



End-to-End Speech Translation Guided by Robust Translation Capability of Large Language Model

Yosuke Higuchi¹, Tetsuji Ogawa¹, Tetsunori Kobayashi¹

¹Department of Communications and Computer Engineering, Waseda University, Japan

Abstract

We present an end-to-end speech translation (ST) model that uses a large language model (LLM) to guide the translation process. Recent advances in LLMs have shown strong contextual understanding and robustness to noisy text, making them beneficial for mitigating automatic speech recognition (ASR) errors. Building on these strengths, we develop an LLM-driven ST model within an encoder-decoder framework, with the encoder handling an auxiliary ASR task and the decoder incorporating an LLM at its front end. Here, the encoder generates an ASR hypothesis that cues the LLM to perform machine translation. The LLM output is then fed into the decoder to yield the final translation. This two-pass design capitalizes on the LLM’s robust and accurate translation capabilities, while enabling end-to-end optimization tailored to specific ST tasks. Experimental results on various ST tasks reveal significant performance gains with our LLM integration, and extensive analyses further validate our approach.

Index Terms: end-to-end speech translation, large language model, language model integration

1. Introduction

Large language models (LLMs) [1–6] have rapidly become a dominant paradigm in natural language processing, driven by the exponential growth of internet-sourced data and significant advancements in GPU-accelerated computing. Among their key strengths is a robust ability to interpret diverse textual inputs [7–11], enabling them to generate contextually appropriate responses—even when queries are imperfect or noisy (as is often the case with human-generated content).

Such advantages are particularly beneficial in developing accurate speech-to-text systems, where automatic speech recognition (ASR) outputs can serve as inputs into LLMs for text generation. A number of studies have demonstrated promising results for enhancing ASR models by refining their hypotheses, focusing on the LLMs’ potential for generative grammatical/spelling error correction [12–18]. Recent research has further extended this approach to encompass a broader range of speech tasks beyond improving ASR [19]. Moreover, spoken language understanding can be effectively realized by applying natural language understanding tasks to ASR results, exhibiting resilience even in the presence of ASR errors [20–23].

In this work, we aim to achieve speech translation (ST) by harnessing the robust machine translation (MT) capabilities of LLMs. A common approach for adopting LLMs for ST involves fine-tuning them on speech data to generate translations [24–27], often using parameter-efficient techniques such as conditional soft-prompting [28, 29] and low-rank adaptation (LoRA) [30]. However, these methods can be computationally

demanding during both training and inference, especially when processing lengthy speech sequences alongside text. Instead, we explore a simpler solution that relies on off-the-shelf LLMs without fine-tuning, capitalizing on their inherent MT abilities that are generally highly accurate [31–34] and robust against error-prone inputs, such as those produced by ASR. Our approach is conceptually similar to fusion strategies (e.g., cold fusion [35, 36]), which build an end-to-end ASR system on top of prior language-specific information captured by an external language model (LM). In our setting, the ST system focuses on bridging speech and its translation, while the pre-trained LLM provides strong guidance on potential translation outputs.

To this end, we propose a novel ST model based on the attention-based encoder-decoder (AED) architecture [37–39], augmented with an auxiliary ASR sub-task trained via connectionist temporal classification (CTC) [40] on the encoder output. We further introduce an LLM-guided decoder [18], wherein a fixed LLM is instructed to translate the intermediate ASR hypothesis (from CTC) into the target language. The LLM output is then fed into the decoder to generate the final translation. This design allows for a formulation unifying cascaded and end-to-end ST features, exploiting the LLM’s strengths in generating accurate translations and handling noisy ASR results, while preserving a direct objective dedicated to specific ST tasks.

2. Background: End-to-End ST

End-to-end ST aims to model a direct mapping from a source-language speech sequence $O = (\mathbf{o}_t \in \mathbb{R}^F | t = 1, \dots, T)$ of length T into its corresponding target-language text sequence $W = (w_n \in \mathcal{V} | n = 1, \dots, N)$ of length N . Here, F denotes the dimensionality of the acoustic features at each frame, and \mathcal{V} represents the vocabulary of the target language.

