Interspeech 2024
1-5 September 2024, Kos, Greece

Comparing Discrete and Continuous Space LLMs for Speech Recognition

Yaoxun Xu[1], *Shi-Xiong Zhang*[2], *Jianwei Yu*[2], *Zhiyong Wu*[1,3,†], *Dong Yu*[2]

[1] Shenzhen International Graduate School, Tsinghua University, Shenzhen, China
[2] Tencent AI Lab
[3] The Chinese University of Hong Kong, Hong Kong SAR, China

xuyx22@mails.tsinghua.edu.cn, zhangshixiong@gmail.com,
tomasyu@tencent.com, zywu@sz.tsinghua.edu.cn

Abstract

This paper investigates discrete and continuous speech representations in Large Language Model (LLM)-based Automatic Speech Recognition (ASR), organizing them by feature continuity and training approach into four categories: supervised and unsupervised for both discrete and continuous types. We further classify LLMs based on their input and autoregressive feedback into continuous and discrete-space models. Using specialized encoders and comparative analysis with a Joint-Training-From-Scratch Language Model (JTFS LM) and pre-trained LLaMA2-7b, we provide a detailed examination of their effectiveness. Our work marks the first extensive comparison of speech representations in LLM-based ASR and explores various modeling techniques. We present an open-sourced achievement of a state-of-the-art Word Error Rate (WER) of 1.69% on LibriSpeech using a HuBERT encoder, offering valuable insights for advancing ASR and natural language processing (NLP) research.

Index Terms: Speech Recognition, Large Language Model, Continuous LLM, GPT, LLaMA2

1. Introduction

Automatic Speech Recognition (ASR) [1, 2] is pivotal in speech processing and human-machine interaction. A critical aspect of ASR is the representation of raw speech signals, where redundancy poses a challenge to computational efficiency. This challenge has spurred the development of more compact speech representations, divided into discrete and continuous forms. While discrete representations offer cost efficiency with limited information capacity, continuous representations, despite being more expensive, encapsulate a richer array of information.

Traditional ASR models [3, 4] utilize discrete speech units, like phonemes [5] or triphones [6], for continuous speech feature modeling, encompassing components such as acoustic models (AMs), pronunciation lexicons and language models (LMs). These units serve as discrete intermediate representations of continuous speech signals in these cascaded systems. Alternatively, end-to-end ASR frameworks [7, 8, 9], jointly modeling the continuous acoustic signal and discrete language models, have gradually become mainstream. These frameworks no longer treat ASR systems as cascaded systems (AM and LM) but instead, they directly transform speech features into continuous representations through encoder [10, 11, 12] and joint model them with word embeddings in the continuous space.

In the realm of natural language processing (NLP), language models conventionally employ discrete text tokens as processing units. Recently, the rise of large language models (LLMs) has notably advanced NLP [13, 14], thanks to increased data availability and computational power. Applying LLMs' advanced language understanding and generation to improve ASR has become a key research area. For instance, VioLA [15] converts audio signals into discrete codecs via a pretrained codebook, then autoregressively generates and decodes tokens to text. SpeechGPT [16] and AudioPaLM [17] use K-means to convert continuous audio into discrete forms for processing with pretrained LLaMA [18] and PaLM [19] models, respectively. These methods incorporate discretized speech tokens into LLMs, while others integrate continuous speech features. [20] feeds HuBERTCTC and HuBERT's continuous outputs to GPT-NeoX [21], directly leverages continuous speech features. Models like SALM [22], Whispering LLaMA [23], SALMONN [24], and Qwen-Audio [25] also integrate continuous speech with LLMs via adapters, showcasing diverse methods to combining ASR with advanced language modeling.

