Interspeech 2025
17-21 August 2025, Rotterdam, The Netherlands

How to Connect Speech Foundation Models and Large Language Models? What Matters and What Does Not

Francesco Verdini[1,3,*], *Pierfrancesco Melucci*[1,2,3,*], *Stefano Perna*[2,3,*], *Francesco Cariaggi*[3,4,*], *Marco Gaido*[5], *Sara Papi*[5], *Szymon Mazurek*[6,7], *Marek Kasztelnik*[7], *Luisa Bentivogli*[5], *Sébastien Bratières*[3,4], *Paolo Merialdo*[2], *Simone Scardapane*[1]

[1]Sapienza University of Rome, Italy; [2]Roma Tre University, Italy; [3]Translated, Italy; [4]Pi School, Italy; [5]Fondazione Bruno Kessler, Italy; [6]AGH University of Krakow, Poland; [7]ACC Cyfronet AGH, Poland

francesco.verdini@uniroma1.it, pierfrancesco.melucci@uniroma1.it, stefano.perna@uniroma3.it, francesco.cariaggi@picampus-school.com

Abstract

The remarkable performance achieved by Large Language Models (LLM) has driven research efforts to leverage them for a wide range of tasks and input modalities. In speech-to-text (S2T) tasks, the emerging solution consists of projecting the output of the encoder of a Speech Foundational Model (SFM) into the LLM embedding space through an adapter module. However, no work has yet investigated how much the downstream-task performance depends on each component (SFM, adapter, LLM) nor whether the best design of the adapter depends on the chosen SFM and LLM. To fill this gap, we evaluate the combination of 5 adapter modules, 2 LLMs (Mistral and Llama), and 2 SFMs (Whisper and SeamlessM4T) on two widespread S2T tasks, namely Automatic Speech Recognition and Speech Translation. Our results demonstrate that the SFM plays a pivotal role in downstream performance, while the adapter choice has moderate impact and depends on the SFM and LLM.

Index Terms: automatic speech recognition, speech translation, LLM, foundation models, adapters

1. Introduction

The success of Large Language Models (LLMs) [1] has attracted significant interest in extending their capabilities to handle various input modalities such as vision [2] and speech [3]. In the speech scenario, several studies [3–7] have proposed the integration of a pretrained Speech Foundation Model (SFM) encoder with a pretrained LLM through an *adapter* module, realizing the SFM+LLM new architectural paradigm [8]. The adapter can be decomposed into two components, as shown in Figure 1: a *length adapter*, which compresses the speech sequence along the time dimension, and a *modality adapter*, which maps the compressed input into an embedding space compatible with the LLM. The SFM+LLM solution exploits, on the one hand, the SFM ability to extract high-quality semantic representations of the speech input and, on the other, the fluency and vast linguistic knowledge of LLMs, achieving competitive scores for widespread tasks such as Automatic Speech Recognition (ASR) [9] and Speech Translation (ST) [7].

However, being a recent trend, research efforts have mostly been devoted to demonstrating the effectiveness of this paradigm over traditional methods, without delving into the single design choices. Specifically, many architectural solutions have been proposed for the length adapter, often employed to both reduce the LLM computational costs and the modality mismatch with the textual sequences. These methods span from fixed downsampling, obtained either with a stack of strided convolutions [9] or with window-level Q-Former [3], to modules with variable compression rates that reduce the input sequence based on its semantic content, such as Continuous Integrate-and-Fire (CIF) [10] and CTC compression [11]. Nonetheless, a comprehensive study on the adapter choice is missing: while some comparisons are present in the literature [5, 7, 12], these evaluations are narrow in scope and assume the optimal solution is independent of the chosen SFM or LLM.

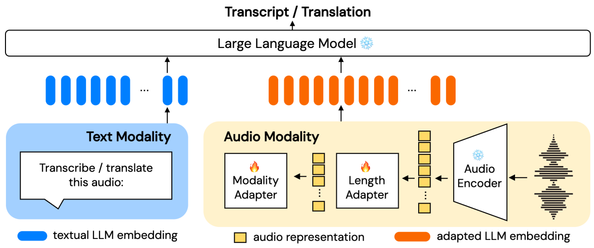

Figure 1: *Schema of the SFM+LLM architecture.*

In this work, we explore whether a one-size-fits-all design choice for the adapter exists that maximizes ASR and ST performance or if this depends on the selected SFM and LLM. Through a systematic comparison of a wide range of adapters proposed in the literature and by analyzing their impact in combination with widely used SFMs (Whisper [13] and SeamlessM4T [14]) and LLMs (Llama [1] and Mistral [15]), our contributions can be summarized as follows:

- We prove that performance highly varies when a different SFM is selected (on average, >2 COMET points for ST and 1 WER for ASR), whereas the choice of the LLM and adapter has a less pronounced impact on the final performance.
- We show that there is no one-size-fits-all solution for the adapter as its choice highly depends on the selected SFM and LLM combination.

