Wav2Prompt: End-to-End Speech Prompt Learning and Task-based Fine-tuning for Text-based LLMs

Keqi Deng, Guangzhi Sun, Philip C. Woodland
Department of Engineering, University of Cambridge, Trumpington St., Cambridge, UK.
{kd502, gs534, pw117}@cam.ac.uk

Abstract

Wav2Prompt is proposed which allows integrating spoken input with a text-based large language model (LLM). Wav2Prompt uses a straightforward training process with only the same data used to train an automatic speech recognition (ASR) model. After training, Wav2Prompt learns continuous representations from speech and uses them as LLM prompts. To avoid task over-fitting issues found in prior work and preserve the emergent abilities of LLMs, Wav2Prompt takes LLM token embeddings as the training targets and utilises a continuous integrate-and-fire mechanism for explicit speech-text alignment. Therefore, a Wav2Prompt-LLM combination can be applied to zero-shot spoken language tasks such as speech translation (ST), speech understanding (SLU), and spoken-query-based question answering (SQQA). It is shown that for these zero-shot tasks, Wav2Prompt performs similarly to an ASR-LLM cascade and better than recent prior work. If relatively small amounts of task-specific paired data are available, the Wav2Prompt-LLM combination can be end-to-end (E2E) fine-tuned and then yields greatly improved results relative to an ASR-LLM cascade for the above tasks. For instance, for English-French ST, a Wav2Prompt-LLM combination gave a 5 BLEU point increase over an ASR-LLM cascade.

1 Introduction

Text-based large language models (LLMs) (Brown et al., 2020; Touvron et al., 2023a,b; Ouyang et al., 2022) have achieved remarkable performance in a wide range of natural language processing (NLP) tasks (Achiam et al., 2024). LLMs are trained on huge quantities of text and are highly flexible. They are able to be applied to a range of tasks for which they have not been explicitly trained, known as the emergent abilities of LLM (Wei et al., 2022; Tang et al., 2024). To further expand the use-cases of LLMs, it is important to enable LLMs to handle other modalities including spoken input.

The conventional approach is to use an automatic speech recognition (ASR) model to transcribe speech into text, which is then used as the LLM input. However, this cascaded system suffers from error accumulation and can not be end-to-end (E2E) fine-tuned. (Fathullah et al., 2024b). Many studies have explored connecting LLMs directly to the speech acoustic encoder (Encoder-LLM) for various speech tasks such as ASR or speech translation (ST) (Fathullah et al., 2024a; Wu et al., 2023; Yu et al., 2024; Chen et al., 2023). However, these approaches restrict the system to a specific task, thereby losing the ability of LLMs to handle a wide range of zero-shot spoken language tasks. Recent work has begun to explore ways to restore the zero-shot capabilities of LLMs. Work in this area includes approaches that make use of audio or speech-based question-answering (QA) tasks (Gong et al., 2024; Fathullah et al., 2024b), as well as the introduction of additional steps in the training pipeline, such as multi-task training (Rubenstein et al., 2023), instruction tuning (Chu et al., 2024; Zhang et al., 2023a; Das et al., 2024), and activation tuning (Tang et al., 2024). However, these approaches greatly complicate model training.

Unlike previous work, this paper enables text-based LLMs to understand speech and perform various untrained speech tasks using a straightforward single-task training process using only easily accessible ASR data. Wav2Prompt allows spoken input to be integrated with an off-the-shelf text-based LLM. After training using ASR data (i.e. speech and associated text transcript), Wav2Prompt generates representations from speech and uses them as LLM prompts for downstream tasks. It allows the Wav2Prompt-LLM combination to work well in a range of zero-shot spoken language tasks. However, Wav2Prompt can also give much improved performance when limited task-specific spoken lan-

Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers), pages 6940–6956
April 29 - May 4, 2025 ©2025 Association for Computational Linguistics

guage data is available through E2E fine-tuning, without updating the LLM. Wav2Prompt serves as an E2E alternative for the ASR-LLM cascade.

Wav2Prompt takes LLM token embeddings as training targets to naturally maintain the zero-shot capability of text-based LLMs. This is a key difference to the conventional ASR task, which takes discrete text tokens as the only target and thus leads to the ASR-LLM cascade approach. However, learning LLM token embeddings is challenging for speech models given the difference in input sequence length between speech and text. Wav2Prompt addresses this issue using a continuous integrate-and-fire (CIF) (Dong and Xu, 2020) mechanism to generate a label-level speech representation and a mean squared error (MSE) loss can be used to enforce consistency with the LLM token embeddings. Wav2Prompt can therefore be combined with a text-based LLM not only for zero-shot speech tasks, but also for E2E fine-tuning with limited task-specific data, which is a key advantage compared to an ASR-LLM cascade.

This paper evaluates Wav2Prompt on diverse spoken language tasks including speech translation (ST), spoken language understanding (SLU), and spoken-query-based question answering (SQQA), all of these are unseen during training as Wav2Prompt only uses ASR training data. The results show that the Wav2Prompt-LLM combination could achieve similar performance to the ASR-LLM cascade in zero-shot cases and greatly surpasses the existing Encoder-LLM method (Fathullah et al., 2024a). In scenarios with limited task-specific available data, after E2E fine-tuning, Wav2Prompt shows improved performance over the ASR-LLM cascade for all of these tasks.

The main contributions of this paper can be summarised in three main parts:

- Wav2Prompt is proposed, which is, to the best of our knowledge, the first step towards using only ASR data to extend LLMs to a range of zero-shot spoken language tasks.

- Task over-fitting to training data is a key issue addressed by Wav2Prompt which has previously limited the application of acoustic encoder enabled LLMs to other spoken language tasks in a zero-shot fashion (Tang et al., 2024). This issue is analysed and it is shown that the key step to unlock zero-shot capability is learning LLM token embeddings.

- Wav2Prompt achieves similar performance to an ASR-LLM cascade in a range of zero-shot speech tasks. In scenarios with limited task-specific data, Wav2Prompt greatly surpasses the performance of an ASR-LLM cascade by leveraging the advantage of E2E fine-tuning.

2 Related Work

Text-based Large Language Model The evolution of text-based LLMs, exemplified by a large increase in model parameters and training data seen in GPT-3 (Brown et al., 2020) and PaLM (Chowdhery et al., 2023), has revolutionised NLP tasks. This progress has facilitated the development of advanced models such as GPT-4 (Achiam et al., 2024), showcasing the remarkable capabilities of LLMs in various domains. Alongside these advancements, “smaller" LLMs like LLaMa (Touvron et al., 2023a,b) have been introduced, achieving a better balance between performance and computational resources. There are several variant models such as Vicuna (Zhang et al., 2023b) developed from conversation-based fine-tuning and multi-lingual LLMs, e.g. BLOOM (Le Scao et al., 2023). One key aspect of LLMs is that they can exhibit remarkable performance in a range of tasks on which they have never been explicitly trained. Examples include zero-shot task transfer (Radford et al., 2021) and few-shot learning (Brown et al., 2020). This is sometimes known as the emergent abilities of LLMs (Tang et al., 2024; Ma et al., 2024a).

Speech-enabled Large Language Model While discrete speech tokens have been explored to build spoken generative LMs (Borsos et al., 2023; Wang et al., 2023), this paper focuses on utilising off-the-shelf text-based LLMs. Recently, several studies have worked on building speech-enabled LLMs to support direct speech input (Fathullah et al., 2024a; Wu et al., 2023; Yu et al., 2024; Chen et al., 2023; Huang et al., 2024). Since the speech input sequence is much longer than the corresponding text, different strategies have been studied for downsampling. (Fathullah et al., 2024a; Yu et al., 2024; Ma et al., 2024b) stacks the acoustic encoder output to achieve a fixed-rate length reduction. (Chen et al., 2023) treats multiple modalities as foreign languages, in which CIF is used to obtain the speech representation. (Yu et al., 2024) explored the use of Q-Former (Li et al., 2023) which transforms input sequences of varying lengths into fixed-length outputs. However, the emergent abilities of LLM are lost in this task-specific training. Unlike work such as (Chen et al., 2023), which focus on enabling LLMs to handle multi-modal inputs including speech, this paper focuses on maintaining the zero-shot abilities of LLMs. Some recent work explored regaining the zero-shot abilities of LLMs. In (Tang et al., 2024), an instruction tuning stage followed by an activation tuning stage was introduced after pre-training to alleviate task over-fitting. In addition, (Fathullah et al., 2024b) simulated a speech QA dataset from ASR data, finding that a speech-enabled LLM trained on it could handle spoken QA tasks and potentially other tasks like ST. Prior work extended speech capabilities for LLM, but they come at the cost of increased complexity and extensive resources based on full-parameter or parameter-efficient training. Since text-based LLMs have a fundamental connection with speech, Wav2Prompt focuses on training a model that can be combined and E2E fine-tuned with a text-based LLM through a straightforward process while keeping the LLM fixed.