Despite the advancements in LLM-based speech recognition, there's a noticeable gap in systematically analyzing discrete versus continuous speech representations within this domain. Our study seeks to fill this void by exploring these representations in LLM-based speech recognition tasks, offering a detailed comparison to guide future research. We classify speech into discrete and continuous categories based on the nature of speech features and further distinguish them by whether they utilize paired speech-transcription data in training the speech encoder, leading to four distinct groups: supervised and unsupervised discrete speech representations, along with supervised and unsupervised continuous speech representations. For each category, we create specialized speech encoder and models to conduct comprehensive comparisons using both a Joint-Training-From-Scratch Language Model (JTFS LM) and the pretrained LLaMA2-7b [26] as benchmarks, shedding light on the effectiveness of these four types of speech representations. The main contributions of this study are as follows:

- To the best of our knowledge, this is the first comprehensive comparative study focusing on discrete and continuous speech representations in LLM-based speech recognition.
- This study proposes and evaluates different modeling approaches within discrete and continuous space LLMs.
- To our knowledge, this represents the state-of-the-art in open-sourced models on LibriSpeech, achieving a WER of 1.69%.[1]

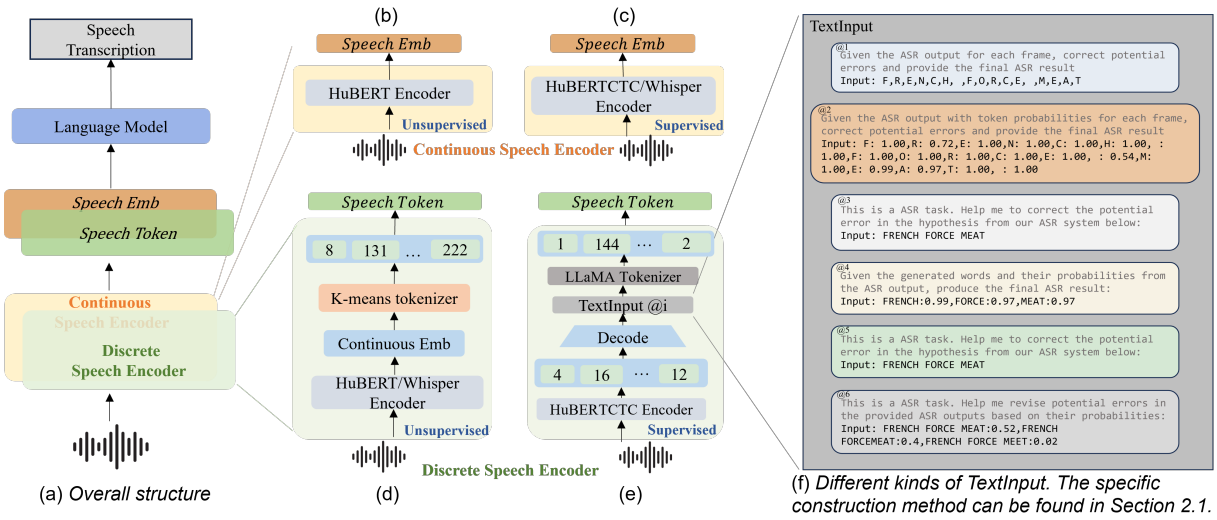
2. LLM based Speech Recogntion

As illustrated in Figure 1(a), we utilize discrete or continuous speech encoders to preprocess speech, yielding either discrete tokens or continuous embeddings. These are then integrated into language models to produce the final transcription results.

† Corresponding author.

[1] The code can be found: https://github.com/xuyaoxun/ASRCompare

10.21437/Interspeech.2024-1533



(f) *Different kinds of TextInput. The specific construction method can be found in Section 2.1.*

Figure 1: *Overall architectures for continuous and discrete speech encoders. Figures (b) to (e) illustrate four distinct speech encoders, each extracting a different type of speech representation.*

2.1. Discrete V.S. Continuous Speech Representations

For the four distinct types of speech representations, we pinpoint and select highly representative features within each category, subsequently designing corresponding feature extractors.

Continuous Unsupervised Representation

In Figure 1(b), we use the pre-trained HuBERT [27] model as speech encoder. HuBERT is a self-supervised learning model designed for speech processing tasks. To evaluate its layer-wise extraction capabilities, we select representations from the 0th layer (post-preprocessing and before entering the HuBERT encoder) and the 8th, 16th, and 24th layers of HuBERT-large as continuous unsupervised speech representations.

Continuous Supervised Representation

In Figure 1(c), we use the pre-trained HuBERT-CTC model as speech encoder. This model augments HuBERT with a prediction layer and reduces dimensions to 32 modeling units, utilizing additional speech-text pairs and CTC [28] loss for optimization. We employ the 24th layer of HuBERT and the 32-dim prediction features from the projection layer. Moreover, we adopt Whisper's [9] encoder as an alternative extractor, which is an encoder-decoder model trained on numerous speech-text pairs, yielding another continuous supervised representation.

