the rest of the study.

6.3. Trainable queries of Q-Former

Recapping Section 3.3, the number of trainable queries used in the Q-Former determines the number of its output tokens. This section compares Q-Formers with different numbers of queries. Table 3 shows that increasing the number of queries to 80 can considerably reduce WERs, which implies that using ~ 80 tokens can retain sufficient information for ASR for inputs with less than 30 seconds (similar WER changes are found when only considering the utterances whose durations are close to 30 seconds), which reveals the strong information compression ability of Q-Former.

#Queries	LibriSpeech	
	test-clean	test-other
40	2.43	5.72
60	2.33	5.43
80	2.28	5.20
160	2.26	5.29
300	2.19	5.25

Table 3. Changes in %WERs by varying the number of trainable queries (equals to the number of output tokens) in Q-Former.

6.4. The sizes of LLMs and speech encoders

In this section, models with LLMs and speech encoders of different sizes are compared in this section. Q-Former connectors with 80 trainable queries are used for all models. From the results Table 4, increasing the sizes of both speech encoder and LLM can result in lower WERs, which is in line with expectations. Doubling the size of the speech encoder (Whisper medium to Whisper large) reduces the WERs more obviously than doubling the size of LLMs, which indicates that a stronger speech encoder is more important for ASR, which does not require much content understanding ability.

Model scale		LibriSpeech	
Encoder	LLM	test-clean	test-other
Whisper base	Vicuna 13B	3.48	9.83
Whisper medium	Vicuna 13B	2.35	5.66
Whisper large-v2	Vicuna 13B	2.28	5.20
Whisper large-v2	Vicuna 7B	2.30	5.48

Table 4. %WERs with different speech encoders and LLMs.

6.5. Results with large-scale training set

This section verifies the performance of the best-performing model configuration by training on a large-scale dataset with 4,000 hours of speech. A Vicuna 13B LLM is connected with a Whisper large-v2 encoder using a Q-Former connector with 80 trainable queries. The results in Table 5 show that the proposed model outperformed Whisper large-v2 by an average of $\sim 11\%$ relative WER reduction on the three in-domain test sets. It also generalises well to out-of-domain data by achieving a $\sim 12\%$ relative WER reduction over Whisper large-v2 on the Eval2000 test sets. These results verify the superior performance of the proposed model in ASR.

Test set	Model	
	Whisper large-v2	ours
LibriSpeech test-clean	2.5 (2.7)	2.1
LibriSpeech test-other	5.2 (5.2)	5.0
Common Voice 7.0 dev	7.9 (-)	7.1
Common Voice 7.0 test	9.8 (-)	8.2
GigaSpeech dev	10.3 (-)	9.0
GigaSpeech test	10.0 (-)	9.2
CallHome test	18.9 (17.6)	15.5
Switchboard test	15.8 (13.8)	12.1

Table 5. %WERs of models trained on the large-scale dataset. Numbers in brackets are the official Whisper %WERs reported in [18].

6.6. Results of segment-level Q-Former

To evaluate the models when recognising speech exceeding the duration limitation of the speech encoder, concatenated long-form test sets were built based on LibriSpeech test-clean and test-other test sets respectively. Utterances of the same chapter were concatenated sequentially to form sequences with a duration limit of 60, 90, or 120 seconds. The models were trained on LibriSpeech 100h subset using random concatenation with T sampled from $\mathcal{U}[0, 90]$.

Different models were compared in Table 6. Without finetuning on data longer than 30 seconds, QF and FC cannot generalise to recognise longer speech. After finetuning, seg-QF outperformed both QF and FC considerably and showed some generalisation to

Concat. Test Set	Model					
	FC†	QF†	FC	QF	Seg-QF	Seg-QF*
test-clean	3.00	2.28	2.89	2.32	2.35	2.19
$t \leq 60$	58.18	61.99	3.43	3.83	3.11	2.68
$t \leq 90$	70.64	75.45	4.98	5.40	3.81	3.24
$t \leq 120$	75.48	76.81	19.14	26.91	17.95	13.60
test-other	6.70	5.20	6.68	5.40	5.37	5.20
$t \leq 60$	62.19	63.49	7.58	7.81	6.43	5.13
$t \leq 90$	74.27	76.03	9.03	11.78	8.07	6.17
$t \leq 120$	78.75	76.14	24.55	32.24	22.25	19.63

Table 6. %WERs on the concatenated test sets. Results based on the LibriSpeech test clean and test other test sets are shown in Row 1-4 and Row 5-8 respectively, and Row 1 and Row 5 are the results with the original segmentation. Each $t \leq T_{\text{test}}$ test set is obtained by concatenating either test clean or test other utterances based on their orders that appeared in the original audiobooks to up to T_{test} seconds. Models with † were not fine-tuned on concatenated training data. Pre-trained checkpoints for Seg-QF* were QF in Table 5, and were FC in Row 1 and QF in Row 5 of Table 2 for other models.