Prompt Tuning Fine-tuning LLMs can be expensive. As an alternative, the prompting technique fixes all LLM parameters and uses a prompt to query LLMs (Liu et al., 2022). Early prompting uses simple keyword-based inputs or fill-in-the-blank style prompts (Gao et al., 2021; Schick and Schütze, 2020). For generative LLMs, natural language prompts can be used (Victor et al., 2022; Brown et al., 2020). However, these discrete prompts can result in sub-optimal performance in numerous cases (Shin et al., 2020; Liu et al., 2022). Instead, prompt tuning adds trainable continuous embeddings, i.e. continuous prompts, to the original input token embedding (Liu et al., 2024; Lester et al., 2021). During training, only the parameters of the continuous prompts are updated (Liu et al., 2022). This paper follows the prompt tuning approach, updating only the parameters of Wav2Prompt while keeping the LLM fixed when E2E fine-tuned on limited task-specific data.

3 Analysis of task over-fitting

Task over-fitting (Tang et al., 2024) occurs when a speech-enabled LLM can only perform tasks that are seen during supervised training and shows limited performance on unseen tasks. This section provides a detailed analysis of this issue which leads to the proposed Wav2Prompt method.

To connect a decoder-only LLM with speech input, speech representations can be prepended to the original text token embedding sequence and the LLM will be conditioned on these speech representations when predicting the next token in order to perform speech tasks. To be more specific, in the normal case with a text-based user-input prompt, the next token probabilities of an LLM can be formulated as:

$$p_n = p(y_n|\boldsymbol{Y}_{0:n-1}, \mathbf{P}, \Gamma) \tag{1}$$

where $\boldsymbol{Y}_{0:n-1} = ([sos], y_1, ..., y_{n-1})$ is the sequence of previously predicted tokens and y_n denotes the n-th token. $\mathbf{P} = (\boldsymbol{p}_1, \cdots, \boldsymbol{p}_m)$ is the text embedding sequence obtained by feeding the text-based user-input prompt into an LLM embedding layer. Γ denotes a task-specific prompt template that contains instructions. When a speech representation $\mathbf{S} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_t)$ is prepended to the text-based input as the prompt supervision to replace $\mathbf{P}$, Eq. 1 can be re-written as $\hat{p}_n = p(y_n|\boldsymbol{Y}_{0:n-1}, \mathbf{S}, \Gamma)$.

It is counter-intuitive to let a fixed text-based LLM attend to speech representation $\mathbf{S}$ as it has never seen speech input during pre-training. However, after E2E training on supervised data, prior work has shown connecting a fixed LLM with the acoustic encoder (referred to as Encoder-LLM) can perform ASR tasks (Fathullah et al., 2024a; Yu et al., 2024). Since the main building block of LLMs is the attention mechanism which is also the first module that interacts with inputs, the process is simplified to a single attention function $\text{Att}(Q, K, V)$ for theoretical analysis where the conclusions can be generalised to the entire LLM. In the normal case with text-based input, if the token embedding of y_{n-1} after the LLM embedding layer is denoted $\boldsymbol{z}_{n-1}$ and $\mathbf{Z} = (\boldsymbol{z}_1, \cdots, \boldsymbol{z}_{n-1})$, Eq. 1 can be expressed as:

$$\boldsymbol{l}_n = \text{Att}(\boldsymbol{z}_{n-1}, [\mathbf{Z}; \mathbf{P}; \Gamma], [\mathbf{Z}; \mathbf{P}; \Gamma]) \tag{2}$$

where output $\boldsymbol{l}_n$ can be used to compute p_n, $\boldsymbol{z}_{n-1}$ is the query, and $[\mathbf{Z}; \mathbf{P}; \Gamma]$ is the keys and values. A speech-enabled LLM must replace $\mathbf{P}$ in Eq. 2 with the speech representation $\mathbf{S}$ and make the resulting prediction $\hat{\boldsymbol{l}}_n = \text{Att}(\boldsymbol{z}_{n-1}, [\mathbf{Z}; \mathbf{S}; \Gamma], [\mathbf{Z}; \mathbf{S}; \Gamma])$ close to ground truth. During E2E supervised training, the cross-entropy loss supervises $\hat{\boldsymbol{l}}_n$ while updating $\mathbf{S}$. Even if the lengths and features of $\mathbf{S}$ and $\mathbf{P}$ are different, $\boldsymbol{l}_n = \hat{\boldsymbol{l}}_n$ is still possible to achieve. Since the attention mechanism is a weighted sum, as long as the weighted sum corresponding to the

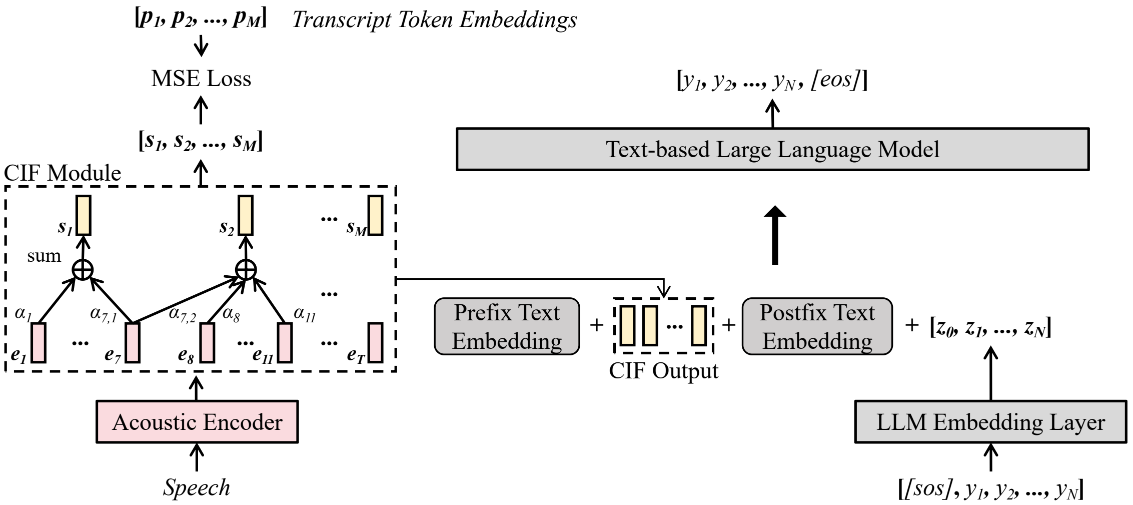

Figure 1: Illustration of the proposed Wav2Prompt architecture. ⊕ denotes addition. Prefix and postfix text are task-specific prompt templates that can contain instructions. Their embeddings are obtained through the LLM embedding layer, and the transcript token embeddings are the same.

use of **S** and **P** are consistent, correct predictions can be obtained.

However, z_{n-1} and Γ differ in different tasks, and **S** learned on a certain task cannot always guarantee that $\boldsymbol{l}_n = \hat{\boldsymbol{l}}_n$ will still hold with a different z_{n-1} and Γ for other tasks, leading to so-called task over-fitting. For example, preliminary experiments found that the Encoder-LLM-based ASR system has trouble following new instructions to perform zero-shot ST.

Prior work relies on task-specific E2E training to implicitly optimise **S** (Fathullah et al., 2024a; Yu et al., 2024; Chen et al., 2023), and through complex training pipelines gradually enables **S** to learn a correct alignment that can preserve the LLM zero-shot capability (Tang et al., 2024). However, to learn a correct **S**, there is already a feasible and clear target, which is the LLM token embedding given that LLM can always flexibly handle text inputs for zero-shot tasks. Unlike prior work, this paper proposes using LLM embeddings as the target to explicitly guide the learning of the speech representation **S**, which greatly simplifies the process and only requires ASR data for training.