Discrete Unsupervised Representation

In Figure 1(d), we train K-means clustering extractors with 500, 1000, and 1500 clusters on a subset of training dataset. Using these K-means extractors, we categorize continuous speech representations and remove duplicates to form discrete unsupervised speech representations.

Discrete Supervised Representation

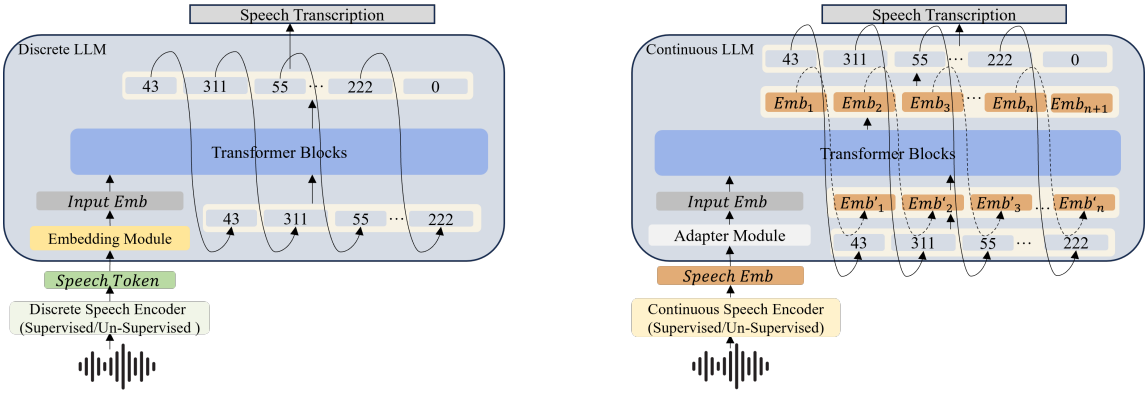
In Figure 1(e), we obtain high-probability tokens from pre-trained HuBERT-CTC logits using softmax, map them to text, and apply post-processing strategies for unique textual prompts. As shown in Figure 1(f), method @1 maps deduplicated HuBERT-CTC logits into the text domain, separating characters with commas. Method @2 adds character probabilities to the input. Method @3 forms a sentence by concatenating characters. Method @4 calculates the mean probability of tokens in a word, based on HuBERT's original logits, and appends it to the input. Method @5 uses the highest-scoring sentence after rescoring speech logits via HuBERT-CTC with a pre-trained 4-gram language model. Method @6 rescores HuBERT-CTC logits using a pre-trained 4-gram language model, providing the top three results and their probabilities as input. After obtaining the TextInput, we seamlessly integrate it by passing it through the LLaMA2 tokenizer to generate discrete tokens, which serve as the supervised discrete representation.

2.2. Discrete V.S. Continuous Modeling

To comprehensively compare discrete and continuous speech representations, we devise models tailored specifically for each type of representation, as illustrated in Figure 2. Specifically, we have employed the Transformer [29]-based language models, LLaMA2 and JTFS LM, as benchmarks for comparison.

In the discrete scenario shown in Figure 2(a), we employ discrete speech encoders (detailed in Section 2.1) to convert speech segments into speech tokens. These tokens are passed through an embedding module to obtain input embeddings, which are then fed into the Transformer blocks. For discrete supervised speech features, the embedding module consists of the text embedding layer from the LLaMA2 tokenizer. In contrast, for discrete unsupervised speech representations, the corresponding embedding module comprises an embedding layer mapping from the K-means cluster count to the Transformer dimensions, along with a two-layer perceptron. During this phase, the JTFS LM trains all parameters jointly, while LLaMA2 is fine-tuned using the LoRA [30] method. Text sequences are generated sequentially through an autoregressive approach.

