ON DECODER-ONLY ARCHITECTURE FOR SPEECH-TO-TEXT AND LARGE LANGUAGE MODEL INTEGRATION

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu,
Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, Yu Wu

Microsoft, One Microsoft Way, Redmond, USA

ABSTRACT

Large language models (LLMs) have achieved remarkable success in the field of natural language processing, enabling better human-computer interaction using natural language. However, the seamless integration of speech signals into LLMs has not been explored well. The "decoder-only" architecture has also not been well studied for speech processing tasks. In this research, we introduce Speech-LLaMA, a novel approach that effectively incorporates acoustic information into text-based large language models. Our method leverages Connectionist Temporal Classification and a simple audio encoder to map the compressed acoustic features to the continuous semantic space of the LLM. In addition, we further probe the decoder-only architecture for speech-to-text tasks by training a smaller scale randomly initialized speech-LLaMA model from speech-text paired data alone. We conduct experiments on multilingual speech-to-text translation tasks and demonstrate a significant improvement over strong baselines, highlighting the potential advantages of decoder-only models for speech-to-text conversion.

***Index Terms*—** decoder-only, LLaMA, LoRA, speech translation

1. INTRODUCTION

In recent times, the large language models (LLMs) have showcased remarkable achievements across various natural language benchmarks, encompassing question answering, machine translation, language understanding and more [1, 2, 3, 4, 5]. By employing a Transformer-based architecture [6] and training to anticipate forthcoming tokens within a sequence, this language model excels in contextual learning abilities. Not only does this significantly enhance its modeling prowess, but more importantly, it enables seamless user interaction that effectively connects cutting-edge research with real-world applications.

As speech represents the most innate and instinctive mode of human communication, integrating speech and LLMs will further boost the user experience of human-machine interaction. Based on this intuition, several attempts in combining speech signals and large language models were carried out [7, 8, 9, 10]. Among them, the cascaded approach is the most straightforward solution. In these systems, the speech signal is firstly transformed into word tokens through existing automatic speech recognition (ASR) [11] models, and LLM processes the recognized words for downstream tasks. Later, inspired by the integration of image information to LLMs [12, 13, 14, 15], researchers also explored the deep combination of speech signals [9, 10, 16, 17, 18, 19]. In [16], the authors proposed to jointly model the speech and text tasks through a unified decoder only network. Similarly, in [19], the authors proposed to optimize the audio token conversion module together with a off-the-shelf LLM. Instead of word pieces, discrete tokens of speech representation from a self-supervised model are used in [17].

While there have been promising outcomes, several crucial challenges regarding the integration of speech and LLMs still require further exploration. Initially, aligning the two modalities (speech and text) using a pretrained LLM poses challenges due to the typically longer sequence length of speech signals compared to text sequences. Moreover, given the costly nature of training LLMs, finding ways to minimize the overall integration cost while maintaining exceptional performance continues to be a challenging task. More importantly, considering the remarkable success of the LLMs, it is crucial to explore the untapped potential of using a decoder-only model [3, 16, 20, 21, 22] as the backbone network architecture for speech to text processing.

In this study, we aim to tackle the aforementioned challenges by exploring an efficient end-to-end integration of speech and language models. Our approach involves designing a simple yet effective architecture where a large language model that operates on text also incorporates acoustic embeddings. This integration enables the LM to condition its transcription or translation of the acoustic information. More specifically, our proposed method utilizes a pre-existing LLM and incorporates a acoustic feature compressor and an acoustic encoder introducing only a small number of free parameters. Diverging from previous approaches that convert speech into discretized tokens, our model directly maps the continuous representation of speech into the semantic space defined by the LM. During the processing stage, the speech feature is

initially compressed by the acoustic compressor to reduce the sequence length. Subsequently, the acoustic encoder transforms the compressed speech signal into continuous vectors in the same semantic space of the text that can be consumed by the LLM. The final output is generated through the decoding process of the LLM.

We thoroughly investigate various practical aspects of our proposed model, such as selecting the appropriate acoustic compressor, attention mask, and fine-tuning methods. Additionally, we apply the proposed model to the task of translating speech in 13 different languages into English (EN) text and compare its performance against a strong baseline on CoVoST dataset. Finally, we demonstrate that the decoder-only model, even trained from scratch using only speech-text paired data, exhibits significant potential and several advantages over the commonly employed encoder-decoder architecture in speech processing. In this work, our contribution can be summarized as follows:

- We introduce an efficient end-to-end integration method called Speech-LLaMA, which effectively integrates existing text-based large language models with speech processing. We have achieved substantial improvements in translation performance compared to strong baselines on various speech translation (ST) tasks.