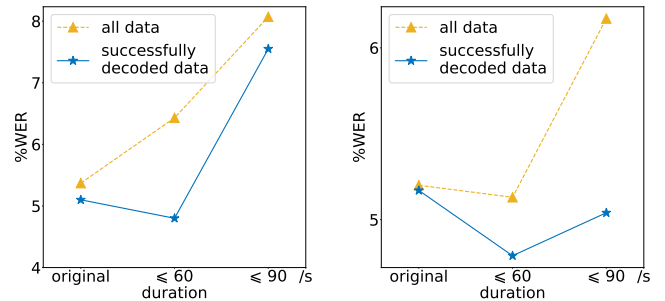


Fig. 3. %WERs of Seg-QF (left) and Seg-QF* (right) in Tab. 6 calculated on all data and successfully decoded data in the test-other set. The x-axis denotes the duration of the concatenated audio.

120-second-long speech inputs. From Table 6, WERs degrade with longer speech inputs. More extreme cases, such as repeated outputs and large chunks of deletions, were observed in case studies when feeding long speech inputs. The WERs for all utterances and those successfully decoded in LibriSpeech test-other are plotted in Fig. 3 for Seg-QF and Seg-QF*. It shows that the contexts in long speech inputs help ASR to reduce WERs if being decoded successfully. However, the overall WERs on the full test-other set increased due to more frequent extreme cases.

7. CONCLUSION

This paper studies to enable LLMs to recognise speech inputs by interfacing with a speech encoder. Three commonly used connectors including fully-connected layers, multi-head cross-attention and Q-Former were compared. The LLMs with Q-Formers demonstrated superior performance over LLMs with other connectors and the Whisper baseline ASR system on all of the in-domain and out-of-domain test sets. Moreover, a novel segment-level Q-Former was proposed to improve the performance with long-form speech inputs whose duration exceeds the limitations of the pre-trained speech encoder. Analyses show that although the rich context information in long-form speech inputs can improve ASR accuracy, overly long inputs can also aggravate the hallucination problem of LLMs.