End-to-end ST systems are typically built using the AED architecture [37–39], where the posterior probability distribution of $p(W|O)$ is modeled using a probabilistic chain rule as

$$p^{\text{aed}}(W|O) = \prod_{n=1}^N p(w_n | W_{<n}, O). \quad (1)$$

The token emission probability of $p(w_n | W_{<n}, O)$ is computed using encoder and decoder networks as

$$H = (\mathbf{h}_1, \dots, \mathbf{h}_{T'}) = \text{Encoder}(O) \in \mathbb{R}^{T' \times D}, \quad (2)$$

$$p(w_n | W_{<n}, O) = \text{Decoder}(W_{<n}, H) \in [0, 1]^{|V|}. \quad (3)$$

In Eq. (2), $\text{Encoder}(\cdot)$ converts the input sequence into D -dimensional hidden vectors H of down-sampled length $T' (< T)$. In Eq. (3), $\text{Decoder}(\cdot)$ comprises causal networks, followed by a linear layer and the softmax function to generate the output vocabulary distribution. The decoder also incorporates the

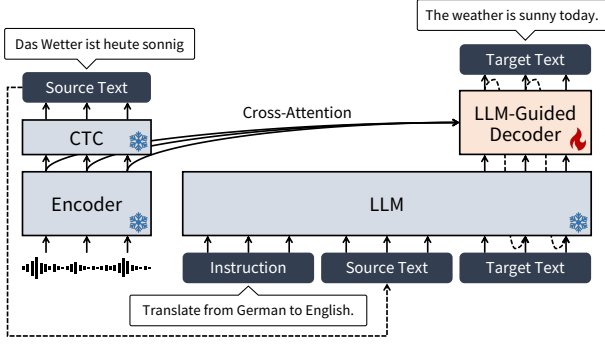


Figure 1: Overview of proposed LLM-guided end-to-end ST.

cross-attention mechanism that attends to the relevant parts of the encoder output H for producing the current token w_n .

The AED model is trained by minimizing the negative log-likelihood of Eq. (1), i.e., $\mathcal{L}^{\text{aed-st}} = -\log p^{\text{aed}}(W|O)$. To facilitate end-to-end ST training, it is common to employ multi-task learning with an auxiliary ASR sub-task [41–43]. Given the source-language transcript $W^{\text{src}} = (w_m^{\text{src}} \in \mathcal{V}^{\text{src}} | m = 1, \dots, M)$ of length M and vocabulary \mathcal{V}^{src} , the objective of end-to-end ST \mathcal{L}^{st} is defined using an ASR loss \mathcal{L}^{asr} as

$$\mathcal{L}^{\text{st}} = (1 - \lambda^{\text{asr}}) \mathcal{L}^{\text{aed-st}}(W|O) + \lambda^{\text{asr}} \mathcal{L}^{\text{asr}}(W^{\text{src}}|O), \quad (4)$$

where λ^{asr} ($0 \leq \lambda^{\text{asr}} < 1$) is a tunable weight. The ASR loss \mathcal{L}^{asr} is defined based on the joint CTC/AED framework [44] as

$$\mathcal{L}^{\text{asr}} = (1 - \lambda^{\text{ctc}}) \mathcal{L}^{\text{aed-asr}}(W^{\text{src}}|O) + \lambda^{\text{ctc}} \mathcal{L}^{\text{ctc}}(W^{\text{src}}|O), \quad (5)$$

where λ^{ctc} ($0 \leq \lambda^{\text{ctc}} < 1$) is a tunable weight for the CTC loss \mathcal{L}^{ctc} [40], which is computed using the encoder output H from Eq. (2). The AED loss $\mathcal{L}^{\text{aed-asr}}$ is computed using an additional ASR decoder, similarly to Eq. (3).

3. LLM-Guided End-to-End ST

Figure 1 shows our proposed end-to-end ST model, which benefits from the robust and accurate translation guidance provided by an off-the-shelf LLM. The proposed model builds on the conventional end-to-end ST approach (as described in Sec. 2) by simply replacing the original decoder with an LLM-guided decoder [18]. In this setup, the LLM is tasked with translating an ASR hypothesis—obtained via the CTC-based ASR sub-task—into the target language, and its output representations are then fed into the decoder to produce the final translation result. This integration allows the model to exploit the LLM’s strengths not only in generating precise translations but also in robustly handling noisy inputs (i.e., those with ASR errors). Furthermore, error propagation from the LLM’s translations can be effectively mitigated through end-to-end training of the decoder, which explicitly attends to the encoder output.

Below, we delve into the structure of our proposed model, which enables an end-to-end ST formulation powered by the LLM’s MT capabilities. Then, we describe the specific training and inference algorithms for the proposed model. Finally, we discuss the relationship to existing work on two-pass modeling.