Our codebase will be released under the Apache 2.0 License upon paper acceptance.

2. Methodology

Figure 1 illustrates the design of SFM+LLM solutions, highlighting the three key components: the SFM encoder, the adapter, and the LLM. The processing pipeline begins with an input audio segment a, which is first encoded by the SFM encoder $\mathcal{S}_E$ into a sequence of learned audio representations.

*These authors contributed equally.

10.21437/Interspeech.2025-2245

These are then processed by the adapter $\mathcal{A}$ and the resulting embeddings are concatenated to the embeddings $\mathcal{P}$ of a task-specific textual prompt.[1] The joining operation J, which combines the audio and prompt embeddings, follows the same methodology established in the framework proposed by LLaVa [2] to construct the final prompt that is passed to the LLM. In particular, with $\mathcal{L}$ as the LLM, we can formally express the generation of the output transcription or translation y through the following equation:

$$y = \mathcal{L}\left(J\left(\mathcal{P}, \mathcal{A}\left(\mathcal{S}_E(a)\right)\right)\right) \quad (1)$$

As the main rationale behind the SFM+LLM solution comes from the possibility of training a high-quality ASR or ST system without large training datasets – thus with limited computational costs and memory requirements – we keep $\mathcal{L}$ and $\mathcal{S}_E$ frozen, training only $\mathcal{A}$. This solution is coherent with previous work [16], which showed that the gains obtained by fine-tuning the whole SFM encoder and LLM do not justify the additional costs when training models at this scale and for these tasks. Within this framework, we answer our research question on the relative importance of the three components for the downstream performance by varying each of them as illustrated in the following sections.

2.1. SFM Encoder

To investigate the impact of the SFM encoder, we use two widely recognized SFMs for speech representation extraction: Whisper [13] and SeamlessM4T [14]. In particular, we use the large version of both of them, namely Whisper large-v3,[2] and SeamlessM4T v2-large.[3] While Whisper is the most popular SFM in recent works on SFM+LLM [3, 4, 17–19], representing a natural choice, the usage of SeamlessM4T has never been explored to the best of our knowledge. Both models have demonstrated strong performance in multilingual speech processing tasks, with Whisper showing particular strength in ASR tasks and SeamlessM4T excelling in speech-to-speech and speech-to-text translation. Moreover, their architectural differences make them interesting candidates for comparative analysis in our study. Nonetheless, we opted for it not only for its recognized quality, but also because its design is very different from Whisper. SeamlessM4T is built with a customized version of Conformer layers [20] instead of Transformer ones, and also the compression factor of the input sequence is very different. While both models process audio sequences where each frame represents 10ms of audio (100 frames per second), they differ in their temporal resolution: Whisper emits one frame every 20ms (50 frames per second, 2× downsampling), while SeamlessM4T's encoder returns one frame every 160ms (6.25 frames per second, 16× downsampling). This substantial difference in temporal resolution, with SeamlessM4T performing much more aggressive downsampling, provides an interesting opportunity to study how these architectural choices impact the optimal adapter design.

2.2. Adapter

As we keep the SFM and LLM frozen, we design adapters with high representation capacity, allowing for an effective mapping of the embeddings to the LLM input space. To do so, we follow [4, 16] which use a stack of vanilla Transformer [21] encoder layers with bidirectional self-attention as modality adapters and investigate different methods as adapters. The adapters are trained using a cross-entropy loss on the output of the LLM having the transcripts for ASR and translations for ST as target, unless stated otherwise. One key characteristic of adapters is how they handle the sequence length. Some, like the Base adapter, keep the original one, while others reduce it using different strategies. Conv-based and WLQ-former-based adapters apply a constant-factor compression, while CIF-based and CTC-based adapters use a content-based one. Overall, we investigate the following 5 adapters, shown in Figure 2.