- We investigate various practical aspects of the proposed speech-LLM integrations that are crucial for enhancing performance. These aspects include acoustic compression of the acoustic feature, attention mask selection, and fine-tuning strategy.

- On large, diverse and real-world data, we show that the decoder-only architecture can be as competitive as the encoder-decoder architecture for speech-to-text tasks. We show that decoder-only to also be more parameter efficient.

2. RELATED WORK

Our model aims at integrating speech signals into large language models, as well as relates to Connectionist Temporal Classification (CTC) feature length compression and low-rank adaptation (LoRA). We discuss these topics in the following.

2.1. Large language models

LLMs are generally pre-trained on vast amounts of textual data that span a wide variety of domains and languages. They usually consist of a stack of transformer layers, following an auto-regressive decoder-only architecture, where each output token is used as the input to predict the next step token. In this work, we select LLaMA-7B [5] as the backbone LLM to build the proposed method. LLaMA-7B model consists of 32 Transformer encoder layers with 32 heads and 4096 attention dimension. The tokenizer from the LLaMA work has a vocabulary size of 32,000 which covers a group of languages.

2.2. CTC compressor

Connectionist Temporal Classification (CTC) compressor [23] was proposed to reduce the sequence length via removing the redundant information in the features. It was applied in speech translation task and was shown to yield better memory consumption and performance. The method adds a linear CTC branch in a middle layer of the encoder which is jointly optimized with the main cross-entropy criteria . The hidden representations of the CTC branch are then compressed according to the distributions of the CTC posteriors and are passed to the succeeding layers. The author investigated a few variations within this method of sequence length compression. They found that averaging the consecutive hidden representations (corresponding to consecutive CTC predictions belonging to the same class) gives the best performance.

2.3. LoRA

Low-Rank Adaptation (LoRA) [24] is a commonly used technology to adapt the large models for new datasets or tasks. It introduces a small amount of free parameters to each Transformer layer of the source large model, while freezing all the original model parameters. Specifically, for each weight matrix $W \in \mathbb{R}^{d\times k}$ in a Transformer layer, 2 new matrices $W_a \in \mathbb{R}^{d\times r}$ and $W_b \in \mathbb{R}^{r\times k}$ are introduced such that $r \ll \min\{d, k\}$. For each matrix multiplication during training, the input x is firstly multiplied with both original weight W and its introduced low-rank approximation W_a, W_b, then the two outputs are summed to form the output for later computation. Only W_a and W_b are updated during fine-tuning while W keeps frozen, thus significantly reducing the memory footprint during training.

3. OUR APPROACH

In this work, we design an architecture named Speech-LLaMA where a text-LLM can also accept acoustic embedding as well as text as conditional prompts for text generation. By converting the speech input to a sequence of acoustic embeddings within the same space of the text embeddings, in the aspect of both length and semantics, the pre-trained text LLM can leverage its in-context learning capacity to absorb the speech signal and output corresponding text for speech translation task.

Overall, given the text prompt $\mathbf{p}$ and audio signals $\mathbf{x}$, the generation of the corresponding text sequence $\mathbf{y} = \{y_0, y_1, \cdots, y_{N-1}\}$ with a text-LLM is formulated as:

$$p(\mathbf{y}|\mathbf{p}, \mathbf{x}; \Theta_{\text{LLM}}) = \prod_{n=0}^{N-1} p(y_n|\mathbf{y}_{<n}, \mathbf{p}, \mathbf{x}; \Theta_{\text{LLM}}) \quad (1)$$

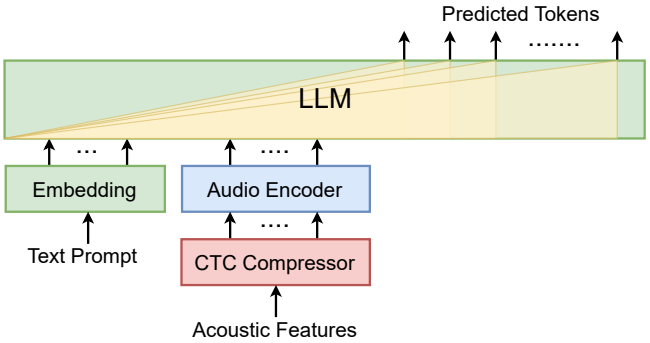

Fig. 1. High-level architecture of our proposed approach with LLM. The green blocks indicate the part of the LLM. In this work, we only learn parameters in the "Audio Encoder", keeping everything else frozen.

where $\mathbf{y}_{<n}$ indicates the generated text sequence before y_n.