3.1. Formulation

The proposed model addresses end-to-end ST by factorizing the posterior distribution $p(W|O)$ over ASR hypotheses as

$$p(W|O) = \sum_{\tilde{W}^{\text{src}} \in \mathcal{H}} p(W|\tilde{W}^{\text{src}}, O) p(\tilde{W}^{\text{src}}|O), \quad (6)$$

where \tilde{W}^{src} denotes a hypothesized source-language transcript, and \mathcal{H} is a set of all ASR hypotheses for the speech input O . In this formulation, the conditional distribution $p(W|\tilde{W}^{\text{src}}, O)$ is further decomposed using a probabilistic chain rule as

$$p(W|\tilde{W}^{\text{src}}, O) = \prod_{n=1}^N p(w_n|W_{<n}, \tilde{W}^{\text{src}}, O). \quad (7)$$

Compared to the standard AED formulation in Eq. (1), the prediction of each target token is conditioned not only on its preceding tokens $W_{<n}$ but also on the ASR hypothesis \tilde{W}^{src} . This integrates both end-to-end and cascaded ST features, enabling the model to exploit complementary information from the intermediate ASR output while directly modeling the translation.

The token emission probability in Eq. (7) is computed similarly to the AED architecture, but the original decoder (i.e., Eq. (3)) is replaced with an LLM-guided decoder as

$$p(w_n|W_{<n}, \tilde{W}^{\text{src}}, O) = \text{LLMGuidedDecoder}(\mathbf{e}_1, \dots, \mathbf{e}_n, H) \in [0, 1]^{|V|}, \quad (8)$$

where H is the encoder output from Eq. (2). The sequence $(\mathbf{e}_1, \dots, \mathbf{e}_n)$ comprises D^{llm} -dimensional hidden vectors derived from an LLM as

$$\mathbf{e}_n = \text{LLM}(W^{\text{ins}}, \tilde{W}^{\text{src}}, W_{<n}) \in \mathbb{R}^{D^{\text{llm}}}, \quad (9)$$

where W^{ins} is an instruction that prompts the LLM to translate \tilde{W}^{src} into the target language. With the implementation shown in Eqs. (8) and (9), the LLM is employed to infuse cascaded ST capabilities into the model (cf. Eq. (7)). Although the LLM is not explicitly trained to translate from erroneous ASR hypotheses, its strong contextual reasoning allows for interpreting noisy inputs. Consequently, the LLM-guided decoder is expected to facilitate end-to-end ST, incorporating the robust MT capabilities of the LLM into direct translation modeling.

3.2. Inference Algorithm

Figure 1 illustrates the decoding flow for the proposed model. The model estimates the most probable translation \hat{W} by solving the following optimization problem based on Eq. (6):

$$\hat{W} = \underset{W}{\text{argmax}} \sum_{\tilde{W}^{\text{src}} \in \mathcal{H}} p(W|\tilde{W}^{\text{src}}, O) p(\tilde{W}^{\text{src}}|O), \quad (10)$$

To simplify the search process, the Viterbi approximation is applied to $p(\tilde{W}^{\text{src}}|O)$, which yields

$$\hat{W} \approx \underset{W}{\text{argmax}} p(W|\tilde{W}^{\text{src}}, O), \quad (11)$$

$$\text{where } \tilde{W}^{\text{src}} = \underset{\tilde{W}^{\text{src}}}{\text{argmax}} p(\tilde{W}^{\text{src}}|O). \quad (12)$$

In Eq. (12), the ASR hypothesis \tilde{W}^{src} is first obtained by performing best path decoding [40] using the CTC module applied to the encoder output. Then, in Eq. (11), the final translation \hat{W} is generated by performing beam search decoding with the LLM-guided decoder, which takes the instruction W^{ins} and the ASR hypothesis \tilde{W}^{src} as inputs.

3.3. Training Algorithm

The proposed end-to-end ST model is trained in two stages:

Stage 1: Train the standard AED-based model using the loss \mathcal{L}^{st} defined in Eq. (4).

Stage 2: Freeze the encoder network with the CTC-based ASR module, then train the LLM-guided decoder while keeping the parameters of the pre-trained LLM fixed.

By freezing all pre-trained components in Stage 2, the LLM-guided decoder can focus solely on end-to-end ST modeling, learning to integrate speech information from the encoder with accurate and robust translation guidance provided by the LLM.

The LLM-guided decoder is optimized by minimizing the negative log-likelihood of Eq. (6) expanded with Eq. (7),

$$\begin{aligned}
 & -\log \sum_{\tilde{W}^{\text{src}} \in \mathcal{H}} \prod_{n=1}^N p(w_n | W_{<n}, \tilde{W}^{\text{src}}, O) p(\tilde{W}^{\text{src}} | O) \quad (13) \\
 & \leq \underbrace{-\mathbb{E}_{\tilde{W}^{\text{src}} \sim p(\tilde{W}^{\text{src}} | O)} \left[\log \prod_{n=1}^N p(w_n | W_{<n}, \tilde{W}^{\text{src}}, O) \right]}_{\triangleq \mathcal{L}^{\text{aed-llm}}}. \quad (14)
 \end{aligned}$$

To handle the intractable summation in Eq. (13), we approximate it with an expectation under the sampling distribution $p(\tilde{W}^{\text{src}} | O)$, which leads to the upper bound shown in Eq. (14). In practice, \tilde{W}^{src} is sampled by running the encoder in training mode (with dropout enabled) and performing best path decoding within the CTC framework [45]. Aside from this sampling procedure, the loss computation follows that of the standard AED decoder (in Eq. (3)).