4 Wav2Prompt

This paper proposes Wav2Prompt, a model that naturally enables text-based LLM to handle speech input while maintaining the zero-shot capabilities of the original LLM. Wav2Prompt provides a straightforward process that can be combined with LLM using only ASR data for training and does not require subsequent multi-stage tuning, it can serve as an E2E alternative approach that is superior to the conventional ASR-LLM cascade.

4.1 Wav2Prompt architecture

Wav2Prompt is illustrated in Fig. 1, which contains three main components: an acoustic encoder, a CIF module, and an LLM (including the LLM embedding layer). The acoustic encoder and the CIF module extract a label-level speech representation $\mathbf{S} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_M)$, which have the same length as the transcript text token embeddings so that mean squared error (MSE) loss can be used to enforce the representation consistency between them. This is one of the main differences from prior work (Fathullah et al., 2024a; Wu et al., 2023; Yu et al., 2024; Chen et al., 2023; Tang et al., 2024; Ma et al., 2024b), which instead simply down-sampled the acoustic encoder output before fed it into the LLM.

In this paper, LLM refers to a text-based decoder-only LLM. The output of the CIF module is used as the LLM prompt, while the LLM parameters are always kept fixed (shown in grey in Fig. 1) following prompt tuning (Liu et al., 2022).

The acoustic encoder employs a Conformer (Gulati et al., 2020) structure. Denote the Conformer-based encoder output as $\mathbf{E} = (\boldsymbol{e}_1, \cdots, \boldsymbol{e}_T)$, where T is the frame length and is normally much larger than the corresponding text sequence length. To learn a label-level speech representation with a flat-start, i.e. not relying on a readily available alignment, the CIF mechanism (Dong and Xu, 2020) is used. As shown in Fig. 1, a scalar weight α_t is learnt for each encoder output frame $\boldsymbol{e}_t$ and a label-level representation is obtained via weighted addition. Following (Deng and Woodland, 2024a,b), this paper uses the last dimension of $\boldsymbol{e}_t$ as the raw scalar attention value of α_t to avoid additional parameters: $\alpha_t = \text{sigmoid}(e_{t,d})$ where d is the dimension size of $\boldsymbol{e}_t$. The weights α_t are summed from left to right (i.e. forward through time) until the sum exceeds a threshold of 1.0. Once reached, the current weight α_t is divided into two parts: one part ensures the current accumulated weight is exactly 1.0, while the remainder is used for the next integration. An example is shown in Fig. 1, where the threshold 1.0 is achieved when $t = 7$ and α_7 is divided into $\alpha_{7,1}$ and $\alpha_{7,2}$. The first label-level speech representation $\boldsymbol{s}_1$ is obtained via:

$$\boldsymbol{s}_1 = \text{FC}(\alpha_1 \cdot \boldsymbol{e}_{1,1:d-1} + \cdots + \alpha_{7,1} \cdot \boldsymbol{e}_{7,1:d-1}) \quad (3)$$

where FC represents a fully connected layer that maps $\boldsymbol{e}_{t,1:d-1}$ to the LLM embedding dimension. The sum is then reset to zero and continues to the right to calculate $\boldsymbol{s}_2$, $\boldsymbol{s}_3$, etc. until the end of the encoder output. To ensure that the extracted label-level speech representation sequence $\mathbf{S}$ has the exactly same length M as the corresponding transcript token sequence at training, a scaled weight $\hat{\alpha}_t$=$\alpha_t \cdot (M/\sum_{i=1}^{T} \alpha_i)$ is computed and used to extract $\mathbf{S} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_M)$ instead α_t at training. To learn the CIF alignment, a quantity loss (Dong and Xu, 2020), $\mathcal{L}_{\text{qua}} = |\sum_{i=1}^{T} \alpha_i - M|$, is computed during training to encourage the accumulated weights approaching the correct length M.

The label-level speech representation from CIF is then fed into the LLM as a prompt along with task-specific prompt templates that contains instructions, denoted as prefix and postfix text in Fig. 1. Suppose $\textbf{emb}^{\text{pre}}$ and $\textbf{emb}^{\text{post}}$ are the embeddings of the prefix and postfix text sequence, the LLM output logits $\mathbf{L} = (\boldsymbol{l}_1, \cdots, \boldsymbol{l}_N)$ are computed as:

$$\mathbf{L} = \text{LLM}(\text{Concat}(\textbf{emb}^{\text{pre}}, \mathbf{S}, \textbf{emb}^{\text{post}}, \mathbf{Z})) \quad (4)$$

where $\mathbf{Z} = (\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_N)$ denotes the embeddings of the LLM input $([sos], y_1, ..., y_N)$ as shown in Fig. 1, Concat$(\cdot)$ denotes concatenation of vector sequences, and LLM$(\cdot)$ denotes the LLM function that takes the concatenated vector sequence as inputs and outputs logits $\mathbf{L}$.

4.2 Training

Wav2Prompt is trained using only ASR data. First, the scaled weight $\hat{\alpha}_t$ is used in training to extract the label-level speech representation $\mathbf{S} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_M)$ that has the same length as the transcript token sequence. After feeding the transcript text tokens into the LLM embedding layer, the embedding sequence $\mathbf{P} = (\boldsymbol{p}_1, \cdots, \boldsymbol{p}_M)$ is used as the training target of $\mathbf{S}$, and an MSE loss is computed:

$$\mathcal{L}_{\text{MSE}} = \sum_{m=1}^{M} \text{MSE}(\boldsymbol{s}_m, \boldsymbol{p}_m) \quad (5)$$

In addition, $\mathbf{P}$ is fed into the LLM and a cross-entropy (CE) loss $\mathcal{L}_{\text{CE}}$ is computed between the LLM output logits $\mathbf{L}$ and target transcripts, ensuring that the speech representation $\mathbf{S}$ can be interpreted by the fixed LLM. Finally, the quantity loss $\mathcal{L}_{\text{qua}}$ is also included to learn the CIF alignment as mentioned above. Therefore, the overall training objective $\mathcal{L}_{\text{Train}}$ of Wav2Prompt is:

$$\mathcal{L}_{\text{Train}} = \mathcal{L}_{\text{CE}} + \gamma\mathcal{L}_{\text{MSE}} + \mu\mathcal{L}_{\text{qua}} \quad (6)$$

where γ and μ are hyper-parameters.

4.3 Zero and limited resource task application

Wav2Prompt is trained solely on the ASR task but can be combined with a text-based LLM (Wav2Prompt-LLM) to perform other speech tasks in zero-shot settings or via E2E fine-tuning with limited data. In this paper, zero-shot refers to not using any task-specific paired data for fine-tuning.

Zero-shot Wav2Prompt preserves the flexible zero-shot capabilities of LLMs. With Wav2Prompt, the generated label-level speech representation $\mathbf{S}$ is fed into the LLM as a prompt, and only the instructions in the prefix and postfix text (as in Fig. 1) need to be modified for unseen tasks. For example, to perform the ST task, the postfix text becomes “Translate the English text into French". The original weight α_t is used instead of $\hat{\alpha}_t$ as the transcript length is unknown during inference. There is essentially no difference from a normal text-based LLM for different text tasks.

Limited resource fine-tuning Compared to an ASR-LLM cascade, an important advantage of Wav2Prompt is that it can be E2E combined with an LLM so that paired data can be used to fine-tune in an E2E fashion. When E2E fine-tuned on task-specific data, the prefix and postfix text are modified as in the zero-shot case. In addition, during E2E fine-tuning, in order to simplify the process, the MSE loss is not used as it requires high-quality ASR transcription and the goal here is to learn the downstream task rather than to keep the zero-shot ability. Another benefit is that the length of the speech representation **S** does not need to exactly match that of its corresponding transcript, allowing α_t to be used to match the inference condition. This overcomes any mismatch between training and inference in the original CIF (i.e., α_t was used during training while $\hat{\alpha}_t$ was used during inference). Finally, the quantity loss $\mathcal{L}_{\text{qua}}$ is still computed, because preliminary experiments have shown that without the regularising effect of $\mathcal{L}_{\text{qua}}$, the length of **S** can undergo drastic changes during optimisation, hindering convergence. Hence, the fine-tuning objective $\mathcal{L}_{\text{tune}}$ of Wav2Prompt is:

$$\mathcal{L}_{\text{tune}} = \mathcal{L}_{\text{CE}} + \mu\mathcal{L}_{\text{qua}} \tag{7}$$

where the hyper-parameter μ has the same value as in Eq. 6 for simplicity.