Overview Our proposed neural model consists of three distinct parts: a pre-trained text neural LLM, an audio encoder and a CTC compressor, as shown in Figure 1. The text-LLM in our case is a LLaMA-7B [14] but this method can be generalized to LLMs of any scale. The CTC compressor reduces the sequence length of the input speech filter-bank to match the length of the text, and the audio encoder transforms the compressed speech signal into continuous vectors in the LLM's semantic space.

CTC compressor Different from the prior work that trained the CTC compressor jointly with the main task [23], our CTC compressor is a pre-trained module, aiming to match the audio and the text duration to the same scale by selecting the representative frames from the audio signal. In this work, we explore two ways to reduce the sequence length of the acoustic features in the CTC compressor: "blank-removal" and "frame-averaging". For "blank-removal", we simply discard all the frames that predicted the blank symbol according to the distribution of the CTC posteriors. On the other hand, for "frame-averaging", we average the hidden states of consecutive frames without blank frames removed, once their CTC predictions belong to the same class.

Audio encoder The audio encoder is used to bridge representations generated from the CTC compressor to the text embeddings of the text-LLM. This module is designed to be relatively small in size and is initialized with random weights. During the fine-tuning process, the audio encoder is optimized to effectively integrate the audio information within the LLM, enhancing the overall performance of the system. Different from the methods in [7, 19], where the audio encoder is trained to firstly map the speech signal into discrete tokens, which is then consumed by LLM, the proposed audio encoder is directly optimized to map the compressed acoustic signal to the continuous semantic space of LLM, allowing a deep integration between the audio encoder and the language model.

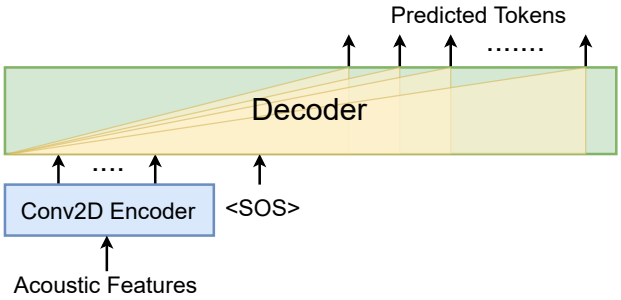

Fig. 2. The architecture of the decoder-only model for the from-scratch training. We use ⟨SOS⟩ token to indicate the starting of the text generation.

Instruct learning For each training sample, we prepend a text prompt that briefly describes the task, e.g., "`audio` ⇒ `English`" and "`transcribe the audio into English`". The text prompt are sampled from a pre-defined list, where some prompts contains the source language ID following the format "`translate` [`source`] `audio into English`". During evaluation, we fix the text prompt as "`translate the audio into English`" for all testing samples.

LoRA fine-tuning On top of the proposed model, we apply the LoRA to four attention matrices in each layer of the LLaMA Transformer (e.g., W_q, W_k, W_v, W_o). To stabilize the training, we adopt a two-stage training scheme which means we train the audio encoder firstly with the CTC compressor and LLaMA frozen and then introduce LoRA to the well-trained model and perform the second stage optimization. The entire system is still trained with cross-entropy loss between the LLM output and the reference transcription sequence on the same training data.

From-scratch training To further explore the potential of decoder-only architecture as a foundational architecture for speech modeling, we also include a "from-scratch" training of a decoder-only architecture. Here, we replace the text prompt, audio encoder, and CTC compressor with a randomly initialized convolutional 2D encoder. We also replace the pretrained LLaMA network with a much smaller randomly initialized autoregressive network. This architecture is shown in Figure 2. We add an ⟨SOS⟩ token at the end of the acoustic sequence to indicate the starting of the generation. In this case, the generation of the text sequence $\mathbf{y}$ with a decoder-only model is conditioned purely on audio signal $\mathbf{x}$ and previously generated text sequence $\mathbf{y}_{<n}$:

$$p(\mathbf{y}|\mathbf{x};\Theta_{\text{DEC}}) = \prod_{n=0}^{N-1} p(y_n|\mathbf{y}_{<n},\mathbf{x};\Theta_{\text{DEC}}) \qquad (2)$$

where Θ_{DEC} refers to the parameters of the decoder model.