3.4. Relationship to Prior Work on Two-Pass ST

Equation (6) can be viewed as a general formulation of *two-pass* modeling, which decomposes the ST objective into ASR and MT tasks for tractable optimization. Previous studies have explored such two-pass structures, mainly focusing on the tight coupling of pre-trained ASR and MT models through joint training [46–49]. In contrast, our approach uses an LLM solely for providing translation guidance in ST, meaning the MT component is not explicitly optimized with ASR information. This enables the use of parameter-heavy LLMs without fine-tuning.

4. Experiments

4.1. Experimental Setup

We implemented our models and conducted experiments using the ESPnet-ST toolkit [50, 51].

Data We trained and evaluated models using CoVoST-2 [52], a large-scale multilingual ST corpus derived from Common Voice [53]. This dataset includes translations from English into 15 languages (En \rightarrow X), with each task providing approximately 430 hours of training data. Our experiments focused on the En \rightarrow {De, Zh, Ja} tasks. CoVoST-2 also supports translations from 21 languages into English (X \rightarrow En), with the amount of training data varying across languages. Specifically, we used the high-resource De \rightarrow En task (184 hours), the mid-resource Zh \rightarrow En task (10 hours), and the low-resource Ja \rightarrow En task (1 hour). All data pre-processing followed the ESPnet recipe for CoVoST-2 [54]. However, for Ja \rightarrow En, we used Kana (Japanese phonological units) as the source transcript to mitigate limited training data, and for all tasks, we tokenized the target text using the LLM vocabulary.

Modeling We developed our baseline model (**AED-ST**) within the standard AED-based framework (as described in Sec. 2). The encoder (in Eq. (2)) consisted of a frozen XLS-R (0.3B) front-end [55] followed by 12 Conformer encoder blocks [56]. The encoder blocks are configured with $D^{\text{h}} = 4$ attention heads, the dimension of a self-attention layer of $D = 256$, the dimension of a feed-forward network of $D^{\text{ff}} = 1024$, and the kernel size of 31. The decoder (in Eq. (3)) employed Transformer decoder blocks [39], with parameters ($D^{\text{h}} = 4$,

$D = 256$, $D^{\text{ff}} = 2048$). In our proposed model (**LLM-Guided AED-ST**), the LLM-guided decoder (in Eq. (8)) retained the same configuration as the standard decoder, while Llama2-Chat (7B) [57] served as the front-end LM. The pre-trained models were accessed through the HuggingFace library [58]: XLS-R [59] and Llama2-Chat [60]. For further analysis, we also developed an AED-based MT model (**AED-MT**) based on Transformer. The model had 6 blocks each in the encoder and decoder, with parameters ($D^{\text{h}} = 4$, $D = 256$, $D^{\text{ff}} = 2048$).

Training and Decoding We primarily followed optimization configurations specified in the ESPnet recipe [54]. All tasks were treated independently, without any multilingual training. Before training the baseline AED-ST, we first trained the joint CTC/AED-based ASR model (**AED-ASR**) using only the loss \mathcal{L}^{asr} in Eq. (5). Then, we initialized the AED-ST training process (with \mathcal{L}^{st} in Eq. (5)) using the encoder from AED-ASR. For evaluation, the ST models used beam search decoding with a beam size of 5, while the ASR models used joint CTC/AED decoding [44] with a beam size of 20 and a CTC weight of 0.3. **LLM Prompting** In the proposed LLM-Guided AED-ST, the instruction W^{ins} and the ASR hypothesis \tilde{W}^{src} (see Eq. (9)) were given to the LLM as follows:

```

<s>[INST] <<SYS>>
You will receive a statement in ${SRC_LNG} enclosed in
quotation marks. Please translate it into ${TGT_LNG}.
<</SYS>>
Translate "${ASR_HYP}" [/INST] Here's the translation:

```

The prompt was adjusted to encourage the LLM to generate the translation immediately after the prefix input.

4.2. Main Results

Table 1 lists case-sensitive detokenized BLEU scores [64] for the test set of each task, computed using SacreBLEU [65]. Our proposed LLM-Guided AED-ST significantly outperformed the baseline AED-ST. The performance improvements were more pronounced for the X \rightarrow En tasks compared to the En \rightarrow X tasks, reflecting Llama2’s stronger proficiency in generating English text. Importantly, these gains were achieved by training only the LLM-guided decoder ($\sim 20\text{M}$ parameters), showing efficient end-to-end ST training by incorporating the frozen LLM.