In the continuous scenario depicted in Figure 2(b), we extract continuous speech embeddings from the continuous speech encoder (as detailed in Section 2.1). These embeddings are then processed through an adapter module composed of a two-layer perceptron to obtain the input embeddings. The input embeddings are fed into the Transformer blocks to generate text sequences autoregressively. Notably, for the JTFS LM (dashed line in Figure 2(b)), we avoid discretizing the Transformer blocks' output to prevent information loss. Instead, we directly transfer the output to the input side, preserving its integrity. Concurrently, the Transformer blocks' output is projected through a mapping layer to obtain the token sequence in the dictionary, continuing until a termination symbol is reached. For LLaMA2, similar to the discrete scenario, we discretize the Transformer blocks' output and pass it through the LLaMA2 tokenizer before re-inserting it into the Transformer blocks.



(a) *Model design for discrete scenarios*

(b) *Model design for continuous scenarios*

Figure 2: *Model design for discrete and continuous scenarios. In Figure 2(b), dashed lines show the data flow for the JTFS LM, and solid lines for the LLaMA2 model*

2.3. Training Loss

In the case of discrete scenarios (Figure 2(a)) and continuous scenarios with LLaMA2 (solid lines in Figure 2(b)), we employ the cross-entropy loss (CELoss) as the loss function during training:

$$\text{loss} = \text{CELoss}(target, predict)$$

Here, *target* denotes the ground truth labels, and *predict* represents the predicted output.

For the continuous LLM with JTFS LM (dashed lines in Figure 2(b)), we employ MSELoss alongside CELoss, ensuring consistency between the Transformer Blocks' input and output while enabling autoregressive generation:

$$\text{loss} = \text{CELoss}(target, predict) + \alpha * \text{MSELoss}(emb\_in, emb\_out)$$

In this equation, α is a hyperparameter that controls the balance between the two loss components. *emb_in* refers to the input embedding, while *emb_out* denotes the output embedding.

3. Experiment

We perform experiments on the LibriSpeech [31] dataset, comprising 960.9 hours of training, 10.7 hours of validation, 5.4 hours of test-clean, and 5.1 hours of test-other audio. We evaluate performance using WER on test-clean and test-other sets. Our JTFS LM consists of 10 stacked Transformer layers with 1024 dimensions and 8 attention heads. In discrete speech representation, we create distinct embedding layers and two-layer perceptrons to project discrete representations into the Transformer's space, depending on the number of clustered tokens. For continuous representations, we use a two-layer perceptron to project speech representations into the Transformer's or LLaMA2's space. In supervised discrete speech representation, we fine-tune LLaMA with LoRA, using rank 16, alpha 16, target modules for gate projection, down projection, and up projection layers, and a dropout rate of 0.05. We set the learning rate at 1e-5, α at 100, and train the model using 8 A100 GPUs.

4. Result

4.1. Results on joint-training-from-scratch language model

We conduct an in-depth analysis of the experimental results derived from the JTFS LM, examining the effects of various configurations. These include the type of speech representation (discrete vs. continuous), the choice of speech encoder (HuBERT vs. Whisper), the number of K-means clusters, and the layer of the HuBERT model.

Table 1: *Experiment results on JTFS LM*

Speech Type	Encoder	Kmeans Cluster	Layer	ID	WER(%) clean/other
Discrete Unsupervised	HuBERT	500	0	#1	109.47/109.54
			8	#2	96.62/98.13
			16	#3	80.54/81.79
			24	#4	76.8/77.51
		1000	0	#5	106.04/107.13
			8	#6	89.12/93.8
			16	#7	72.2/75.57
			24	#8	74.08/72.69
		1500	0	#9	119.01/117.2
			8	#10	87.33/92.11
			16	#11	70.13/71.91
			24	#12	71.85/70.39
	Whisper	500	—	#13	75.59/76.94
		1000	—	#14	58.27/60.01
		1500	—	#15	49.18/51.90
Continuous Unsupervised	HuBERT	—	0	#16	41.13/67.55
		—	8	#17	12.47/23.93
		—	16	#18	7.17/12.21
		—	24	#19	18.27/26.14
Continuous Supervised	Whisper	—	—	#20	5.28/9.74

The Impact of Continuous and Discrete Configurations: The continuous configuration outperforms the discrete one, as shown by the WER comparisons between experiments #11 and #18, and #15 and #20. This notable reduction in WER in the continuous setting highlights its effectiveness. The advantage stems from the Joint-Training-From-Scratch Language Model (JTFS LM), which, beginning with random initialization, leverages either discrete or continuous speech representations. Discrete tokens undergo significant information loss during clustering, whereas continuous representations retain most of the information crucial for ASR. This indicates that information loss increases with the discreteness of tokens.