4. EXPERIMENTS

The speech translation (ST) task [25, 26, 27, 28, 29] has been chosen as the primary evaluation benchmark for assessing the proposed methods. In this task, the goal is to develop a system that can accurately translate spoken language from 13 source languages to English.

4.1. Data and metric

The 13 source languages we want to translate to EN are German (DE), Chinese (ZH), Arabic (AR), Spanish (ES), French (FR), Italian (IT), Dutch (NL), Japanese (JA), Russian (RU), Portuguese (PT), Estonian (ET), Swedish (SV) and Slovenian (SL). We chose these languages based on availability of training and testing data. The training data for each language contains 1K hours of in-house speech data. To make the model more robust, we also include 1K hours of EN data, bringing the total to 14K hours. The original source transcriptions for non-English speech utterances are fed into an in-house translation service to generate the corresponding English transcriptions with both punctuation and capitalization. Those pseudo-label English transcriptions are used as the target transcription for ST task training. All the training data was anonymized with personally identifiable information removed.

Our speech translation models are evaluated on the 13 languages from the above list. The corresponding test sets are selected from CoVoST 2 dataset [30]. We evaluate the BLEU [31] scores for the performance comparison.

4.2. Models configuration

4.2.1. CTC compressor

The CTC compressor contains 2 convolution-2D layers followed by 4 Transformer layers for a 4-times subsampling with 15.8M parameters in total. Each transformer layer has a 512-dimensional self-attention module with 8 heads and a 2048 dimensional feed-forward network (FFN). Each convolution 2D layer has a stride size of 2 and kernel size of 3. We pre-trained CTC compressor with the paired speech and text data (i.e., ASR task) from 13 languages using the CTC objective function because that in our preliminary experiments, the BLEU score with ASR task training is much better than the one with the ST task training. Once CTC compressor is trained, the parameters are frozen during later training stages.

For comparison, we include a convolution-based subsampling module as a baseline, which shares the same architecture with the CTC compressor but with additional 3 1D convolution layers on top, allowing $4 \times 8 = 32$ times feature length reduction in total. The convolution-based subsampling is jointly trained with the audio encoder parameters.

4.2.2. Audio encoder

The audio encoder consists of 4 Transformer layers, where each layer has the same setting as in the CTC compressor except that the output tensor of the last layer is converted to the dimension of 4096, in order to match the dimension of semantic embedding in LLaMA.

For each training sample, we concatenate the embeddings of the text prompt and representations from the audio encoder along the time axis and use that as the prefix feature sequence to feed to the LLaMA model to generate the target language (EN) transcriptions.

Two attention mask strategies are explored within the LLaMA model. The first follows the language model training, where a causal, i.e., lower triangle attention mask is applied for each transformer layer to constrain the self-attention to not look into the future. As the proposed model is "non-streaming" in nature, we also explore a non-causal full attention mask strategy for the prefix part only [32], i.e., text prompt and audio encoder representations, to enable the full context learning on the acoustic information.

4.2.3. LoRA fine-tuning

We simply choose rank value of 2 for LoRA fine-tuning experiments according to the results of the LoRA work [24], i.e., 8 rank-2 matrices in the shape of 2×4096 are introduced to each LLaMA Transformer layer as an adaptor, which results in 2.1M more parameters in total. The LoRA fine-tuning is conducted on a well-trained Speech-LLaMA model, where the CTC compressor and the LLaMA parameters are frozen. We still update the audio encoder to learn better representations together with the adapted LLaMA.