5 Experimental setup

After training on ASR data, Wav2Prompt was evaluated on a range of unseen tasks, including speech translation (ST), spoken language understanding (SLU), and spoken-query-based question answering (SQQA) tasks.

5.1 Datasets

ST experiments were conducted on Europarl-ST (Iranzo-Sánchez et al., 2020) English-Spanish (En-ES) and English-French (En-Fr) pairs. The corresponding English ASR data was used to train Wav2Prompt and the ASR models. In the scenarios of fine-tuning with limited resources, 10 hours of paired data was randomly selected from the training data set as limited fine-tuning data.

For the SLU and SQQA tasks, the LibriSpeech corpus (Panayotov et al., 2015) was used as the ASR data. For SLU, the Fluent Speech Commands (FSC) corpus (Lugosch et al., 2019) was used to conduct the intent classification task. In the scenarios of fine-tuning with limited resources, 2 hours of paired data were randomly selected from the training data set as limited fine-tuning data. For the SQQA task, the WikiQA (Yang et al., 2015) test set with synthesised speech queries provided by (Tang et al., 2024) was used. More details of the datasets used are listed in Appendix A.

5.2 Model specifications

Four different systems were built to compare with Wav2Prompt, and all these models used a 12-layer Conformer encoder. In all cases, speech features were extracted via a fixed WavLM Large model (Chen et al., 2021). The LLMs in this paper were always fixed following prompt tuning.

Wav2Prompt-LLM Based on the Conformer encoder, Wav2Prompt only used an extra fully connected (FC) layer that mapped the speech representation to the LLM embedding dimension (i.e. 4096) as mentioned in Eq. 3. The LLM was fixed.

ASR-LLM Cascade Based on the Conformer encoder, a connectionist temporal classification (Graves et al., 2006) (CTC)-based ASR model was built and only had an extra FC output layer. The recognised text from the ASR model was fed into the LLM, along with the prefix and postfix text, forming a cascaded system. In this paper, the Wav2Prompt aims to achieve similar results to the ASR-LLM Cascade in the zero-shot scenarios.

Oracle-LLM An oracle system was built, in which the speech ground truth transcripts were fed into the LLM. The prefix and postfix text remained the same as the ASR-LLM Cascade system.

Encoder-LLM A prior work speech-enabled LLM (Fathullah et al., 2024a) was implemented. Based on the Conformer encoder, every 8 consecutive encoder output frames were stacked to downsample the sequence length. Then, an extra FC layer was used to map the stacked encoder output to the LLM embedding dimension (i.e. 4096) before being fed into the LLM just like Wav2Prompt. In this paper, the Wav2Prompt aims to greatly surpass the Encoder-LLM in the zero-shot scenarios.

Flat-start Encoder-LLM This paper further explores directly training the Encoder-LLM on unseen tasks with limited resources in an E2E fashion without first training on ASR data, denoted as Flat-start Encoder-LLM. This is to evaluate the importance of ASR-learned alignment for other tasks.

Therefore, the trainable component of all these built systems (except for the Oracle-LLM) consisted of the same encoder along with an additional FC layer. More detail and the task-specific prompt templates can be found in Appendices B and D.

5.3 LLMs and metrics

For the ST task, BLOOMZ-7B1 (Muennighoff et al., 2023) was used. Case-sensitive detokenised BLEU (Papineni et al., 2002) results are reported to evaluate translation quality.

Model	Zero-shot		10h Data	
	En-Es	En-Fr	En-Es	En-Fr
Oracle-LLM	32.9	25.8	32.9	25.8
ASR-LLM Cascade	28.5	22.3	28.5	22.3
Encoder-LLM	15.4	6.2	29.7	26.2
Flat-start Encoder-LLM	—	—	1.3	2.5
Wav2Prompt-LLM	25.1	21.7	31.5	27.3

Table 1: %BLEU (↑) results on Europarl-ST test sets. Zero-shot means no speech-translation paired data available for fine-tuning, while 10h data refers to 10 hour paired data. Note the ASR training data includes the ASR part of the 10h ST data, so the ASR-LLM cascade results remain the same in both cases.

Model	Zero-shot SQQA
SALMONN (Tang et al., 2024)	41.0
SpeechGPT (Zhang et al., 2023a)	11.8
Qwen2-Audio (Chu et al., 2024)	59.8
Oracle-LLM	68.10
ASR-LLM Cascade	60.03
Encoder-LLM	37.96
Wav2Prompt-LLM	60.21

Table 2: Accuracy (%) (↑) of zero-shot SQQA on synthesised WikiQA.

For the SLU and SQQA tasks, Vicuna-7B-1.5 (Zhang et al., 2023b) was used, and ASR performance was also evaluated using word error rate (WER). For the SLU task, accuracy was used to measure the intent classification. For the SQQA task, following (Maaz et al., 2023), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023), which is an instruction fine-tuned LLM, was used to evaluate whether the answers predicted were correct based on the question and the right answer. Accuracy was used as the metric. The prompt template used to measure the accuracy is listed in Appendix C.

6 Experimental results

In zero-shot cases, Wav2Prompt aims to achieve results close to the ASR-LLM cascade while greatly surpassing the Encoder-LLM. With limited task-specific data and E2E fine-tuning, it aims to greatly surpass ASR-LLM and match the Encoder-LLM.

6.1 ST task results

The ST results are shown in Table 1. Since the Oracle-LLM and ASR-LLM Cascade systems cannot utilise speech-to-translation paired data for E2E fine-tuning, their results remain the same in both cases. Note this paper uses prompt tuning and the LLM was fixed. As shown in Table 1, in the zero-shot scenario, Wav2Prompt-LLM achieves results close to the ASR-LLM Cascade, which shows that the proposed Wav2Prompt maintains the advantage of the LLM zero-shot capability. Moreover, Wav2Prompt-LLM greatly outperformed the Encoder-LLM in the zero-shot scenario, which is in line with the findings of the prior work (Tang et al., 2024) that Encoder-LLM overfits to the ASR task when trained on ASR data, and exhibits limited performance in unseen ST tasks.

In the scenario with 10h ST paired data, Table 1 shows that after E2E fine-tuning, Wav2Prompt-LLM exceeded the performance of the ASR-LLM Cascade, which is the main advantage of Wav2Prompt compared to standard ASR models combined with LLMs. In addition, after fine-tuning with limited data, the Encoder-LLM also shows strong results, only slightly poorer than Wav2Prompt. This indicates that although the Encoder-LLM overfits to the ASR tasks, this issue can be ameliorated via few-shot fine-tuning. Moreover, after E2E fine-tuning, both the Wav2Prompt-LLM and Encoder-LLM surpassed the Oracle-LLM on En-Fr translation, highlighting the effectiveness of prompt tuning since the Oracle-LLM prompt is a fixed natural language phrase. The Flat-start Encoder-LLM results show that the limited data in the few-shot case was insufficient for an encoder, with no exposure to the LLM, to learn how to connect it to the LLM.

Thus, Wav2Prompt is an effective E2E alternative to traditional ASR when combined with LLMs. Appendix E gives the WER results of the ASR model. Appendix F shows extended results by reducing the data size used for fine-tuning, where Wav2Prompt still greatly outperformed the ASR-LLM Cascade even with only 30 minutes of ST paired data for E2E fine-tuning. Appendix G give the cross-domain results of fine-tuned Wav2Prompt.

6.2 SQQA task results

The zero-shot results of SQQA are shown in Table 2, where the SQQA task was evaluated in a cross-domain scenario, making it more challenging. Overall, the ASR-LLM Cascade and Wav2Prompt-

Model	Zero-shot	2h Data
Oracle-LLM	97.86	97.86
ASR-LLM Cascade	89.19	94.17
Encoder-LLM	72.20	99.34
Wav2Prompt-LLM	88.24	99.45

Table 3: Cross-domain intent classification accuracy (%) (↑) on FSC corpus for models trained on LibriSpeech corpus. Vicuna-7B-1.5 was used.