Base. We use 4 transformer encoder layers with hidden size 768, with 3072 as the intermediate size and using 12 attention heads. This adapter does not perform any length adaptation.

Conv-based. We introduce 2 1D-convolutional layers with stride 2 after the second layer of the Base adapter, resulting in a final compression factor of 4. No auxiliary loss is used.

CIF-based. Similarly to Conv-based, the adapter is extended by introducing a Continuous Integrate-and-Fire (CIF) [10] length adapter after the second Transformer layer. CIF is a sequence compression mechanism that accumulates input features over time and emits an output when a given integration threshold is reached, enabling variable-length sequence compression while preserving key information. To train this module, we add two auxiliary losses: a Connectionist Temporal Classification (CTC) loss [22] with the transcripts as target, following [23], and a quantity loss that controls the compression factor. The weight associated with both auxiliary losses is 0.1. On average, this corresponds to a compression factor of 3 with SeamlessM4T and 25 with Whisper.

CTC-based. In this case, the length adapter is a CTC-based compression [11], which collapses consecutive equal predictions of a CTC module by averaging the corresponding vectors, trained on the transcripts with an auxiliary CTC loss as done in CIF-based. On average, this corresponds to a compression factor of 2 for SeamlessM4T and 13 for Whisper.

WLQ-former. This adapter performs both modality and length adaptation with a window-level Q-Former [3]. This module processes variable-length encoded speech sequences by dividing them into fixed-length windows of encoded frames and feeding each of these non-overlapping windows to a Q-former architecture [24]. The Q-former uses a fixed and configurable number of learnable query vectors to attend to each window through cross-attention. As a result, the compression factor is controlled by the window length and the number of queries, which we set to 0.33 seconds and 1 respectively as per [3], and therefore results in 2 for SeamlessM4T and 16 for Whisper.

Table 1: *Compression rate for each configuration of SFM/Adapter.*

SFM	Adapter	Compression ratio	Sampling rate (Hz)
Seamless	Base	1:1	6.25
	CIF-based	3:1	2.08
	Conv-based	4:1	1.56
	CTC-based	2:1	3.12
	WLQ-former	2:1	3.12
Whisper	Base	1:1	50.00
	CIF-based	25:1	2.00
	Conv-based	4:1	12.50
	CTC-based	13:1	3.85
	WLQ-former	16:1	3.12

[1] In our experiments, we use as textual prompt *"can you transcribe L_s?"* for ASR, where L_s is the language of the audio, and *"can you translate from L_s to L_t?"* for ST, where L_t is the target language.

[2] https://huggingface.co/openai/whisper-large-v3

[3] https://huggingface.co/facebook/seamless-m4t-v2-large

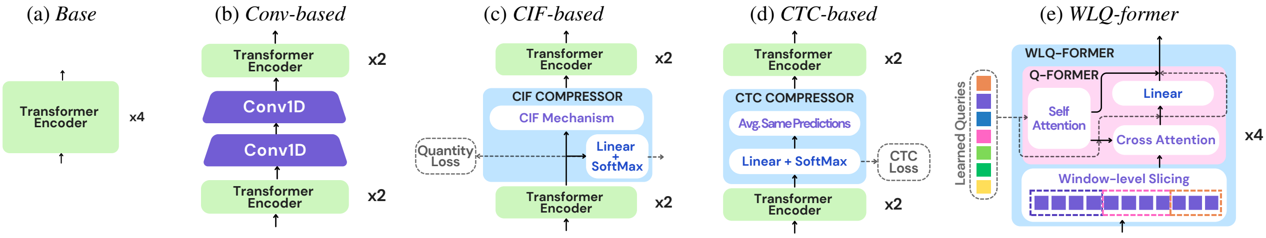

Figure 2: *Representation of the adapters analyzed in the paper.*

Table 2: *Number of parameters for each adapter.*

Adapter	# trainable parameters (M)		
	len. adapt.	mod. adapt.	Total
Base	0	28.35	28.35
Conv-based	25.20	28.35	53.55
CIF-based	28.15	28.35	56.50
CTC-based	25.20	28.35	53.55
WLQ-former	33.09		

In Table 1, we report compression factors for each adapter. We also distinguish between the SFMs (SeamlessM4T and Whisper) as their output sequence length is very different and this influences the compression factor, especially for the content-based adapters (CIF-based and CTC-based). For the content-based length adapters, we present the estimated compression ratios on the test set, evaluated after training. Notably, the CIF-based adapter reaches the highest compression ratio (25:1) with Whisper, which corresponds to a $\sim$2 Hz frequency of output vectors. The CTC-based, on the other hand, reduces the output sequence to $\sim$3 Hz using SeamlessM4T while it reduces the sequence to approximately 4 Hz with Whisper. The textual representations of the ground truth transcriptions, as tokenized by the LLM, contain $\sim$2.7 token per second, i.e. slightly longer than those produced by CIF-based and slightly shorter than those produced by the CTC-based and WLQ-former.