To evaluate the LLM’s MT capabilities, we developed cascaded ST systems that combine AED-ASR with either the LLM or AED-MT. Table 2 reports the corresponding BLEU scores. Note that while the LLM operates in a zero-shot manner on the CoVoST-2 tasks, AED-MT is trained on CoVoST-2 using task-specific training data. On the De \rightarrow En task, the LLM-based cascaded system achieved performance comparable to our end-to-end system, demonstrating robust zero-shot translation even in the presence of ASR errors. Conversely, for En \rightarrow De, the LLM was not effective; even with reference text, its results fell short of the proposed model’s performance. For a fairer comparison, Sec. 4.3 evaluates a cascaded system where the LLM is explicitly adapted to CoVoST-2. The cascaded system using AED-MT outperformed the LLM-based system on En \rightarrow De, likely due to the availability of ample training data specific to the MT task. This raises the question of whether the LLM or AED-MT is more effective for constructing our proposed model, which is further examined in Sec. 4.3.

In Table 1, we also list previous end-to-end ST results, all obtained using only CoVoST-2 as supervised ST data. Although direct comparisons are difficult, given the wide variations in architectures and optimization strategies driven by available computational resources, our findings suggest potential directions

Table 1: BLEU scores on $X \rightarrow \{De, Zh, Ja\}$ and $\{De, Zh, Ja\} \rightarrow X$ tasks in CoVoST-2, comparing baseline ST model (AED-ST) with our proposed ST model using LLM (LLM-Guided AED-ST). For reference, we also report scores of previous models trained exclusively on CoVoST-2 tasks, with numbers obtained from original papers. *Evaluated using character-level BLEU. †Used without fine-tuning.

End-to-End ST Model	Pre-Trained		X → En			En → X		
	Speech Model	Language Model	De	Zh	Ja	De	Zh*	Ja (Ja*)
AED-ST	XLS-R† (0.3B)	–	23.9	8.7	2.8	23.0	32.2	26.6 (37.8)
LLM-Guided AED-ST	XLS-R† (0.3B)	Llama2-Chat† (7B)	30.9	12.4	4.6	26.0	35.3	27.8 (39.2)
XLS-R [55]	XLS-R (0.3B)	mBART (0.5B)	26.7	4.9	0.6	23.6	33.5	– (36.9)
	XLS-R (2B)	mBART (0.5B)	33.6	9.4	3.5	28.3	38.5	– (41.5)
mSLAM [61]	mSLAM-CTC (2B)	–	35.9	10.0	3.3	–	–	– (–)
CoT-ST [62]	Whisper-L-v3 Enc.† (0.8B)	Qwen2 (7B)	–	–	–	28.7	47.7	30.8 (–)
LLaST [63]	Whisper-L-v2 Enc. (0.8B)	Llama2-Chat (13B)	41.2	24.8	28.8	–	–	– (–)

Table 2: BLEU scores for cascaded ST systems.

Model	De → En	En → De
LLM-Guided AED	30.9	26.0
AED-ASR → LLM (zero-shot)	29.7	17.2
Reference → LLM (zero-shot)	34.3	22.6
AED-ASR → AED-MT	21.2	21.6
Reference → AED-MT	24.0	28.8

Table 3: Ablation study on proposed LLM-Guided AED-ST.

Model	CoVoST-2		MuST-C-v2
	De → En	En → De	En → De
AED-ST	23.9	23.0	12.4
LLM-Guided AED-ST	30.9	26.0	17.5
(A1) Remove Prompt	24.5	22.7	–
(A2) Remove Cross-Attention	28.6	23.0	–
(A3) Remove XLS-R	29.1	25.7	–
(A4) Replace LLM with AED-MT	21.5	22.2	11.8

for further improvement. The XLS-R results [55] are derived from an AED-based model that fine-tunes pre-trained XLS-R and mBART models applied to the encoder and decoder, respectively. Using XLS-R (0.3B), our model outperforms these results without requiring any fine-tuning of pre-trained models, highlighting the effectiveness of our end-to-end ST formulation guided by an LLM. Employing larger and more sophisticated pre-trained speech models, e.g., XLS-R (2B) and mSLAM [61], could further enhance our model. CoT-ST [62] and LLaST [63] directly fine-tune LLMs for ST using parameter-efficient techniques (e.g., soft-prompting [28, 29] and LoRA [30]). While increasing training costs, similar strategies could be applied to our approach to help the LLM better handle translation tasks affected by ASR errors. Employing the Whisper encoder [66], trained by supervised ST tasks across various languages, is another promising direction for future enhancement.