Model	test		dev	
	clean	other	clean	other
SLAM-ASR	2.4	4.9	—	—
SALMONN	2.1	4.9	—	—
Whisper-large-v3	2.7	5.2	—	—
Encoder-LLM	2.4	4.5	2.3	4.3
Wav2Prompt-LLM	2.5	4.4	2.4	4.1

Table 4: %ASR WER (↓) results on LibriSpeech test sets. Published results from SLAM-ASR (Ma et al., 2024b), SALMONN (Tang et al., 2024), and Whisper-large-v3 (Radford et al., 2023) are given.

LLM achieved similar results, with Wav2Prompt-LLM being slightly better, while the Encoder-LLM showed a noticeable performance gap due to task over-fitting. Hence, the experimental conclusion is consistent with the above ST experiments that Wav2Prompt effectively retains the LLM zero-shot ability, making it an E2E alternative to conventional ASR when combined with an LLM. Note the published results from Tang et al. (2024); Zhang et al. (2023a); Chu et al. (2024) are also shown but the comparison is not well-controlled due to the different training data.

Appendix H gives the SQQA results with different LLMs as the judges.

6.3 SLU task results

The SLU task in this paper requires the LLM to classify the intent of the speech. Furthermore, this experiment was conducted in a cross-domain scenario, i.e., the model trained on LibriSpeech ASR data was directly evaluated on FSC data. The results of intent classification are presented in Table 3. In the scenario with limited SLU data available (i.e. 2h), while the ASR-LLM cascade system cannot leverage SLU-paired data for E2E fine-tuning, the corresponding ASR data can still be used by the ASR model for domain adaptation in a cross-domain setting. Therefore, the few-shot results of the ASR-LLM Cascade showed a noticeable improvement compared to the zero-shot results. In the zero-shot scenario, Wav2Prompt-LLM achieved performance close to the ASR-LLM Cascade and greatly surpassed the Encoder-LLM, which again shows that the Encoder-LLM overfits to the ASR task that it was used in training, while Wav2Prompt retains the LLM zero-shot ability. After fine-tuning using 2h data, even compared to the domain-adapted ASR-LLM Cascade, Wav2Prompt-LLM gave improved results, showing the advantage of Wav2Prompt-LLM in E2E fine-tuning.

Model	SQQA
Oracle-LLM	68.10
ASR-LLM Cascade	60.03
Encoder-LLM	37.96
Wav2Prompt-LLM	60.21
Wav2Prompt-LLM w/o MSE Loss	36.97

Table 5: Ablation studies on the MSE loss. Zero-shot SQQA accuracy (%) (↑) results were shown.

Appendix F gives the results of fine-tuning using 10 minutes of data, with consistent conclusions.

6.4 ASR task results

While this paper focuses on downstream tasks such as ST, SLU, and SQQA, the ASR results are also given in Table 4. Wav2Prompt-LLM and Encoder-LLM both gave competitive ASR results on the LibriSpeech benchmark data. Published results from SLAM-ASR (Ma et al., 2024b), SALMONN (Tang et al., 2024), and Whisper (Radford et al., 2023) are also given, where SLAM-ASR uses WavLM Large encoder. However, the comparison with these three models is not well-controlled due to different encoders and training data.

6.5 Ablation study

In Wav2Prompt, the MSE loss is used to enforce the consistency between label-level speech representation and LLM embeddings, thereby preserving the zero-shot capability of text-based LLMs. As shown in Table 5, without the MSE loss, the zero-shot SQQA performance of Wav2Prompt-LLM dropped noticeably, so that Wav2Prompt-LLM overfits to the ASR task, similar to how the Encoder-LLM struggled with unseen tasks. Therefore, learning to match the LLM embeddings with the MSE loss is the key to unlocking the zero-shot capability of speech-enabled LLM.

Appendix I uses DTW (Sakoe and Chiba, 1978) to further evaluate speech-text alignment learning, which verifies that after trained with ASR data, Wav2Prompt enforces consistency between speech representations and LLM token embeddings.

7 Conclusions

This paper describes Wav2Prompt, which proposes a method to connect spoken input with text-based LLMs using only ASR training data while retaining the zero-shot capability for other spoken language tasks. Wav2Prompt extracts label-level speech representations using the CIF mechanism and explicitly enforces the consistency between the speech representations and LLM embeddings using the MSE loss function, thus avoiding the issue of task over-fitting. Experiments on a range of tasks, including ST, SLU, and SQQA, showed that Wav2Prompt can achieve results close to the ASR-LLM cascade system in zero-shot scenarios and greatly outperforms the existing speech-enabled LLM method. It gives results that exceed those of the ASR-LLM cascade in cases with limited task-specific data. Wav2Prompt is an E2E alternative to conventional ASR when combined with text LLMs.

Limitations

This paper is limited in the following aspects: First, this paper explores the use of off-the-shelf text-based LLMs, so the upper bound of performance is determined by the accessible text-based LLMs. However, due to the limitations of computing resources, this paper explored various 7B LLMs. Further larger LLMs are challenging given our current computing resources. Moreover, Wav2Prompt relies on open-source LLMs and cannot use closed-source LLMs, such as GPT4.

Second, this paper follows the prompt tuning approach without updating the LLM parameters. Considering many systems have been built to compare with our proposed Wav2Prompt, fine-tuning the LLM parameters would greatly increase the resources required for training, which is challenging given our limited computing resources. Future work may explore the performance of Wav2Prompt when updating the LLM parameters. However, the prompt tuning approach also has the advantage that only one LLM needs to be maintained for a series of tasks, making it highly memory-efficient for real-world deployment.

Third, for tasks like SQQA, this paper only compared performance under a zero-shot scenario because the synthesised test set was provided by the previous work of (Tang et al., 2024), and we do not have a powerful speech synthesis system available. Fourth, limited by training data and computing resources, we were not able to train our Wav2Prompt as extensively as some pre-trained speech models like Whisper (Radford et al., 2023), but we have conducted extensive experiments on many corpora, including LibriSpeech, the most widely used ASR data. This paper evaluates Wav2Prompt on English speech data, including ST between two European language pairs (En-Es and En-Fr). We believe Wav2Prompt can also be applied to other languages, including multi-lingual tasks, but its performance has not been verified and is left as future work.

Finally, this paper focuses on semantic speech tasks and has validated Wav2Prompt on four unseen tasks (including two ST tasks) and also ASR task. However, due to limited resources, there are still other potential applications, e.g. emotion recognition, which is left as future work.

Ethics Statement

Wav2Prompt allows easy integration of spoken input with text-based LLMs and provides more use-cases for off-the-shelf text-based LLMs. Wav2Prompt provides similar performance to a conventional cascade of ASR followed by a text based LLM for a zero shot performance on a range of tasks. However since it allows E2E fine-tuning, Wav2Prompt provides much improved performance in few-shot scenarios on tasks including speech translation, spoken intent classification and spoken question answering. This ability to perform very well on a range of spoken language tasks when Wav2Prompt is initially trained only on ASR training data is beneficial in many circumstances where task-specific training data is limited. This is true for many tasks, and the issue of limited spoken training data is even more severe for under-resourced languages and hence this is a significant benefit of Wav2Prompt. Wav2Prompt does not give rise to any additional potential biases beyond the ones directly inherited from the pre-trained LLM checkpoints and the speech training data used.

Acknowledgments

Keqi Deng is funded by the Cambridge Trust. This work has been performed using resources provided by the Cambridge Tier-2 system operated by the

University of Cambridge Research Computing Service (www.hpc.cam.ac.uk) funded by EPSRC Tier-2 capital grant EP/T022159/1.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2024. GPT-4 technical report. *Preprint*, arXiv:2303.08774.

Zalán Borsos, Raphaël Marinier, Damien Vincent, Eugene Kharitonov, Olivier Pietquin, Matt Sharifi, Dominik Roblek, Olivier Teboul, David Grangier, Marco Tagliasacchi, et al. 2023. AudioLM: A language modeling approach to audio generation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 31:2523–2533.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. In *Proc. NeurIPS*.