The Impact of Different Encoder: The choice of the encoder is critical, with Whisper encoders consistently outperforming HuBERT encoders in both discrete (with identical K-means) and continuous settings, as shown in comparisons #12 with #15, and #18 with #20. These results highlight Whisper's enhanced feature extraction ability, indicating that supervised training can further improve the alignment of speech representations to text, thus boosting performance.

The Impact of K-means Clusters Number: WER consistently decreases with an increase in the number of K-means clusters, evident from comparisons #13, #14, and #15. This pattern holds true for the HuBERT encoder, suggesting that more clusters enhance performance by capturing a greater volume of information. However, a higher cluster count also means a more complex feature extraction process, which may demand additional computational resources and time. Therefore, balancing the number of clusters with resource constraints is key to optimizing performance.

The Impact of Encoder Layers: In the continuous setting using the HuBERT encoder, there's a marked decrease in WER from layer 0th to 16th. However, an uptick in WER is observed at layer 24th, as evidenced by #16 showing higher WER compared to #17 and #18, with #19 experiencing a slight increase. This could be due to HuBERT's focus on predicting cluster IDs during training, potentially leading to the capture of more acoustically driven but semantically irrelevant features.

4.2. Results on LLaMA2

Table 2: *LLaMA2 Exp. For details of TextInput@i see Fig. 1. HuBERT24 Emb refers to HuBERT's 24th layer features, and HuBERTCTC Emb refers to HuBERTCTC's final layer features.*

Speech Type	Method	ID	WER(%) clean/other
Discrete Supervised	HuBERTCTC	#21	2.08/4.24
	HuBERTCTC+4-gram	#22	1.82/3.59
	TextInput@1+LLaMA2	#23	2.14/4.13
	TextInput@2+LLaMA2	#24	2.52/4.41
	TextInput@3+LLaMA2	#25	1.99/4.03
	TextInput@4+LLaMA2	#26	1.96/3.97
	TextInput@5+LLaMA2	#27	1.72/3.57
	TextInput@6+LLaMA2	#28	1.80/3.61
	HuBERTCTC(xlarge)+ 4-gram+LLaMA2	#29	**1.69/3.03**
Discrete Unsupervised	HuBERT24-1500+LLaMA2	#30	65.26/75.85
Continuous Unsupervised	HuBERT24 Emb+LLaMA2	#31	13.05/16.77
Continuous Supervised	HuBERTCTC24 Emb+LLaMA2	#32	6.26/7.09
	HuBERTCTC Emb+LLaMA2	#33	9.99/11.93

LLM as Discrete Error Token Corrector: In discrete supervised scenarios, encompassing Experiments #21 to #29, LLMs function as correctors of erroneous tokens. Here, the encoder initially converts continuous speech into discrete tokens, which are then refined by LLMs. This correction relies on long-context LM probabilities, effectively acting as a second-pass decoding. For instance, the comparison between #22 and #21 demonstrates how employing a pre-trained 4-gram language model in the first-pass WFST decoding [32] significantly improves performance by leveraging enhanced contextual information. Moreover, LLMs further reduce WER, as seen in comparisons #27 with #22. Comparing models #23 with #25 and #24 with #26 shows LLaMA2's enhanced word sensitivity and error correction, surpassing character-level input.

N-best and Confidence Score: When comparing #28 with #27 and #22, it's observed that although #28, offering 3-best results, enriches the information set, it doesn't outperform #27, yet it does exceed #22 in performance. A comparison between #26 and #25 shows that #26, with additional confidence scores, achieves marginally better results. This suggests that while LLaMA2 benefits from extra data, optimizing its 7B parameters for improved outcomes might necessitate more comprehensive N-best lists and confidence score inputs.

State-of-The-Art: Using the TextInput@5 approach with an upgrade from the HuBERT-large to the xlarge encoder, we achieved a WER of 1.6/3.0, as demonstrated in #29. This marks, to our knowledge, the best-reported WER on LibriSpeech using a HuBERT encoder. Our performance surpasses that of [27], which reported a WER of 1.8% using the same HuBERT encoder. While [33] reached a lower WER of 1.5%, their model uses more data and has not been made publicly available.