2.3. LLM Decoder

As LLMs, we select Mistral-7B-Instruct-v0.3,[4] and Llama-3.1-8B-Instruct[5]. The former is an English-centric model while the latter is multilingual, covering English, German, French, Italian, Portuguese, Hindi, Spanish, and Thai. Moreover, Llama is the most popular choice in previous works [4, 18, 25]. Also in this case, our goal is to maximize the difference between the investigated LLMs while keeping the dimension close.

3. Experimental Settings

We trained all our models on CoVoST 2 [26] using English, German, Spanish, French, and Italian as source languages and German and English as target languages, and on MuST-C [27] using German, French, Italian, Spanish as target languages. The datasets for ASR were obtained by taking the audios in the source languages and their corresponding transcriptions for CoVoST, while for MuST-C we took audios and associated transcriptions from the English-German partition. Our training process consisted of 28,000 steps over 2 epochs. This was done using either 4 NVIDIA GH200 96GB GPUs with a micro-batch size of 10 samples and 4 gradient accumulation steps, or 8 NVIDIA A100 40GB GPUs with a micro-batch size of 10 samples and 2 gradient accumulation steps. This approach allowed us to maintain a total batch size of 160 samples for both configurations. We took the last checkpoint as the final model for evaluation in all cases since early experiments showed better results compared to averaging the last ones. A single training run took approximately 24 hours on both described systems. We used AdamW as the optimizer and a cosine scheduler for the learning rate with 10^{-4} as the peak value and a linear warmup of 840 steps. The hyperparameters of all the Transformer layers in the adapters are the same as in the $\text{BERT}_{\text{BASE}}$ model.[6] To ensure a fair comparison, we kept the number of modality adaptation parameters constant across the different architectures, with the exception of the WLQ-Former where a distinction between length and modality adaptation parameters cannot be made. The exact number of trainable parameters for each configuration, with a separation of length adaptation and modality adaptation parameters where possible, is reported in Table 2.

Results are reported as COMET[7] [28] and WER[8] for, respectively, ST and ASR on the CoVoST 2 test sets. Statistical significance is computed using bootstrap resampling [29].

4. Results

In Table 3, we report the ASR and ST results of the SFM+LLM architectures with the various combinations of SFMs, LLMs, and length adapters introduced in Section 2. We did not benchmark inference times as autoregressive text decoding dominates over non-autoregressive encoding of (adapted) audio features.

As previous works mostly rely on BLEU [30], we highlight that our Whisper, Base adapter, and Llama model – a basic configuration adopted by previous SFM+LLM works – scores 28.7 BLEU on CoVoST en-de (the most popular language direction), a significantly higher result than the 25.1 BLEU of Qwen-Audio [31], one of the best performing SFM+LLM solutions. This confirms the soundness of our experimental settings and the competitiveness of our results.

First, we observe that the choice of **the SFM is the most critical factor** in terms of downstream performance. This is shown not only by the fact that, in each configuration, the version equipped with SeamlessM4T outperforms the counterpart with Whisper on both tasks (ASR and ST) on average, but also by the average improvement of more than 2 COMET on ST and more than 1 WER on ASR of the best SeamlessM4T configurations (with Llama and the WLQ-former or Base adapters) over the best Whisper ones (with Llama and the Conv-based adapter

[4]https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.3
[5]https://huggingface.co/meta-llama/Meta-Llama-3.1-8B-Instruct
[6]https://huggingface.co/google-bert/bert-base-uncased
[7]Model: https://huggingface.co/Unbabel/wmt22-comet-da.
[8]Computed using `jiwer` (https://pypi.org/project/jiwer/).