4.3. Ablation Study

Table 3 reports BLEU scores for the De → En and En → De tasks, summarizing the results on various ablation studies (A1~A4) performed on the proposed LLM-Guided AED-ST.

LLM prompt matters. We trained our proposed model without the instruction W^{ins} and the ASR hypothesis \tilde{W}^{src} (A1), thereby the LLM functioning as a simple LM. This modification greatly degraded the scores and yielded only marginal improvements over the baseline AED-ST. This highlights the importance of designing a prompt that elicits the translation task [18].

Speech information is necessary. To evaluate the effectiveness of end-to-end training (as formulated in Eq. (7)), we ablated the cross-attention layers from the LLM-guided decoder (A2), thus ignoring speech cues from the encoder. This makes the decoder act as an adapter for the LLM outputs, resulting in a cascaded system with LLM translations adapted to the CoVoST-2 tasks. The ablation led to a decrease in the performance of the proposed model, demonstrating the importance of directly accessing speech information (i.e., conditioning of O in Eq. (7)) rather than relying solely on the ASR hypothesis. Furthermore, when compared with the unadapted cascaded system reported in Table 2, the LLM adaptation had a negative impact on De → En, with the BLEU score declining from 29.7 to 28.6. In contrast, it was beneficial for En → De, with the score improved from 17.2 to 23.0. This suggests that an adaptive mechanism is crucial for the LLM when generating text in a less proficient language.

LLM is the key to improvement. To assess the contribution of the LLM in the proposed model, we first trained it without the pre-trained XLS-R frontend (A3). This led to a slight drop in BLEU scores, suggesting that the performance gains were primarily attributable to the use of the pre-trained LLM. We then replaced the LLM with AED-MT (A4). Specifically, in Eq. (8), e_n is generated by the decoder of AED-MT using the ASR-hypothesis as input. This end-to-end training with AED-MT improved upon the cascaded system shown in Table 2, showing the advantages of tightly coupled AED-ASR and AED-MT training, as similarly demonstrated in [48]. However, it significantly degraded performance compared to our LLM-Guided AED-ST, indicating that the formulation in Eq. (7) greatly benefits from the robust and general translation capabilities, which are successfully realized through the LLM. We further confirmed the benefits of the LLM’s generalizability through an out-of-domain evaluation using the MuST-C-v2 test set (tst-COMMON) [67], where our LLM-Guided AED-ST achieved larger gains than its performance on in-domain CoVoST-2.

5. Conclusion

We proposed a novel end-to-end ST model that employed an LLM to guide its direct translation process. Specifically, we prompted the LLM to translate the ASR hypothesis and then used the resulting output to generate the final translation. Our approach successfully capitalized on the LLM’s accurate and robust MT capabilities, while achieving end-to-end modeling of ST. One limitation of the proposed model is its reliance on a particular LLM used during training. Future work should enable seamless integration with any LLM to allow easy adaptation to rapidly evolving models, thereby maximizing the advantages of the proposed model that does not require LLM fine-tuning.