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

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, Jian Wu, Long Zhou, Shuo Ren, Yanmin Qian, Yao Qian, Micheal Zeng, and Furu Wei. 2021. WavLM: Large-scale self-supervised pre-training for full stack speech processing. *IEEE J. Sel. Top. Sig. Process.*, 16:1505–1518.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. 2023. PaLM: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):1–113.

Yunfei Chu, Jin Xu, Qian Yang, Haojie Wei, Xipin Wei, Zhifang Guo, Yichong Leng, Yuanjun Lv, Jinzheng He, Junyang Lin, Chang Zhou, and Jingren Zhou. 2024. Qwen2-audio technical report. *Preprint*, arXiv:2407.10759.

Nilaksh Das, Saket Dingliwal, Srikanth Ronanki, Rohit Paturi, Zhaocheng Huang, Prashant Mathur, Jie Yuan, Dhanush Bekal, Xing Niu, Sai Muralidhar Jayanthi, Xilai Li, Karel Mundnich, Monica Sunkara, Sundararajan Srinivasan, Kyu J Han, and Katrin Kirchhoff. 2024. Speechverse: A large-scale generalizable audio language model. *Preprint*, arXiv:2405.08295.

Keqi Deng and Phil Woodland. 2024a. Label-synchronous neural transducer for E2E simultaneous speech translation. In *Proc. ACL*.

Keqi Deng and Philip C. Woodland. 2024b. Label-synchronous neural transducer for adaptable online e2e speech recognition. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 32:3507–3516.

Mattia A Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a multilingual speech translation corpus. In *Proc. NAACL-HLT*.

Linhao Dong and Bo Xu. 2020. CIF: Continuous integrate-and-fire for end-to-end speech recognition. In *Proc. ICASSP*.

Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Junteng Jia, Yuan Shangguan, Ke Li, Jinxi Guo, Wenhan Xiong, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. 2024a. Prompting large language models with speech recognition abilities. In *Proc. ICASSP*.

Yassir Fathullah, Chunyang Wu, Egor Lakomkin, Ke Li, Junteng Jia, Yuan Shangguan, Jay Mahadeokar, Ozlem Kalinli, Christian Fuegen, and Mike Seltzer. 2024b. AudioChatLlama: Towards general-purpose speech abilities for LLMs. *Preprint*, arXiv:2311.06753.

Philip Gage. 1994. A new algorithm for data compression. *The C Users Journal*, 12(2):23–38.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2021. Making pre-trained language models better few-shot learners. In *Proc. ACL/IJCNLP*.

Yuan Gong, Hongyin Luo, Alexander H. Liu, Leonid Karlinsky, and James R. Glass. 2024. Listen, think, and understand. In *Proc. ICLR*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proc. ICML*.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang. 2020. Conformer: Convolution-augmented Transformer for speech recognition. In *Proc. Interspeech*.

Mutian He and Philip N. Garner. 2023. Can ChatGPT detect intent? evaluating large language models for spoken language understanding. In *Proc. Interspeech*.

Chien-yu Huang, Ke-Han Lu, Shih-Heng Wang, Chi-Yuan Hsiao, Chun-Yi Kuan, Haibin Wu, Siddhant Arora, Kai-Wei Chang, Jiatong Shi, Yifan Peng, et al. 2024. Dynamic-Superb: Towards a dynamic, collaborative, and comprehensive instruction-tuning benchmark for speech. In *Proc. ICASSP*.

J. Iranzo-Sánchez, J. A. Silvestre-Cerdà, J. Jorge, N. Roselló, A. Giménez, A. Sanchis, J. Civera, and A. Juan. 2020. Europarl-ST: A multilingual corpus for speech translation of parliamentary debates. In *Proc. ICASSP*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Teven Le Scao, Angela Fan, Christopher Akiki, Ellie Pavlick, Suzana Ilić, Daniel Hesslow, Roman Castagné, Alexandra Sasha Luccioni, François Yvon, Matthias Gallé, et al. 2023. Bloom: A 176b-parameter open-access multilingual language model. *Preprint*, arXiv:2211.05100.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proc. EMNLP*.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. 2023. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proc. ICML*.

Xiao Liu, Kaixuan Ji, Yicheng Fu, Weng Tam, Zhengxiao Du, Zhilin Yang, and Jie Tang. 2022. P-tuning: Prompt tuning can be comparable to fine-tuning across scales and tasks. In *Proc. ACL*.

Xiao Liu, Yanan Zheng, Zhengxiao Du, Ming Ding, Yujie Qian, Zhilin Yang, and Jie Tang. 2024. GPT understands, too. *AI Open*, 5:208–215.

Loren Lugosch, Mirco Ravanelli, Patrick Ignoto, Vikrant Singh Tomar, and Yoshua Bengio. 2019. Speech Model Pre-Training for End-to-End Spoken Language Understanding. In *Proc. Interspeech*.

Rao Ma, Adian Liusie, Mark J. F. Gales, and Kate M. Knill. 2024a. Investigating the emergent audio classification ability of ASR foundation models. *Preprint*, arXiv:2311.09363.

Ziyang Ma, Guanrou Yang, Yifan Yang, Zhifu Gao, Jiaming Wang, Zhihao Du, Fan Yu, Qian Chen, Siqi Zheng, Shiliang Zhang, and Xie Chen. 2024b. An embarrassingly simple approach for LLM with strong ASR capacity. *Preprint*, arXiv:2402.08846.

Muhammad Maaz, Hanoona Rasheed, Salman Khan, and Fahad Shahbaz Khan. 2023. Video-ChatGPT: Towards detailed video understanding via large vision and language models. *Preprint*, arXiv:2306.05424.

Niklas Muennighoff, Thomas Wang, Lintang Sutawika, Adam Roberts, Stella Biderman, Teven Le Scao, M Saiful Bari, Sheng Shen, Zheng-Xin Yong, Hailey Schoelkopf, Xiangru Tang, Dragomir Radev, Alham Fikri Aji, Khalid Almubarak, Samuel Albanie, Zaid Alyafeai, Albert Webson, Edward Raff, and Colin Raffel. 2023. Crosslingual generalization through multitask finetuning. *Preprint*, arXiv:2211.01786.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. 2022. Training language models to follow instructions with human feedback. *Proc. NeurIPS*.

V. Panayotov, G. Chen, D. Povey, and S. Khudanpur. 2015. LibriSpeech: an ASR corpus based on public domain audio books. In *Proc. ICASSP*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *Proc. ACL*.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *Proc. ICML*.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine Mcleavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proc. ICML*.

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, Hannah Muckenhirn, Dirk Padfield, James Qin, Danny Rozenberg, Tara Sainath, Johan Schalkwyk, Matt Sharifi, Michelle Tadmor Ramanovich, Marco Tagliasacchi, Alexandru Tudor, Mihajlo Velimirović, Damien Vincent, Jiahui Yu, Yongqiang Wang, Vicky Zayats, Neil Zeghidour, Yu Zhang, Zhishuai Zhang, Lukas Zilka, and Christian Frank. 2023. AudioPaLM: A large language model that can speak and listen. *Preprint*, arXiv:2306.12925.

H. Sakoe and S. Chiba. 1978. Dynamic programming algorithm optimization for spoken word recognition. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 26(1):43–49.

Timo Schick and Hinrich Schütze. 2020. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proc. EACL*.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting knowledge from language models with automatically generated prompts. In *Proc. EMNLP*.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun MA, and Chao Zhang. 2024. SALMONN: Towards generic hearing abilities for large language models. In *Proc. ICLR*.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. LLaMA: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023b. LLaMa 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Sanh Victor, Webson Albert, Raffel Colin, Bach Stephen, Sutawika Lintang, Alyafeai Zaid, Chaffin Antoine, Stiegler Arnaud, Raja Arun, Dey Manan, et al. 2022. Multitask prompted training enables zero-shot task generalization. In *Proc. ICLR*.

Chengyi Wang, Sanyuan Chen, Yu Wu, Ziqiang Zhang, Long Zhou, Shujie Liu, Zhuo Chen, Yanqing Liu, Huaming Wang, Jinyu Li, Lei He, Sheng Zhao, and Furu Wei. 2023. Neural codec language models are zero-shot text to speech synthesizers. *Preprint*, arXiv:2301.02111.