4.2.4. Baseline

We adopt a seq2seq [33, 34] based speech translation model as a baseline. More specifically, we use the Whisper [35] architecture with 240M parameters and train it on the 14K hour data mentioned in Section 4.1. It contains a 12-layer audio Transformer encoder and a 12-layer text Transformer encoder, where the attention dimension and head number is 768 and 12, respectively. We optimize the model using cross-entropy as the primary objective function but also augment this architecture with a CTC loss on the encoder. We train the whole network end-to-end in a multi-task fashion. Please note that, for a fair comparison, we start the seq2seq training from scratch and do not initialize with pretrained open source Whisper weights. During beam search inference, we do a joint-decoding (prefix-decoding) [36] with CTC. To make the comparison with LLaMA boot-strapped models more appropriate, we also present results with n-best rescoring of the seq2seq model with LLaMA. To accomplish that, we do a simple log-linear interpolation between the scores from seq2seq and LLaMA for each of the n-best hypotheses

Table 1. BLEU scores of the 13 languages on the baseline and the proposed models.

Model	Seq2seq		Decoder-only	Speech-LLaMA						
ID	B1	B2	D1	E0	E1	E2	E3	E4	E5	E6
Compressor	–	–	–	×	CTC (remove)		CTC (average)			
Learnable #Param.	240M		150M	29M	14M		14M	16.1M	14M	16.1M
Prefix Non-causal Mask	–	–	✓	×	×	✓	×	×	✓	✓
LoRA	–	–	–	×	×	×	×	E3	×	E5
LLaMA Rescore	–	✓	–	–	–	–	–	–	–	–
AR	22.8	24.9	21.2	16.9	24.6	24.7	24.6	26.3	25.9	28.2
DE	22.6	23.6	21.3	16.9	22.6	22.8	24.3	26.0	25.4	27.1
ZH	7.0	7.2	6.7	3.4	9.6	10.1	10.1	11.4	10.8	12.3
ES	23.7	24.9	22.7	19.6	23.5	24.0	25.4	27.3	26.2	27.9
FR	21.8	22.7	20.6	15.4	20.9	21.1	22.6	24.5	23.2	25.2
IT	20.7	21.6	19.8	16.7	21.4	21.0	23.7	25.3	24.0	25.9
NL	34.6	36.0	35.2	28.3	32.4	35.0	34.1	36.0	34.9	36.5
JA	15.3	15.7	16.3	10.3	17.5	17.1	17.7	19.8	19.2	19.9
RU	26.4	27.7	26.0	22.8	31.0	32.0	33.3	35.5	34.3	36.8
PT	28.9	30.2	27.2	22.8	26.8	27.7	29.2	31.3	30.2	32.0
ET	9.4	9.4	7.4	15.4	17.0	18.3	17.2	18.1	18.0	18.7
SV	24.4	25.6	27.5	26.3	25.3	28.8	26.7	27.4	27.2	29.0
SL	13.3	12.7	13.3	22.2	20.3	22.9	22.8	22.2	22.1	22.7
Average BLEU ↑	**20.8**	**21.7**	**20.4**	**18.2**	**22.5**	**23.5**	**24.0**	**25.5**	**24.7**	**26.3**

and then re-rank accordingly. We use $n = 5$ for seq2seq beam-search decoding and the re-ranking experiments.

4.2.5. From-scratch training

In this setting, the structure of the convolutional 2D encoder contains 2 convolutional layers which is the same as the one in the CTC compressor, which introduces a 4-times subsampling rate. For the Transformer decoder, we follow the implementation of LLaMA, where pre-normalization, SwiGLU activation function [37] and rotary positional embeddings (RoPE) [38] are adopted. Similar to the configuration of the seq2seq baseline, each decoder layer contains a 12-head self-attention module with the 768 attention dimension. The dimension of the feed-forward network is set as 4076.

4.3. Training and evaluation

We extract an 80-dim log mel-filterbank using 25 msec window and 10 msec hop size as the acoustic features. Global mean and variance normalization is applied. All models were trained with AdamW optimizer [39] with $\beta_1 = 0.9$ and $\beta_2 = 0.98$ on 16 V100 GPUs and a warmup and linear decay learning rate strategy is used. Batch size varies with the model size. CTC compressor was trained for 100K steps with source language transcriptions, tokenized by LLaMA's tokenizer. The peak learning rate was set to 0.001. In the first stage training of Speech-LLaMA, We perform a 500K step training with a peak learning rate of 0.015 while in the later LoRA fine-tuning stage, we use additional 100K optimization steps with a peak learning rate of $2e^{-4}$. The from-scratch decoder-only models were trained with a peak learning rate of 0.001 for at most 300K steps. We use the beam search algorithm with a beam size 4 for the decoding of all the decoder-only models, unless noted otherwise. Both seq2seq and decoder-only models use English-only byte pair encoding (BPE [40]) model for the tokenization which has a vocabulary size of 5,857 while the Speech-LLaMA models keep using LLaMA's tokenizer.