6. References

- [1] A. Radford *et al.*, “Language models are unsupervised multitask learners,” *OpenAI blog*, vol. 1, no. 8, p. 9, 2019.
- [2] T. L. Scao *et al.*, “BLOOM: A 176B-parameter open-access multilingual language model,” *hal-03850124*, 2022.
- [3] A. Chowdhery *et al.*, “PaLM: Scaling language modeling with pathways,” *Journal of Machine Learning Research*, vol. 24, no. 240, pp. 11 324–11 436, 2023.
- [4] OpenAI, “GPT-4 technical report,” *arXiv preprint arXiv:2303.08774*, 2023.
- [5] Gemma Team, “Gemma: Open models based on Gemini research and technology,” *arXiv preprint arXiv:2403.08295*, 2024.
- [6] A. Dubey *et al.*, “The Llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024.
- [7] T. Brown *et al.*, “Language models are few-shot learners,” in *Proc. NeurIPS*, 2020, pp. 1877–1901.
- [8] L. Ouyang *et al.*, “Training language models to follow instructions with human feedback,” in *Proc. NeurIPS*, 2022, pp. 27 730–27 744.
- [9] J. Wei *et al.*, “Emergent abilities of large language models,” *Transactions on Machine Learning Research*, 2022.
- [10] —, “Finetuned language models are zero-shot learners,” in *Proc. ICLR*, 2022.
- [11] H. W. Chung *et al.*, “Scaling instruction-finetuned language models,” *Journal of Machine Learning Research*, vol. 25, no. 70, pp. 1–53, 2024.
- [12] T. Fang *et al.*, “Is ChatGPT a highly fluent grammatical error correction system? a comprehensive evaluation,” *arXiv preprint arXiv:2304.01746*, 2023.
- [13] R. Ma *et al.*, “Can generative large language models perform asr error correction?” *arXiv preprint arXiv:2307.04172*, 2023.
- [14] C.-H. H. Yang *et al.*, “Generative speech recognition error correction with large language models and task-activating prompting,” in *Proc. ASRU*, 2023.
- [15] G. Song *et al.*, “Contextual spelling correction with large language models,” in *Proc. ASRU*, 2023.
- [16] S. Radhakrishnan *et al.*, “Whispering LLaMA: A cross-modal generative error correction framework for speech recognition,” in *Proc. EMNLP*, 2023, pp. 10 007–10 016.
- [17] C. Chen *et al.*, “HyPoradise: an open baseline for generative speech recognition with large language models,” in *Proc. NeurIPS*, 2023, pp. 31 665–31 688.
- [18] Y. Higuchi, T. Ogawa, and T. Kobayashi, “Harnessing the zero-shot power of instruction-tuned large language model in end-to-end speech recognition,” in *Proc. ICASSP*, 2025.
- [19] C.-H. H. Yang *et al.*, “Large language model based generative error correction: A challenge and baselines for speech recognition, speaker tagging, and emotion recognition,” in *Proc. SLT*, 2024, pp. 371–378.
- [20] G. Li, L. Chen, and K. Yu, “How ChatGPT is robust for spoken language understanding?” in *Proc. Interspeech*, 2023, pp. 2163–2167.
- [21] M. He and P. N. Garner, “Can ChatGPT detect intent? evaluating large language models for spoken language understanding,” in *Proc. Interspeech*, 2023, pp. 1109–1113.
- [22] Z. Zhu *et al.*, “Zero-shot spoken language understanding via large language models: A preliminary study,” in *Proc. LREC-COLING*, 2024, pp. 17 877–17 883.
- [23] P. Dighe *et al.*, “Leveraging large language models for exploiting ASR uncertainty,” in *Proc. ICASSP*, 2024, pp. 12 231–12 235.
- [24] J. Wu *et al.*, “On decoder-only architecture for speech-to-text and large language model integration,” in *Proc. ASRU*, 2023, pp. 1–8.
- [25] Z. Huang *et al.*, “Speech translation with large language models: An industrial practice,” *arXiv preprint arXiv:2312.13585*, 2023.
- [26] Z. Chen *et al.*, “SALM: Speech-augmented language model with in-context learning for speech recognition and translation,” in *Proc. ICASSP*, 2024, pp. 13 521–13 525.
- [27] C.-W. Huang *et al.*, “Investigating decoder-only large language models for speech-to-text translation,” in *Proc. Interspeech*, 2024, pp. 832–836.
- [28] B. Lester, R. Al-Rfou, and N. Constant, “The power of scale for parameter-efficient prompt tuning,” in *Proc. EMNLP*, 2021, pp. 3045–3059.
- [29] M. Tsimpoukelli *et al.*, “Multimodal few-shot learning with frozen language models,” in *Proc. NeurIPS*, 2021, pp. 200–212.
- [30] E. J. Hu *et al.*, “LoRA: Low-rank adaptation of large language models,” in *Proc. ICLR*, 2022.
- [31] W. Jiao *et al.*, “Is ChatGPT a good translator? yes with GPT-4 as the engine,” *arXiv preprint arXiv:2301.08745*, 2023.
- [32] L. Wang *et al.*, “Document-level machine translation with large language models,” in *Proc. EMNLP*, 2023, pp. 16 646–16 661.
- [33] N. Robinson *et al.*, “ChatGPT MT: Competitive for high-(but not low-) resource languages,” in *Proc. WMT*, 2023, pp. 392–418.