Shinji Watanabe, Takaaki Hori, Shigeki Karita, Tomoki Hayashi, Jiro Nishitoba, Yuya Unno, Nelson Enrique Yalta Soplin, Jahn Heymann, Matthew Wiesner, Nanxin Chen, Adithya Renduchintala, and Tsubasa Ochiai. 2018. ESPnet: End-to-end speech processing toolkit. In *Proc. Interspeech*.

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Trans. Mach. Learn. Res.*, 2022.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2020. Transformers: State-of-the-art natural language processing. In *Proc. EMNLP (Demos)*, pages 38–45.

Jian Wu, Yashesh Gaur, Zhuo Chen, Long Zhou, Yimeng Zhu, Tianrui Wang, Jinyu Li, Shujie Liu, Bo Ren, Linquan Liu, and Yu Wu. 2023. On decoder-only architecture for speech-to-text and large language model integration. In *Proc. ASRU*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2024. A paradigm shift in machine translation: Boosting translation performance of large language models. In *Proc. ICLR*.

Shu-Wen Yang, Po-Han Chi, Yung-Sung Chuang, Cheng-I Jeff Lai, Kushal Lakhotia, Yist Y. Lin, Andy T. Liu, Jiatong Shi, Xuankai Chang, Guan-Ting Lin, Tzu-Hsien Huang, Wei-Cheng Tseng, Kotik Lee, Da-Rong Liu, Zili Huang, Shuyan Dong, Shang-Wen Li, Shinji Watanabe, Abdelrahman Mohamed, and Hung-yi Lee. 2021. SUPERB: Speech processing universal performance benchmark. In *Proc. Interspeech*.

Yi Yang, Wen-tau Yih, and Christopher Meek. 2015. WikiQA: A challenge dataset for open-domain question answering. In *Proc. EMNLP*.

Wenyi Yu, Changli Tang, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2024. Connecting speech encoder and large language model for ASR. In *Proc. ICASSP*.

Dong Zhang, Shimin Li, Xin Zhang, Jun Zhan, Pengyu Wang, Yaqian Zhou, and Xipeng Qiu. 2023a. SpeechGPT: Empowering large language models with intrinsic cross-modal conversational abilities. In *Proc. EMNLP (Findings)*.

Xiaoying Zhang, Baolin Peng, Kun Li, Jingyan Zhou, and Helen Meng. 2023b. SGP-TOD: Building task bots effortlessly via schema-guided LLM prompting. In *Proc. EMNLP (Findings)*.

Table 6: Statistics of datasets used in this paper

	Europarl-ST	
ASR Data Train Set	train	
-Hours	81 hours	
-Samples	34K	
Few-shot ST Data Train sets	train-en-es-10h	train-en-fr-10h
-Hours	10 hours	10 hours
-Samples	4.2K	4.2K
ST Data Test sets	test-en-es	test-en-fr
-Hours	2.9 hours	2.8 hours
-Samples	1.3K	1.2K
	LibriSpeech	
ASR Data Train set	train-960	
-Hours	960 hours	
-Samples	281K	
Test sets	test-clean / other	dev-clean / other
-Hours	5.4 / 5.3 hours	5.4 / 5.1 hours
-Samples	2.6 / 2.9K	2.7 / 2.9K
	Synthesised WikiQA	
SQQA Data test set	test	
-Hours	0.5 hours	
-Samples	0.6K	
	Fluent Speech Commands (FSC)	
Few-shot SLU Data train set	train-2h	
-Hours	2 hours	
-Samples	3.2K	
SLU Data test set	test	
-Hours	2.4 hours	
-Samples	3.8K	

A Data Set Statistics

The training and test data set statistics for the corpora used in the experiments are shown in Table 6. The Europarl-ST data was collected from the European Parliament debate (Iranzo-Sánchez et al., 2020). LibriSpeech is an audiobook reading corpus (Panayotov et al., 2015). The WikiQA (Yang et al., 2015) test set with synthesised speech queries provided by (Tang et al., 2024) was used for the SQQA task, in which the answers generated from GPT4 were used as the reference answers. The FSC data (Lugosch et al., 2019) was collected from English commands commonly used for a smart home or virtual assistant, which has 31 distinct intents.

B Training and hyper-parameter details

For the Conformer encoder, the kernel size of the convolution module was set to 31. The attention dimension, feed-forward dimension, and attention heads of the Conformer encoder were set to 256, 2048, and 4. The data was pre-processed following ESPnet (Watanabe et al., 2018) recipes. Following the ESPnet recipe, The S3PRL toolkit (Yang et al., 2021) was used to extract speech features from a fixed WavLM Large model (Chen et al., 2021). Convolutional layers were used to down-sample in time by a factor of 2. The CTC models used 1000 BPE (Gage, 1994) modelling units. For models trained on the ASR data of Europarl-ST, the CTC ASR model was trained for 20 epochs using a learning rate $3 \cdot 10^{-3}$ with 25k warmup steps, and the Wav2Prompt-LLM, Encoder-LLM, and Flat-start Encoder-LLM converged after 20 epochs of training. In scenarios with limited task-specific data, the models were fine-tuned for 10 epochs. For models trained on LibriSpeech data, the models were trained for 10 epochs. Speed perturbation was used

Table 7: Prompt used in this paper to evaluate SQQA task.

Prompt	Please evaluate the following question-answer pair: Question: [question] Correct Answer: [answer] Predicted Answer: [prediction] Provide your evaluation only as a yes/no and score where the score is an integer value between 0 and 5, with 5 indicating the highest meaningful match. Please generate the response in the form of a Python dictionary string with keys ‘pred’ and ‘score’, where value of ‘pred’ is a string of ‘yes’ or ‘no’ and value of ‘score’ is in INTEGER, not STRING. DO NOT PROVIDE ANY OTHER OUTPUT TEXT OR EXPLANATION. Only provide the Python dictionary string. For example, your response should look like this:{‘pred’: ‘yes’, ‘score’: 4}.

with factors 0.9 and 1.1. In scenarios of fine-tuning using the FSC data, the models were fine-tuned for 20 epochs. The number of trainable parameters of the Wav2Prompt-LLM, ASR-LLM Cascade, Encoder-LLM, Flat-start Encoder-LLM systems were 37.45 M, 36.65 M, 44.79 M, and 44.79 M, respectively.

For the Europarl-ST, 250 M batch bins (as implemented by ESPnet) were used, and each epoch took about 3 hours using NVIDIA A100. For the LibriSpeech, 140 M batch bins (as implemented by ESPnet) were used, and each epoch took about 23 hours using NVIDIA A6000 Ada. For the proposed Wav2Prompt, γ and μ in Eq. 6 were set to 20 and 0.05, respectively. During decoding, the beam size was set to 5. For ASR decoding, the repetition penalty as implemented by Huggingface (Wolf et al., 2020) was set to 1.5.

C SQQA evaluation

Mistral-7B-Instruct-v0.2 was used to evaluate the accuracy of the model prediction in the SQQA task. The prompt used in this paper is listed in Table 7, which follows (Maaz et al., 2023). The accuracy was computed by counting the frequency with which the Mistral LLM outputs ‘yes’.

D Task-specific prompt templates

The prefix and postfix text used in this paper as task-specific prompt templates are listed in Table 10, which were designed based on the intended use of different LLM (Zhang et al., 2023b; Touvron et al., 2023b; Le Scao et al., 2023) or related work (Xu et al., 2024; He and Garner, 2023).

Model	test		dev		FSC
	clean	other	clean	other	Test
ASR Model	2.5	4.4	2.4	4.1	12.9
+2h data fine-tune	—	—	—	—	3.1

Table 8: %ASR WER (↓) results for the model trained on LibriSpeech data. The FSC test set is cross-domain for LibriSpeech. The 2h data refers to the ASR part of the 2h SLU data in limited resource scenarios.

Model	Europarl-ST ASR part	
	test	dev
ASR Model	16.4	16.4

Table 9: %ASR WER (↓) results for the model trained on the corresponding English ASR data of the Europarl-ST corpus. Note that punctuation is also considered when calculating WER.

E Extra ASR results

The WER results of the ASR model used in the ASR-LLM cascade are shown in Table 8 and Table 9. The FSC test set is a cross-domain set for LibriSpeech, therefore the performance on it was greatly improved after fine-tuning using 2h data. For the WER results on Europarl-ST ASR test sets, punctuation is included, consistent with how it is considered in the ST BLEU results in this paper.