5. RESULTS AND DISCUSSIONS

The results of the experiments are presented in Table 1, where several observations can be gleaned.

5.1. Baselines

For baselines, we report results on 2 systems. B1 is a seq2seq model described in Section 4.2.4 and B2 is B1 with LLaMA n-best rescoring. As expected [41], a 0.9 better BLEU score can be observed from B2 system over B1. This suggests that shallow integration with LLM can still bring benefits to the speech models.

5.2. Deeper integration with LLaMA

While shallow integration can boost performance, the gains using a deep integration technique like Speech-LLaMA

should be much higher. Systems E1 $\sim$ E6 describe Speech-LLaMA models in various configurations. We can find all Speech-LLaMA configurations significantly outperform the baselines with the limited learnable parameters, resulting in up to 4.6 absolute BLEU score improvement (21.2% relative). These results show the efficacy of the proposed system and also suggests the necessity for deeper integration between the speech models and text-LLMs.

5.3. CTC compressor

Results from system E0, E1 and E3 describe the importance of CTC compressor for audio length reduction, in our design. Comparing E1 over E0, we obtain consistently better performance showing the effectiveness of CTC compressor over the convolution one. This gain is despite the fact that CTC compressor is frozen during the training while the convolution compressor was fine-tuned with the rest of audio encoder. One hypothesis for the better performance of CTC compressor is that it leverages the transcription of each source language during pre-training stage as we also observe that replacing the current CTC compressor model with the one trained with ST labels brings worse BLEU scores in our preliminary experiments. This observation also suggests that a potentially better performance might be obtained if the source transcription is also used during the training stage. We leave this line of exploration for future works.

Within the CTC compressor, comparing system E3 over E1, the "frame-averaging" strategy shows a 1.5 better average BLEU score over "blank-removal" strategy. We believe that it is because the CTC compressor can't very reliably distill all relevant information into non-blank representations. Thus the frames selected by the CTC compressor might lose some acoustic information which cause the degradation of the performance. The averaging strategy is more robust to this compression error which aligns with the prior work [23].

5.4. Effect of non-causal attention mask

It is expected that the full attention mask over text prompt and acoustic representations would usually result in better speech representation, and consequently better results. For each type of CTC compression strategy, our experiments demonstrate that using a non-causal attention mask over a causal mask can indeed bring gains. Comparing system E2 over E1, we see that switching to a non-causal mask brings an additional gain of 1.5 average BLEU score when using the "blank-removal" strategy within CTC compressor. Similarly, comparing systems E5 over E3, we again observe a gain of 0.7 average BLEU score, when using the "frame-averaging" strategy within CTC compressor. The gain with non-causal mask is understandably larger in "blank-removal" strategy, since future acoustic information can help compensate for potential loss of information caused due to removal of frames corresponding to the blank symbol of the CTC loss. Even in LoRA fine-tuning systems, e.g., comparing E6 and E4, we can still observe a gain of 0.8 average BLEU score with non-causal mask applied.

5.5. LoRA fine-tuning

E4 and E6 represent our systems with LoRA fine-tuning. Comparing E4 over E3 shows the gains using LoRA fine-tuning when using a causal attention mask while comparing E6 over E5 show corresponding gains when using a non-causal attention mask. We can obtain an additional increase of 1.5 and 1.6 average BLEU score, respectively. Note that only 2.1M additional parameters are added as adaptors. Potentially better performance might be observed when larger rank is used. We leave this exploration for future works.

5.6. Decoder-only vs Encoder-Decoder

Finally, the results for the randomly initialized decoder-only model are shown as system D1 in Table 1. This model achieves only slightly worse (0.4 lower BLEU score) performance compared to the seq2seq baseline. But the total parameter for the decoder-only model in our study is also significantly lower than the seq2seq baseline. We think that decoder-only architecture can be more parameter efficient than the encoder-decoder architecture. This is because a single module is used to learn representations for both source and target sequences in the former while separate modules (encoder *and* decoder) are used to generate representations for source and target sequences in the latter. This sharing of parameters to process input and output jointly can bring out better parameter efficiency in the decoder-only architecture. Our results do seem to validate this theory. In future, we will conduct more extensive analysis of how model size effects performance in these 2 architectures.

