Blending LLMs into Cascaded Speech Translation: KIT's Offline Speech Translation System for IWSLT 2024

Sai Koneru, Thai-Binh Nguyen, Ngoc-Quan Pham, Danni Liu, Zhaolin Li,
Alexander Waibel, Jan Niehues

Karlsruhe Institute of Technology

firstname.lastname@kit.edu

Abstract

Large Language Models (LLMs) are currently under exploration for various tasks, including Automatic Speech Recognition (ASR), Machine Translation (MT), and even End-to-End Speech Translation (ST). In this paper, we present KIT's offline submission in the constrained + LLM track by incorporating recently proposed techniques that can be added to any cascaded speech translation. Specifically, we integrate Mistral-7B[1] into our system to enhance it in two ways. Firstly, we refine the ASR outputs by utilizing the N-best lists generated by our system and fine-tuning the LLM to predict the transcript accurately. Secondly, we refine the MT outputs at the document level by fine-tuning the LLM, leveraging both ASR and MT predictions to improve translation quality. We find that integrating the LLM into the ASR and MT systems results in an absolute improvement of 0.3% in Word Error Rate and 0.65% in COMET for tst2019 test set. In challenging test sets with overlapping speakers and background noise, we find that integrating LLM is not beneficial due to poor ASR performance. Here, we use ASR with chunked long-form decoding to improve context usage that may be unavailable when transcribing with Voice Activity Detection segmentation alone.

1 Introduction

This paper provides an overview of Karlsruhe Institute of Technology's speech translation (ST) system developed for the offline track of IWSLT 2024. We participated in the constrained plus large language models (LLMs) condition, focusing on the translation direction from English to German. Under this condition, LLMs with parameters of around 7 billion are allowed, and they have proven effective in many NLP tasks. One of the interesting aspects of this condition is how one can effectively integrate them into ST systems.

In recent years, there has been a significant interest in developing several open-sourced and medium-scale LLMs (Touvron et al., 2023; Jiang et al., 2023). The adaptability of LLMs to diverse tasks, using techniques such as In-Context-Learning (Brown et al., 2020) or Parameter-efficient fine-tuning with 4-bit quantization (Hu et al., 2021; Dettmers et al., 2024), enables their exploitation even with limited resources.

With these recent advancements, exploiting LLMs for ST shows great promise and offers several potential benefits. For instance, one common challenge in Automatic Speech Recognition (ASR) is dealing with input noise, which can often render it difficult to comprehend the speaker's words. However, LLMs, trained on vast amounts of data, may excel at predicting words compared to decoders trained solely during ASR. Moreover, LLMs possess a richer vocabulary and understanding of complex terminology that task-specific ASR systems may lack. Motivated by these advantages, various studies have explored the integration of LLMs into ASR (Chen et al., 2024; Pu et al., 2023), Machine Translation (MT) (Koneru et al., 2023), and ST (Hu et al., 2024).

Chen et al. (2024) employ the LLM to generate a new hypothesis based on the N-best list of the ASR model. This strategy relies on the observation that N-best lists tend to exhibit enough diversity, especially during uncertain conditions, allowing accurate transcript prediction by examining the list. On the other hand, for MT, Koneru et al. (2023) proposes leveraging the LLM to automatically postedit translations by analyzing the source and hypothesis documents to rectify contextual errors. Both approaches are system-agnostic and have demonstrated successful enhancement of system quality. Furthermore, it is also the case that cascaded systems are shown to be superior than end to end systems in previous IWSLT findings and submissions making the leveraging of LLMs

[1] mistralai/Mistral-7B-Instruct-v0.1

Proceedings of the 21st International Conference on Spoken Language Translation (IWSLT 2024), pages 231–239
August 15-16, 2024 ©2024 Association for Computational Linguistics

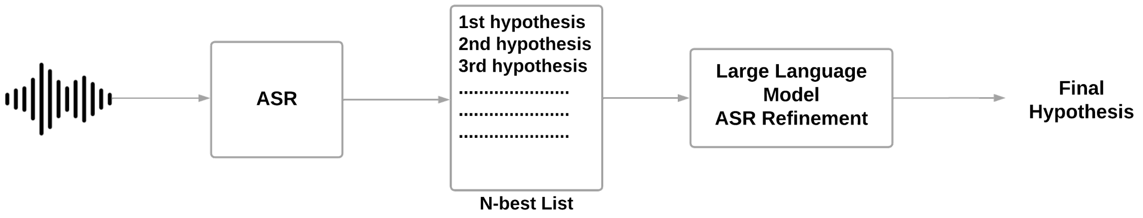

Figure 1: **ASR Refinement**: The ASR system generates a few candidate hypotheses with beam search, and the LLM generates a new hypothesis based on all the candidates as proposed in Chen et al. (2024). We use the top 5 candidates in all our experiments.

easily compatible. (Agarwal et al., 2023; Liu et al., 2023).