8. REFERENCES

- [1] OpenAI, “GPT-4 technical report,” *arXiv:2308.11276*, 2023.
- [2] T. Brown, B. Mann, N. Ryder, et al., “Language models are few-shot learners,” in *Proc. NeurIPS*, 2020.
- [3] R. Anil, A.M. Dai, O. Firat, et al., “PaLM 2 technical report,” *arXiv:2305.10403*, 2023.
- [4] H. Touvron, T. Lavril, et al., “LLaMA: Open and efficient foundation language models,” *arXiv:2302.13971*, 2023.
- [5] W. Chiang, Z. Li, Z. Lin, Y. Sheng, Z. Wu, H. Zhang, L. Zheng, S. Zhuang, Y. Zhuang, et al., “Vicuna: An open-source chatbot impressing GPT-4 with 90%* ChatGPT quality,” <https://vicuna.lmsys.org> (accessed 14 April 2023), 2023.
- [6] W. Zeng, X. Ren, T. Su, et al., “Pangu- α : Large-scale autoregressive pretrained Chinese language models with auto-parallel computation,” *arXiv:2104.12369*, 2021.
- [7] R. Huang, M. Li, D. Yang, et al., “AudioGPT: Understanding and generating speech, music, sound, and talking head,” *arXiv:2304.12995*, 2023.
- [8] Y. Shen, K. Song, X. Tan, D. Li, W. Lu, and Y. Zhuang, “HuggingGPT: Solving ai tasks with chatGPT and its friends in huggingface,” *arXiv:2303.17580*, 2023.
- [9] D. Zhang, S. Li, X. Zhang, J. Zhan, P. Wang, Y. Zhou, and X. Qiu, “SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities,” *arXiv:2305.11000*, 2023.
- [10] P.K. Rubenstein, C. Asawaroengchai, D.D. Nguyen, et al., “AudioPaLM: A large language model that can speak and listen,” *arXiv:2306.12925*, 2023.
- [11] F. Chen, M. Han, H. Zhao, Q. Zhang, J. Shi, S. Xu, and B. Xu, “X-LLM: Bootstrapping advanced large language models by treating multi-modalities as foreign languages,” *arXiv:2305.04160*, 2023.
- [12] Y. Shu, S. Dong, G. Chen, et al., “LLaSM: Large language and speech model,” *arXiv:2308.15930*, 2023.
- [13] J. Wu, Y. Gaur, Z. Chen, et al., “On decoder-only architecture for speech-to-text and large language model integration,” *arXiv:2307.03917*, 2023.
- [14] Y. Fathullah, C. Wu, E. Lakomkin, J. Jia, Y. Shangguan, K. Li, J. Guo, W. Xiong, J. Mahadeokar, O. Kalinli, et al., “Prompting large language models with speech recognition abilities,” *arXiv:2307.11795*, 2023.
- [15] Y. Li, Y. Wu, J. Li, and S. Liu, “Prompting large language models for zero-shot domain adaptation in speech recognition,” *arXiv:2306.16007*, 2023.
- [16] R. Ma, M. Qian, P. Manakul, M. Gales, and K. Knill, “Can generative large language models perform asr error correction?,” *arXiv:2307.04172*, 2023.
- [17] P. Dighe, Y. Su, S. Zheng, et al., “Leveraging large language models for exploiting asr uncertainty,” *arXiv:2309.04842*, 2023.
- [18] A. Radford, J. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, “Robust speech recognition via large-scale weak supervision,” in *Proc. ICML*, Honolulu, 2023.
- [19] C. Lyu, M. Wu, L. Wang, et al., “Macaw-LLM: Multi-modal language modeling with image, audio, video, and text integration,” *arXiv:2306.09093*, 2023.
- [20] J. Li, D. Li, S. Savarese, and S. Hoi, “BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models,” in *Proc. ICML*, Vienna, 2023.
- [21] W. Hsu, B. Bolte, Y. 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.
- [22] Y. Chung, Y. Zhang, W. Han, C. Chiu, J. Qin, R. Pang, and Y. Wu, “W2v-BERT: Combining contrastive learning and masked language modeling for self-supervised speech pre-training,” in *Proc. ASRU*, Cartagena, 2021.
- [23] Y. Zhang, W. Han, et al., “Google USM: Scaling automatic speech recognition beyond 100 languages,” *arXiv:2303.01037*, 2023.
- [24] J.B. Alayrac, J. Donahue, P. Luc, et al., “Flamingo: A visual language model for few-shot learning,” in *Proc. NeurIPS*, New Orleans, 2022.
- [25] D. Zhu, J. Chen, X. Shen, X. Li, and M. Elhoseiny, “MiniGPT-4: Enhancing vision-language understanding with advanced large language models,” *arXiv:2304.10592*, 2023.
- [26] W. Dai, J. Li, D. Li, A.M.H. Tiong, J. Zhao, W. Wang, B. Li, P. Fung, and S. Hoi, “InstructBLIP: Towards general-purpose vision-language models with instruction tuning,” *arXiv:2305.06500*, 2023.
- [27] H. Zhang, X. Li, and L. Bing, “Video-LLaMA: An instruction-tuned audio-visual language model for video understanding,” *arXiv:2306.02858*, 2023.
- [28] Y. Su, T. Lan, H. Li, J. Xu, Y. Wang, and D. Cai, “PandaGPT: One model to instruction-follow them all,” *arXiv:2305.16355*, 2023.
- [29] G. Chen, Y.D. Zheng, et al., “VideoLLM: Modeling video sequence with large language models,” *arXiv:2305.13292*, 2023.
- [30] M. Maaz, H. Rasheed, S. Khan, and F.S. Khan, “Video-ChatGPT: Towards detailed video understanding via large vision and language models,” *arXiv:2306.05424*, 2023.
- [31] Y. Gong, H. Luo, A.H. Liu, L. Karlinsky, and J. Glass, “Listen, think, and understand,” *arXiv:2305.10790*, 2023.
- [32] S. Liu, A. Hussain, C. Sun, and Y. Shan, “Music understanding LLaMA: Advancing text-to-music generation with question answering and captioning,” *arXiv:2308.11276*, 2023.
- [33] A. Vaswani, N. Shazeer, N. Parmar, et al., “Attention is all you need,” in *Proc. NeurIPS*, Long Beach, 2017.
- [34] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, “Librispeech: An ASR corpus based on public domain audio books,” in *Proc. ICASSP*, South Brisbane, 2015.
- [35] R. Ardila, M. Branson, K. Davis, et al., “Common Voice: A massively-multilingual speech corpus,” in *Proc. LREC*, Marseille, 2020.
- [36] G. Chen, S. Chai, G. Wang, et al., “GigaSpeech: An evolving, multi-domain asr corpus with 10,000 hours of transcribed audio,” in *Proc. Interspeech*, Brno, 2021.
- [37] Y. Lin, T. Han, H. Xu, et al., “Random utterance concatenation based data augmentation for improving short-video speech recognition,” in *Proc. Interspeech*, Dublin, 2023.