Table 3: *ASR and ST results on CoVoST test sets. The best result for each (SFM, LLM) configuration is underlined, while the overall best is **bolded**. The difference with Base is statistically significant ($p < 0.05$) unless for scores marked with *.*

SFM	LLM	Adapter	ST - COMET (↑)						ASR - WER (↓)					
			en-de	**de-en**	**es-en**	**fr-en**	**it-en**	**avg**	**en**	**es**	**fr**	**it**	**de**	**avg**
SeamlessM4T	Mistral	Base	84.94	84.75	86.65	**84.71**	**85.42**	85.29	6.48	6.56	**9.69**	**7.8**	**8.36**	**7.78**
		CIF-based	84.31	84.33	86.31	84.32	85.07	84.87	7.10	6.92	10.23	8.60	9.38	8.45
		Conv-based	84.33	84.15	86.20	84.11	84.98	84.75	7.53	7.83	11.38	10.07	11.44	9.65
		CTC-based	82.95	82.48	85.20	82.85	83.57	83.41	7.94	7.90	12.51	10.31	12.29	10.19
		WLQ-former	84.67	84.71*	86.60*	84.59	85.29	85.17	**6.38***	6.80	9.83*	8.05	8.48*	7.91
	Llama 3.1	Base	85.12	84.15	86.17	84.08	84.78	84.86	7.15	7.46	10.67	9.20	9.96	8.89
		CIF-based	84.65	83.87	85.98	83.86	84.65	84.60	7.66	7.47*	12.36	10.18	10.50	9.63
		Conv-based	85.42	84.42	86.43	84.31	85.17	85.15	7.16*	7.08	10.79*	8.99	9.83*	8.77
		CTC-based	83.78	82.49	85.21	82.83	83.60	83.58	7.95	8.04	12.17	9.94	11.22	9.90
		WLQ-former	**85.65**	**84.84**	**86.66**	84.68	85.39	**85.44**	6.62	6.69	9.96	7.97	8.71	7.99
Whisper	Mistral	Base	78.98	81.38	84.79	81.63	82.69	81.89	11.37	7.57	12.81	10.14	10.88	10.55
		CIF-based	77.79	80.35	84.11	80.79	81.83	80.99	12.57	8.45	14.24	12.32	13.09	12.13
		Conv-based	78.73	81.26	84.72*	81.52*	82.58	81.76	11.78	7.60	13.23	10.67	11.52	10.96
		CTC-based	75.56	76.53	81.75	78.33	78.55	78.14	14.69	10.63	17.15	15.09	16.50	14.81
		WLQ-former	79.07*	81.44*	84.92	81.68*	82.92	82.00	11.82	8.21	13.60	15.77	12.55	12.39
	Llama 3.1	Base	80.43	82.15	85.21	82.33	83.06	82.64	9.90	**6.33**	11.27	8.52	9.09	9.02
		CIF-based	78.32	78.94	82.51	80.09	80.27	80.02	12.82	8.53	14.31	12.80	13.53	12.40
		Conv-based	80.84	82.57	85.49	82.60	83.51	83.00	9.90*	6.46	11.49*	8.75*	9.00*	9.12
		CTC-based	76.47	73.80	80.16	77.19	76.59	76.84	14.02	10.98	17.55	16.29	17.21	15.21
		WLQ-former	79.95	81.56	84.88	81.56	82.89	82.17	11.98	7.90	14.52	11.10	12.84	11.67

for ST and Llama and Base for ASR). Instead, the choice of the LLM is less critical, as demonstrated by the small gap (<0.2 on both ASR and ST) between the best configuration with Llama (Seamless as SFM and WLQ-former as adapter) and that with Mistral (SeamlessM4T as SFM and Base as adapter). Moreover, the best results on average in ST are obtained with Llama, while the best ones in ASR with Mistral.

Second, results clearly show that **there is no one-size-fits-all solution for the adapter**. Interestingly, the LLM plays an important role in the choice of the adapter. With Mistral, the Base adapter generally yields the best results, even though the WLQ-former is competitive, especially in ST. With Llama, instead, the best adapter varies with the SFM used. While WLQ-former is the best option with SeamlessM4T, the Conv-based and Base adapters emerge with Whisper, with the former being the best in ST and close in ASR, where differences among the two are almost always not statistically significant. Across all SFM and LLM configurations and tasks, the Base adapter always ranks first or second except for Llama+SeamlessM4T in ST, where it is third. Moreover, content-based length adapters consistently underperform other strategies. Together with the observation that there is no clear trend of the results with respect to the compression factor, these insights suggest that reducing the length mismatch between textual and speech representations is not critical for the quality of the outputs. However, reducing the speech sequence length lowers computational costs, making length adapters still a useful module to consider.