6. CONCLUSION & FUTURE WORK

In this work, we propose a method to infuse an off-the-shelf large language model with acoustic information. The proposed model presents a deep integration between the audio with the LLM by directly mapping acoustic representation into the semantic space of LLM. We also explore several practical aspects of the proposed model for better performance including compression of the acoustic feature, attention mask design and adapter fine-tuning. We show that on a 13 language to English speech translation task, the proposed model significantly outperforms a strong sequence-to-sequence baseline model. We also show that the decoder-only architecture, trained from scratch, can achieve comparable performance with around 40% fewer parameters, which verifies the potential of decoder-only models for general speech-to-text modeling.

7. REFERENCES

[1] OpenAI, "Introducing chatgpt," *OpenAI Blog*, 2022.

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

[3] Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al., "Language models are few-shot learners," *Advances in neural information processing systems*, vol. 33, pp. 1877–1901, 2020.

[4] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al., "Palm: Scaling language modeling with pathways," *arXiv preprint arXiv:2204.02311*, 2022.

[5] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al., "Llama: Open and efficient foundation language models," *arXiv preprint arXiv:2302.13971*, 2023.

[6] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Lukasz Kaiser, and Illia Polosukhin, "Attention is all you need," *Advances in neural information processing systems*, vol. 30, 2017.

[7] Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, et al., "Audiogpt: Understanding and generating speech, music, sound, and talking head," *arXiv preprint arXiv:2304.12995*, 2023.

[8] Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang, "Hugginggpt: Solving ai tasks with chatgpt and its friends in huggingface," *arXiv preprint arXiv:2303.17580*, 2023.

[9] Feilong Chen, Minglun Han, Haozhi Zhao, Qingyang Zhang, Jing Shi, Shuang Xu, and Bo Xu, "X-llm: Bootstrapping advanced large language models by treating multi-modalities as foreign languages," *arXiv preprint arXiv:2305.04160*, 2023.

[10] Soham Deshmukh, Benjamin Elizalde, Rita Singh, and Huaming Wang, "Pengi: An audio language model for audio tasks," *arXiv preprint arXiv:2305.11834*, 2023.

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

[12] Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny, "Minigpt-4: Enhancing vision-language understanding with advanced large language models," *arXiv preprint arXiv:2304.10592*, 2023.

[13] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee, "Visual instruction tuning," *arXiv preprint arXiv:2304.08485*, 2023.

[14] Peng Gao, Jiaming Han, Renrui Zhang, Ziyi Lin, Shijie Geng, Aojun Zhou, Wei Zhang, Pan Lu, Conghui He, Xiangyu Yue, et al., "Llama-adapter v2: Parameter-efficient visual instruction model," *arXiv preprint arXiv:2304.15010*, 2023.

[15] Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, et al., "Flamingo: a visual language model for few-shot learning," *Advances in Neural Information Processing Systems*, vol. 35, pp. 23716–23736, 2022.

[16] Tianrui Wang, Long Zhou, Ziqiang Zhang, Yu Wu, Shujie Liu, Yashesh Gaur, Zhuo Chen, Jinyu Li, and Furu Wei, "Viola: Unified codec language models for speech recognition, synthesis, and translation," *arXiv preprint arXiv:2305.16107*, 2023.

[17] Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu, "Speechgpt: Empowering large language models with intrinsic cross-modal conversational abilities," *arXiv preprint arXiv:2305.11000*, 2023.

[18] Eliya Nachmani, Alon Levkovitch, Julian Salazar, Chulayutsh Asawaroengchai, Soroosh Mariooryad, RJ Skerry-Ryan, and Michelle Tadmor Ramanovich, "Lms with a voice: Spoken language modeling beyond speech tokens," *arXiv preprint arXiv:2305.15255*, 2023.

[19] Paul K Rubenstein, Chulayuth Asawaroengchai, Duc Dung Nguyen, Ankur Bapna, Zalán Borsos, Félix de Chaumont Quitry, Peter Chen, Dalia El Badawy, Wei Han, Eugene Kharitonov, et al., "Audiopalm: A large language model that can speak and listen," *arXiv preprint arXiv:2306.12925*, 2023.