Supervised *V.S.* Unsupervised: Model #27, which employs HuBERT-CTC trained with transcribed data to generate discrete tokens, significantly outperforms Model #30. The latter relies on an unsupervised clustering algorithm for token production. This performance disparity is credited to LLaMA2's advanced sensitivity to textual features, whereas K-means method used in the unsupervised approach focuses more on auditory characteristics, offering a narrower grasp of semantic content. Additionally, the constrained cluster count in unsupervised learning causes phonetically similar sounds to be merged into the same category, diminishing the model's ability to distinguish nuanced pronunciation differences, thus impacting recognition accuracy.

In continuous methods, supervised Model #32 outshines unsupervised Model #31 due to fine-tuning with speech-text pairs, resulting in text-oriented representations and lower WER, showcasing the advantages of supervised learning in achieving nuanced speech recognition.

Impact of Matched Tokens: Comparing Discrete Supervised with Continuous Supervised methods reveals that despite the richer data from continuous representations, #27 still outperforms #33. This advantage is due to #27 generating token sequences that matched with LLaMA2's pretraining tokens, enhancing compatibility and performance.

Continuous Unsupervised *V.S.* Discrete Unsupervised: In the comparison between Discrete Unsupervised and Continuous Unsupervised methods, #31 surpasses #30. The disparity in performance stems from the discrete tokens derived via clustering, focusing on frame-level acoustic features. The constrained cluster categories in discrete representations lead to a significant loss of acoustic details. As a result, discrete representations, being less informative than their continuous counterparts, present a more substantial challenge for the language model in recognizing speech accurately.

JTFS LM *V.S.* LLM: A comparison of #31 with #19 and #30 with #12 clearly reveals that the pre-trained LLaMA2 model delivers superior recognition results for both discrete unsupervised and continuous unsupervised representations. This outcome underscores the effectiveness of pre-training in enhancing language model performance. The pre-trained LLaMA2 model utilizes its prior knowledge, gleaned from extensive text data, to effectively recognize and decode speech representations, regardless of whether they are discrete or continuous.

5. Conclusion

This study investigates discrete and continuous speech representations in LLM-based ASR, organizing them into supervised and unsupervised categories for both types. We further classify LLMs into continuous and discrete-space models according to their input and feedback mechanisms. Through the development of specialized speech encoders and detailed comparative analysis with a Joint-Training-From-Scratch Language Model and the pre-trained LLaMA2, we conduct the first thorough assessment of these representations' impact on LLM-based ASR. Our rigorous experiments have led to a state-of-the-art, open-sourced WER of 1.69% on the LibriSpeech dataset.

6. Acknowledgement

This work is supported by National Natural Science Foundation of China (62076144), Shenzhen Science and Technology Program (WDZC20220816140515001, JCYJ20220818101014030) and Tencent AI Lab Rhino-Bird Focused Research Program (RBFR2023015).

7. References

[1] D. Yu and L. Deng, *Automatic speech recognition*. Springer, 2016, vol. 1.

[2] O. Abdel-Hamid, A.-r. Mohamed, H. Jiang, L. Deng, G. Penn, and D. Yu, "Convolutional neural networks for speech recognition," *IEEE/ACM Transactions on audio, speech, and language processing*, vol. 22, no. 10, pp. 1533–1545, 2014.

[3] M. Gales, S. Young *et al.*, "The application of hidden markov models in speech recognition," *Foundations and Trends® in Signal Processing*, vol. 1, no. 3, pp. 195–304, 2008.

[4] R. Thangarajan, A. Natarajan, and M. Selvam, "Word and triphone based approaches in continuous speech recognition for tamil language," *WSEAS transactions on signal processing*, vol. 4, no. 3, pp. 76–86, 2008.

[5] M. Yusnita, M. Paulraj, S. Yaacob, S. A. Bakar, A. Saidatul, and A. N. Abdullah, "Phoneme-based or isolated-word modeling speech recognition system? an overview," in *2011 IEEE 7th International Colloquium on Signal Processing and its Applications*. IEEE, 2011, pp. 304–309.