All in all, our results demonstrate the need for experimenting in different settings in terms of SFM and LLM when comparing adapter solutions, as improvements in one specific scenario may not generalize. In addition, LLMs shown to be robust to input sequences of very different lengths, as the Base adapter, which does not compress the speech sequence, and the WLQ-former, which has high compression factors (16 with Whisper), achieve competitive scores in most settings.

5. Conclusions

This work systematically analyzed the importance and design of the various building blocks that compose speech-to-text models by connecting an SFM encoder and an LLM through an adapter. To this aim, we compared all the combinations of 2 SFMs, 2 LLMs, and 5 adapters, which mostly differ in their length reduction module. With comprehensive experiments covering two tasks–ASR and ST–and 5 language directions, our results demonstrate that the choice of the SFM is the most critical factor influencing downstream performance. We also established that there is no one-size-fits-all solution for the length adapter, as the optimal choice varies depending on the specific combination of SFM and LLM. Notably, the Base and WLQ-former adapters, which feature very different compression factors, demonstrate strong performance across tasks, suggesting that reducing sequence length mismatch between speech and text is less crucial than previously assumed.

6. Acknowledgments

This paper has received funding from the European Union's Horizon research and innovation programme under grant agreement No 101135798, project Meetween (My Personal AI Mediator for Virtual MEETtings BetWEEN People).

We gratefully acknowledge Poland's high-performance Infrastructure PLGrid ACC Cyfronet AGH for providing computer facilities and support within computational grant no PLG/2024/016971.

Marco Gaido was supported by the PNRR project FAIR - Future AI Research (PE00000013), under the NRRP MUR program funded by the NextGenerationEU.

This work has been carried out while Pierfrancesco Melucci was enrolled in the Italian National Doctorate on Artificial Intelligence run by Sapienza University of Rome in collaboration with Università di Roma Tre.

7. References

[1] A. Dubey, A. Jauhri, A. Pandey, A. Kadian, A. Al-Dahle, A. Letman, A. Mathur, A. Schelten, A. Yang, A. Fan *et al.*, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024.

[2] H. Liu, C. Li, Q. Wu, and Y. J. Lee, "Visual instruction tuning," in *Advances in Neural Information Processing Systems*, A. Oh, T. Naumann, A. Globerson, K. Saenko, M. Hardt, and S. Levine, Eds., vol. 36, 2023, pp. 34 892–34 916.

[3] 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," in *The Twelfth International Conference on Learning Representations*, 2024.

[4] J. Wu, Y. Gaur, Z. Chen, L. Zhou, Y. Zhu, T. Wang, J. Li, S. Liu, B. Ren, L. Liu *et al.*, "On decoder-only architecture for speech-to-text and large language model integration," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE, 2023, pp. 1–8.

[5] C. Wang, M. Liao, Z. Huang, J. Lu, J. Wu, Y. Liu, C. Zong, and J. Zhang, "Blsp: Bootstrapping language-speech pre-training via behavior alignment of continuation writing," *arXiv preprint arXiv:2309.00916*, 2023.

[6] S. Hu, L. Zhou, S. Liu, S. Chen, H. Hao, J. Pan, X. Liu, J. Li, S. Sivasankaran, L. Liu *et al.*, "Wavllm: Towards robust and adaptive speech large language model," *arXiv preprint arXiv:2404.00656*, 2024.

[7] C. Wang, M. Liao, Z. Huang, and J. Zhang, "Blsp-kd: Bootstrapping language-speech pre-training via knowledge distillation," *arXiv preprint arXiv:2405.19041*, 2024.

[8] M. Gaido, S. Papi, M. Negri, and L. Bentivogli, "Speech translation with speech foundation models and large language models: What is there and what is missing?" in *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Bangkok, Thailand, Aug. 2024, pp. 14 760–14 778.

[9] 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," in *ICASSP 2024-2024 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2024, pp. 12 637–12 641.

[10] L. Dong and B. Xu, "Cif: Continuous integrate-and-fire for end-to-end speech recognition," in *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2020, pp. 6079–6083.

[11] M. Gaido, M. Cettolo, M. Negri, and M. Turchi, "CTC-based compression for direct speech translation," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, Online, Apr. 2021, pp. 690–696.