F Extended results with different sizes of fine-tuning data

This section further adjusts the data size when fine-tuning with limited resources. As shown in Table 11, even 30 minutes of ST data benefit E2E approaches like Wav2Prompt-LLM and Encoder-

Table 10: Prefix and postfix text used in this paper

ASR Train: Prefix	""
ASR Train: Postfix	Repeat the above English text:
ST Test: Prefix	Translate the following English sentence into [target language]:
ST Test: Postfix	[target language]:
SLU Test: Prefix	We will show you some commands given by a user to a voice assistant like Siri or Olly. Please classify the intent of the command. There are 31 unique intents in total, which are divided into three slots: "action", "object", and "location". A slot takes on one of multiple values: the "action" slot can take on the values: "change language", [...]; the "object" slot can take on the values: "none", "music", [...]; the "location" slot can take on the values: "none", "kitchen", [...]. The format of intent is: "action_object_location". The list of all the intents are: "increase_volume_none", [...]. You can first repeat the command and then think about the intent. Please give answers like: {"Command": <your_repeated_command>, "Intent": <your_intent_prediction>}. For example: [...]. The intent in your answer must match one of the intents given above. If you are uncertain, choose the one that you think is the most likely. Here are the commands: USER:
SLU Test: Postfix	Repeat the above English text and classify the intent: ASSISTANT:
SQQA Test: Prefix	Give a precise and clear answer to the question. Don't be verbose. You can first repeat the question and then think about the answer. Please give answers like: {"Question": <your_repeated_question>, "Answer": <your _answer>}. If you are not sure, leave the answer blank, like {"Question": <your_repeated_question>, "Answer": ""}. Here are the questions: USER:
SQQA Test: Postfix	Repeat the above English text and answer the question: ASSISTANT:
ASR Test: Prefix	""
ASR Test: Postfix	Resume the above English text:

LLM in improving performance. Moreover, with 30 minutes of ST data, Wav2Prompt still greatly outperformed the ASR-LLM Cascade.

As shown in Table 12, even 10 minutes of SLU data fine-tuning allows the E2E approaches to greatly surpass the cascaded system, which uses 2 hours of ASR data to fine-tune the ASR model, highlighting the advantage of E2E methods in fine-tuning. Furthermore, with just 10 minutes of SLU data, Wav2Prompt-LLM can achieve almost the same results as Oracle-LLM.

Model	En-Fr (BLEU)
Oracle-LLM	25.8
ASR-LLM Cascade	22.3
Encoder-LLM (zero-shot)	6.2
+ 30min data fine-tune	21.3
+ 1h data fine-tune	22.1
+ 10h data fine-tune	26.2
Wav2Prompt-LLM (zero-shot)	21.7
+ 30min data fine-tune	24.6
+ 1h data fine-tune	25.0
+ 10h data fine-tune	27.3

Table 11: %BLEU (↑) results on the test sets of Europarl-ST En-Fr pair.

Model	SLU Accuracy
Oracle-LLM	97.86%
ASR-LLM Cascade	89.19%
+ 2h ASR data fine-tune	94.17%
Encoder-LLM (zero-shot)	72.20%
+ 10min data E2E fine-tune	97.23%
+ 2h data E2E fine-tune	99.34%
Wav2Prompt-LLM (zero-shot)	88.24%
+ 10min data E2E fine-tune	97.84%
+ 2h data E2E fine-tune	99.44%

Table 12: Intent classification accuracy (%) (↑) on FSC corpus for models trained on LibriSpeech corpus.

G Cross-domain results of fine-tuned Wav2Prompt on ST task

This section additionally tests the performance of the fine-tuned model in a cross-domain setting. For the Wav2Prompt model fine-tuned with 10 hours of supervised data from Europarl-ST En-Fr on speech translation, we directly tested it on the Must-C (Di Gangi et al., 2019) test set, and the results are shown in Table 13. The results show that the performance drop on the MuST-C cross-domain test set is acceptable, indicating that Wav2Prompt is generally robust.

En-Fr ST Model	Europarl-ST	MuST-C
Wav2Prompt-LLM	27.3	22.3

Table 13: Cross-domain ST results on MuST-C of Wav2Prompt fine-tuned from Europarl-ST.

H SQQA results with different LLMs as Judges

This section explores the use of different LLMs as the judge to evaluate the SQQA performance. The results in Table 14 show that while there are some differences in accuracy between the GPT API and the open-source models as evaluation models, the experimental conclusions remain consistent across the different models acting as judges.

I Extended results on DTW scores

Dynamic Time Warping (DTW) (Sakoe and Chiba, 1978) were used to evaluate the speech-text alignment learnt by Wav2Prompt. The DTW scores were computed to measure the similarity between LLM token embeddings $\mathbf{Z} = (\boldsymbol{z}_0, \boldsymbol{z}_1, \cdots, \boldsymbol{z}_N)$ and speech representations $\mathbf{S} = (\boldsymbol{s}_1, \cdots, \boldsymbol{s}_M)$ on LibriSpeech test-clean set for Encoder-LLM and Wav2Prompt. DTW score is computed by finding the optimal alignment that minimises the cumulative distance between them. The DTW score is normalised by dividing by the length of **S**, i.e., M. When computing the distance between any two vectors (e.g., $\boldsymbol{z}_i$ and $\boldsymbol{s}_j$) in these two sequences, the squared Euclidean distance is used: $d(\boldsymbol{z}_i, \boldsymbol{s}_j) = \frac{1}{K}\sum_{k=1}^{K}(z_{i,k} - s_{j,k})^2$ where K is the feature dimension.

As shown in the DTW scores from Table 15, the Wav2Prompt representations are well-aligned with corresponding LLM token embeddings, whereas the representations of the Encoder-LLM are inconsistent, leading to task over-fitting.

J Assets and licenses

The following licenses apply to the models used in this paper:

- LLaMA2: `https://huggingface.co/meta-llama/Llama-2-7b-hf/blob/main/LICENSE.txt` applies to Vicuna-7B-1.5.
- Apache-2.0: `https://www.apache.org/licenses/LICENSE-2.0` applies to Mistral-7B-Instruct-v0.2.
- BigScience RAIL License v1.0: `https://huggingface.co/spaces/bigscience/license` applies to BLOOMZ-7B1.

The following licenses apply to the datasets used in this paper:

- CC BY-NC 4.0: `https://spdx.org/licenses/CC-BY-NC-4.0` applies to Europarl-ST data.

Models	**LLM Evaluation Models**	**SQQA Accuracy (%)**
Oracle-LLM	Mistral-7B-Instruct-v0.2	68.10
ASR-LLM Cascade	Mistral-7B-Instruct-v0.2	60.03
Encoder-LLM	Mistral-7B-Instruct-v0.2	37.96
Wav2Prompt-LLM	Mistral-7B-Instruct-v0.2	60.21
Oracle-LLM	Llama-3.1-8B-Instruct	69.40
ASR-LLM Cascade	Llama-3.1-8B-Instruct	60.96
Encoder-LLM	Llama-3.1-8B-Instruct	38.10
Wav2Prompt-LLM	Llama-3.1-8B-Instruct	61.80
Oracle-LLM	GPT API	57.59
ASR-LLM Cascade	GPT API	48.26
Encoder-LLM	GPT API	21.15
Wav2Prompt-LLM	GPT API	49.37

Table 14: SQQA Accuracy Comparison Across Different Models

Model	DTW scores
Encoder-LLM	17.5970
Proposed Wav2Prompt-LLM	0.0049

Table 15: DTW (↓) results on LibriSpeech test set.

- CC BY 4.0: `https://spdx.org/licenses/CC-BY-4.0` applies to LibriSpeech data.
- CC-BY-NC-ND-4.0: `https://spdx.org/licenses/CC-BY-NC-ND-4.0` applies to Fluent Speech Commands data.

The following license applies to the code and Python package used in this paper:

- Apache-2.0: applies to Huggingface Transformers (`https://github.com/huggingface/transformers/blob/main/LICENSE`) and ESPnet (`https://github.com/espnet/espnet/blob/master/LICENSE`).