[6] D. Chen, B. Mak, C.-C. Leung, and S. Sivadas, "Joint acoustic modeling of triphones and trigraphemes by multi-task learning deep neural networks for low-resource speech recognition," in *2014 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2014, pp. 5592–5596.

[7] A. Graves, "Sequence transduction with recurrent neural networks," *arXiv preprint arXiv:1211.3711*, 2012.

[8] W. Chan, N. Jaitly, Q. V. Le, and O. Vinyals, "Listen, attend and spell," *arXiv preprint arXiv:1508.01211*, 2015.

[9] 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.

[10] A. Gulati, J. Qin, C.-C. Chiu, N. Parmar, Y. Zhang, J. Yu, W. Han, S. Wang, Z. Zhang, Y. Wu *et al.*, "Conformer: Convolution-augmented transformer for speech recognition," *Interspeech 2020*, 2020.

[11] Z. Yao, L. Guo, X. Yang, W. Kang, F. Kuang, Y. Yang, Z. Jin, L. Lin, and D. Povey, "Zipformer: A faster and better encoder for automatic speech recognition," in *The Twelfth International Conference on Learning Representations*, 2023.

[12] Z. Gao, S. Zhang, I. McLoughlin, and Z. Yan, "Paraformer: Fast and accurate parallel transformer for non-autoregressive end-to-end speech recognition," *arXiv preprint arXiv:2206.08317*, 2022.

[13] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang *et al.*, "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, 2023.

[14] W. X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong *et al.*, "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

[15] T. Wang, L. Zhou, Z. Zhang, Y. Wu, S. Liu, Y. Gaur, Z. Chen, J. Li, and F. Wei, "Viola: Unified codec language models for speech recognition, synthesis, and translation," *arXiv preprint arXiv:2305.16107*, 2023.

[16] 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 preprint arXiv:2305.11000*, 2023.

[17] P. K. Rubenstein, C. Asawaroengchai, D. D. Nguyen, A. Bapna, Z. Borsos, F. d. C. Quitry, P. Chen, D. E. Badawy, W. Han, E. Kharitonov *et al.*, "Audiopalm: A large language model that can speak and listen," *arXiv preprint arXiv:2306.12925*, 2023.

[18] 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.

[19] R. Anil, A. M. Dai, O. Firat, M. Johnson, D. Lepikhin, A. Passos, S. Shakeri, E. Taropa, P. Bailey, Z. Chen *et al.*, "Palm 2 technical report," *arXiv preprint arXiv:2305.10403*, 2023.

[20] 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.

[21] S. Black, S. Biderman, E. Hallahan, Q. Anthony, L. Gao, L. Golding, H. He, C. Leahy, K. McDonell, J. Phang *et al.*, "Gpt-neox-20b: An open-source autoregressive language model," in *Proceedings of BigScience Episode# 5–Workshop on Challenges & Perspectives in Creating Large Language Models*, 2022, pp. 95–136.

[22] 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.

[23] S. Radhakrishnan, C.-H. H. Yang, S. A. Khan, R. Kumar, N. A. Kiani, D. Gomez-Cabrero, and J. N. Tegner, "Whispering llama: A cross-modal generative error correction framework for speech recognition," *arXiv preprint arXiv:2310.06434*, 2023.

[24] 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," *arXiv preprint arXiv:2310.13289*, 2023.

[25] 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.

[26] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, "Llama 2: Open foundation and fine-tuned chat models," *arXiv preprint arXiv:2307.09288*, 2023.

[27] 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.

[28] 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*, 2006, pp. 369–376.

[29] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[30] 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.

[31] 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.

[32] Y. Miao, M. Gowayyed, and F. Metze, "Eesen: End-to-end speech recognition using deep rnn models and wfst-based decoding," in *2015 IEEE workshop on automatic speech recognition and understanding (ASRU)*. IEEE, 2015, pp. 167–174.

[33] Y. Zhang, J. Qin, D. S. Park, W. Han, C.-C. Chiu, R. Pang, Q. V. Le, and Y. Wu, "Pushing the limits of semi-supervised learning for automatic speech recognition," *arXiv preprint arXiv:2010.10504*, 2020.