[12] Z. Ma, G. Yang, Y. Yang, Z. Gao, J. Wang, Z. Du, F. Yu, Q. Chen, S. Zheng, S. Zhang, and X. Chen, "An embarrassingly simple approach for llm with strong asr capacity," *arXiv preprint arXiv:2310.06825*, 2024.

[13] A. Radford, J. W. Kim, T. Xu, G. Brockman, C. McLeavey, and I. Sutskever, "Robust speech recognition via large-scale weak supervision," in *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23, 2023.

[14] L. Barrault, Y.-A. Chung, M. C. Meglioli, D. Dale, N. Dong, P.-A. Duquenne, H. Elsahar, H. Gong, K. Heffernan, J. Hoffman *et al.*, "Seamlessm4t-massively multilingual & multimodal machine translation," *arXiv preprint arXiv:2308.11596*, 2023.

[15] A. Q. Jiang, A. Sablayrolles, A. Mensch, C. Bamford, D. S. Chaplot, D. d. l. Casas, F. Bressand, G. Lengyel, G. Lample, L. Saulnier *et al.*, "Mistral 7b," *arXiv preprint arXiv:2310.06825*, 2023.

[16] M. Wang, W. Han, I. Shafran, Z. Wu, C.-C. Chiu, Y. Cao, N. Chen, Y. Zhang, H. Soltau, P. K. Rubenstein, L. Zilka, D. Yu, G. Pundak, N. Siddhartha, J. Schalkwyk, and Y. Wu, "Slm: Bridge the thin gap between speech and text foundation models," in *2023 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, 2023, pp. 1–8.

[17] Y. Chu, J. Xu, Q. Yang, H. Wei, X. Wei, Z. Guo, Y. Leng, Y. Lv, J. He, J. Lin *et al.*, "Qwen2-audio technical report," *arXiv preprint arXiv:2407.10759*, 2024.

[18] X. Chen, S. Zhang, Q. Bai, K. Chen, and S. Nakamura, "LLaST: Improved end-to-end speech translation system leveraged by large language models," in *Findings of the Association for Computational Linguistics ACL 2024*, Bangkok, Thailand and virtual meeting, Aug. 2024, pp. 6976–6987.

[19] S. Khurana, C. Hori, A. Laurent, G. Wichern, and J. Le Roux, "Zerost: Zero-shot speech translation," in *Interspeech 2024*, 2024, pp. 392–396.

[20] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu, and R. Pang, "Conformer: Convolution-augmented transformer for speech recognition," in *Interspeech 2020*, 2020, pp. 5036–5040.

[21] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, "Attention is all you need," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, ser. NIPS'17, Red Hook, NY, USA, 2017, p. 6000–6010.

[22] A. Graves, S. Fernández, F. Gomez, and J. Schmidhuber, "Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks," in *Proceedings of the 23rd International Conference on Machine Learning*, ser. ICML '06, New York, NY, USA, 2006, p. 369–376.

[23] S. Kim, T. Hori, and S. Watanabe, "Joint ctc-attention based end-to-end speech recognition using multi-task learning," in *2017 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2017, pp. 4835–4839.

[24] 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 *Proceedings of the 40th International Conference on Machine Learning*, ser. ICML'23, 2023.

[25] H. Zhang, N. Si, Y. Chen, W. Zhang, X. Yang, D. Qu, and X. Jiao, "Tuning large language model for end-to-end speech translation," *arXiv preprint arXiv:2310.02050*, 2023.

[26] C. Wang, A. Wu, J. Gu, and J. Pino, "Covost 2 and massively multilingual speech translation," in *Interspeech 2021*, 2021, pp. 2247–2251.

[27] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "MuST-C: a Multilingual Speech Translation Corpus," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 2012–2017.

[28] R. Rei, C. Stewart, A. C. Farinha, and A. Lavie, "COMET: A neural framework for MT evaluation," in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Online, Nov. 2020, pp. 2685–2702.

[29] P. Koehn, "Statistical significance tests for machine translation evaluation," in *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, D. Lin and D. Wu, Eds., Barcelona, Spain, Jul. 2004, pp. 388–395.

[30] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, "Bleu: a method for automatic evaluation of machine translation," in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, Philadelphia, Pennsylvania, USA, Jul. 2002, pp. 311–318.

[31] 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.