[20] Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, et al., "Neural codec language models are zero-shot text to speech synthesizers," *arXiv preprint arXiv:2301.02111*, 2023.

[21] Ziqiang Zhang, Long Zhou, Chengyi Wang, Sanyuan Chen, Yu Wu, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei, "Speak foreign languages with your own voice: Cross-lingual neural codec language modeling," 2023.

[22] Zihao Fu, Wai Lam, Qian Yu, Anthony Man-Cho So, Shengding Hu, Zhiyuan Liu, and Nigel Collier, "Decoder-only or encoder-decoder? interpreting language model as a regularized encoder-decoder," *arXiv preprint arXiv:2304.04052*, 2023.

[23] Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi, "Ctc-based compression for direct speech translation," *arXiv preprint arXiv:2102.01578*, 2021.

[24] Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen, "Lora: Low-rank adaptation of large language models," *arXiv preprint arXiv:2106.09685*, 2021.

[25] Laura Cross Vila, Carlos Escolano, José AR Fonollosa, and Marta R Costa-Jussa, "End-to-end speech translation with the transformer.," in *Proceedings of Interspeech*, 2018, pp. 60–63.

[26] Matthias Sperber and Matthias Paulik, "Speech translation and the end-to-end promise: Taking stock of where we are," in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020, pp. 7409–7421.

[27] Jian Xue, Peidong Wang, Jinyu Li, and Eric Sun, "A weakly-supervised streaming multilingual speech model with truly zero-shot capability," *arXiv preprint arXiv:2211.02499*, 2022.

[28] Jian Xue, Peidong Wang, Jinyu Li, Matt Post, and Yashesh Gaur, "Large-scale streaming end-to-end speech translation with neural transducers," *INTERSPEEcH*, vol. abs/2204.05352, 2022.

[29] Peidong Wang, Eric Sun, Jian Xue, Yu Wu, Long Zhou, Yashesh Gaur, Shujie Liu, and Jinyu Li, "Lamassu: Streaming language-agnostic multilingual speech recognition and translation using neural transducers," *INTERSPEECH*, 2023.

[30] Changhan Wang, Anne Wu, and Juan Pino, "Covost 2: A massively multilingual speech-to-text translation corpus," 2020.

[31] Matt Post, "A call for clarity in reporting BLEU scores," in *Proceedings of the Third Conference on Machine Translation: Research Papers*, Brussels, Belgium, Oct. 2018, pp. 186–191, Association for Computational Linguistics.

[32] Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu, "Exploring the limits of transfer learning with a unified text-to-text transformer," *The Journal of Machine Learning Research*, vol. 21, no. 1, pp. 5485–5551, 2020.

[33] A. Berard, O. Pietquin, C. Servan, and L. Besacier, "Listen and translate: A proof of concept for end-to-end speech-to-text translation," in *NIPS Workshop on End-to-end Learning for Speech and Audio Processing*, 2016.

[34] Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen, "Sequence-to-sequence models can directly translate foreign speech," *Proc. Interspeech*, pp. 2625–2629, 2017.

[35] Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever, "Robust speech recognition via large-scale weak supervision," in *International Conference on Machine Learning*. PMLR, 2023, pp. 28492–28518.

[36] Takaaki Hori, Shinji Watanabe, and John Hershey, "Joint CTC/attention decoding for end-to-end speech recognition," in *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Vancouver, Canada, July 2017, pp. 518–529, Association for Computational Linguistics.

[37] Noam Shazeer, "Glu variants improve transformer," *arXiv preprint arXiv:2002.05202*, 2020.

[38] Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu, "Roformer: Enhanced transformer with rotary position embedding," *arXiv preprint arXiv:2104.09864*, 2021.

[39] Ilya Loshchilov and Frank Hutter, "Decoupled weight decay regularization," *arXiv preprint arXiv:1711.05101*, 2017.

[40] Taku Kudo and John Richardson, "Sentencepiece: A simple and language independent subword tokenizer and detokenizer for neural text processing," *arXiv preprint arXiv:1808.06226*, 2018.

[41] Yuang Li, Yu Wu, Jinyu Li, and Shujie Liu, "Prompting large language models for zero-shot domain adaptation in speech recognition," *arXiv preprint arXiv:2306.16007*, 2023.