- [34] W. Zhu *et al.*, “Multilingual machine translation with large language models: Empirical results and analysis,” in *Proc. Findings of NAACL*, 2024, pp. 2765–2781.
- [35] A. Sriram *et al.*, “Cold fusion: Training seq2seq models together with language models,” in *Proc. Interspeech*, 2018, pp. 387–391.
- [36] C. Shan *et al.*, “Component fusion: Learning replaceable language model component for end-to-end speech recognition system,” in *Proc. ICASSP*, 2019, pp. 5361–5635.
- [37] J. K. Chorowski *et al.*, “Attention-based models for speech recognition,” in *Proc. NeurIPS*, 2015, pp. 577–585.
- [38] W. Chan *et al.*, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *Proc. ICASSP*, 2016, pp. 4960–4964.
- [39] A. Vaswani *et al.*, “Attention is all you need,” in *Proc. NeurIPS*, 2017, pp. 6000–6010.
- [40] A. Graves *et al.*, “Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks,” in *Proc. ICML*, 2006, pp. 369–376.
- [41] R. J. Weiss *et al.*, “Sequence-to-sequence models can directly translate foreign speech,” in *Proc. Interspeech*, 2017, pp. 2625–2629.
- [42] A. Bérard *et al.*, “End-to-end automatic speech translation of audio-books,” in *Proc. ICASSP*, 2018, pp. 6224–6228.
- [43] P. Bahar, T. Bieschke, and H. Ney, “A comparative study on end-to-end speech to text translation,” in *Proc. ASRU*, 2019, pp. 792–799.
- [44] S. Watanabe *et al.*, “Hybrid CTC/attention architecture for end-to-end speech recognition,” *IEEE Journal of Selected Topics in Signal Processing*, vol. 11, no. 8, pp. 1240–1253, 2017.
- [45] Y. Higuchi *et al.*, “BERT meets CTC: New formulation of end-to-end speech recognition with pre-trained masked language model,” in *Proc. Findings of EMNLP*, 2022.
- [46] A. Anastasopoulos and D. Chiang, “Tied multitask learning for neural speech translation,” in *Proc. NAACL-HLT*, 2018, pp. 82–91.
- [47] M. Sperber *et al.*, “Attention-passing models for robust and data-efficient end-to-end speech translation,” *Transactions of the Association for Computational Linguistics*, vol. 7, pp. 313–325, 2019.
- [48] P. Bahar *et al.*, “Tight integrated end-to-end training for cascaded speech translation,” in *Proc. SLT*, 2021, pp. 950–957.
- [49] S. Dalmia *et al.*, “Searchable hidden intermediates for end-to-end models of decomposable sequence tasks,” in *Proc. NAACL-HLT*, 2021, pp. 1882–1896.
- [50] S. Watanabe *et al.*, “ESPnet: End-to-end speech processing toolkit,” in *Proc. Interspeech*, 2018, pp. 2207–2211.
- [51] H. Inaguma *et al.*, “ESPnet-ST: All-in-one speech translation toolkit,” in *Proc. ACL: System Demonstrations*, 2020, pp. 302–311.
- [52] C. Wang *et al.*, “CoVoST 2 and massively multilingual speech translation,” in *Proc. Interspeech*, 2021, pp. 2247–2251.
- [53] R. Ardila *et al.*, “Common voice: A massively-multilingual speech corpus,” in *Proc. LREC*, 2020, pp. 4218–4222.
- [54] “ESPnet CoVoST-2 recipe,” <https://github.com/espnet/espnet/tree/master/egs2/covost2>, [Online; Accessed on December-16-2024].
- [55] A. Babu *et al.*, “XLS-R: Self-supervised cross-lingual speech representation learning at scale,” in *Proc. Interspeech*, 2022, pp. 2278–2282.
- [56] A. Gulati *et al.*, “Conformer: Convolution-augmented Transformer for speech recognition,” in *Proc. Interspeech*, 2020, pp. 5036–5040.
- [57] H. Touvron *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” *arXiv preprint arXiv:2307.09288*, 2023.
- [58] T. Wolf *et al.*, “Transformers: State-of-the-art natural language processing,” in *Proc. EMNLP: System Demonstrations*, 2020, pp. 38–45.
- [59] “facebook/wav2vec2-xls-r-300m,” <https://huggingface.co/facebook/wav2vec2-xls-r-300m>, [Online; Accessed on January-13-2025].
- [60] “meta-llama/llama-2-7b-chat-hf,” <https://huggingface.co/meta-llama/llama-2-7b-chat-hf>, [Online; Accessed on January-13-2025].
- [61] A. Bapna *et al.*, “mSLAM: Massively multilingual joint pre-training for speech and text,” *arXiv preprint arXiv:2202.01374*, 2022.
- [62] Y. Du *et al.*, “CoT-ST: Enhancing LLM-based speech translation with multimodal chain-of-thought,” *arXiv preprint arXiv:2409.19510*, 2024.
- [63] X. Chen *et al.*, “LLaST: Improved end-to-end speech translation system leveraged by large language models,” in *Findings of ACL*, 2024, pp. 6976–6987.
- [64] K. Papineni *et al.*, “BLEU: a method for automatic evaluation of machine translation,” in *Proc. ACL*, 2002, pp. 311–318.
- [65] M. Post, “A call for clarity in reporting BLEU scores,” in *Proc. WMT: Research Papers*, 2018, pp. 186–191.
- [66] A. Radford *et al.*, “Robust speech recognition via large-scale weak supervision,” in *Proc. ICML*, 2023, pp. 28 492–28 518.
- [67] M. A. Di Gangi *et al.*, “MuST-C: a multilingual speech translation corpus,” in *Proc. NAACL-HLT*, 2019, pp. 2012–2017.