Our system builds on these two approaches to effectively use the LLMs to improve the cascaded ST pipeline by refining the intermediate outputs at both ASR and MT while maintaining its modular structure. We utilize pre-trained models to create the individual components and fine-tune them with the allowed data. Specifically, we employ WavLM (Chen et al., 2022) and MBART50 (Liu et al., 2020) to initialize the ASR, and NLLB-200 (3.3B) (Costa-jussà et al., 2022) for the MT module. As for the LLM, we opt for Mistral 7B Instruction-Tuned (Jiang et al., 2023), considering it to be the most recent model within the allowable options.

We present our main findings below:

- We demonstrate that LLMs can be tailored to enhance both ASR (Section 4.1) and MT systems (Section 4.2), resulting in an absolute improvement of 0.3% in Word Error Rate and 0.65% in COMET, respectively, on the tst2019 test set.

- While we observe significant enhancements in in-domain scenarios, we find that these techniques are not applicable in challenging scenarios (such as Overlapping Speakers, Background noise, etc.) due to poor ASR performance.

- We demonstrate that employing chunked long-form decoding[2] significantly improves ASR performance in challenging scenarios, such as the case of the ITV dev set. Specifically, we observe a decrease in the word error rate from 37.83% to 30.98%

2 Data

This section describes the evaluation and training data we use in our experiments. For evaluation, we report results on the tst2019 and ACLdev (Salesky et al., 2023) test sets to compare with findings from previous works (Anastasopoulos et al., 2021; Agarwal et al., 2023). We also use the EPTV (European Parlament activities), Itv (TV Series), and Peloton (Fitness TV) dev sets from the subtitling track consisting of overlapping speakers with different accents to evaluate the ASR performance in challenging scenarios.

As the data conditions did not change from IWSLT23 to this year, we rely on the data processed from last year's submission (KIT'23) (Liu et al., 2023). For the training data of ASR, we use the same system that used Common Voice (Ardila et al., 2020), LibriSpeech (Panayotov et al., 2015), MuST-C v2 (Di Gangi et al., 2019), TED-LIUM v3 (Hernandez et al., 2018), and VoxPopuli (Wang et al., 2021).

While for MT fine-tuning, we use the cleaned training data from last year created from the available parallel data. This includes Europarl v7 and v10 (Koehn, 2005), NewsCommentary v16, OpenSubtitles v2018 (Lison and Tiedemann, 2016), Tatoeba (Tiedemann, 2012), ELRC-CORDIS_News and TED2020 (Reimers and Gurevych, 2020) and consists in total of 23 million sentence pairs. For the rest of the paper, we refer to the full parallel data as *seed* and TED2020 as *in-domain*.

3 Overview

In this section, we provide an overview of our proposed cascaded system, detailing each individual component. First, the input audio is sent to the ASR system, which undergoes segmentation, and N-best lists are generated for each segmented utter-

[2] We derive the terminology from this blog post.

ance. Next, the top candidates in the N-best list are fed as input to the LLM, which is trained to refine the ASR output and generate a final ASR hypothesis. Following this, the final ASR hypotheses are passed on to the sentence-level MT system, which produces translations. Finally, the sentence-level automatic transcripts and translations are fed into another adapted LLM, which automatically post-edits and generates a coherent document translation of the talk.

3.1 Automatic Speech Recognition

We employed the ASR model from our previous year's submission (Liu et al., 2023), considering its effectiveness in transcribing the TED domain. For initialization, we utilized WavLM and mBART50 for the encoder and decoder, respectively, before fine-tuning on the ASR data described in Section 2. However, we encountered below-par ASR performance on the challenging sets EPTV, Itv, and Peloton.

We identified several issues that hindered the effectiveness of our ASR model with these sets. Firstly, the model itself was trained on single-talker datasets but inferred with multi-talker noisy datasets, leading to a mismatch in data distribution. Secondly, our typical use of the SHAS model for audio segmentation introduced challenges, as it sometimes missed segmentations and overlooked segments containing human speech.

Data shift is difficult to handle when the training dataset has not changed since last year. We focused more on handling the latter by incorporating long-form decoding. The key idea is to better use context (at the text or signal level) for decoding. The long audio file is chunked into smaller segments with a small overlap between adjacent segments. The model is run over each chunk, and the inferred text is joined at the strides by finding the longest common sequence between overlaps.

3.2 ASR refinement

Once we have generated the N-best list, we select the top 5 candidates and utilize an LLM to produce the final hypothesis as shown in Figure 1. In this step, we can adapt the LLM to the task using either few-shot prompting or LoRA fine-tuning techniques. We choose to fine-tune the LLM with adapters based on the findings from (Chen et al., 2024). However, it is crucial to train the LLM under conditions that simulate the test environment, where it should fix errors of our ASR output rather than on the whisper generated in Chen et al. (2024).

To generate the dataset for fine-tuning, we perform inference on our in-domain training data using the gold segmentation. We create pairs comprising the N-best list and the corresponding reference. It is worth noting that we utilized the same data to train the ASR system, which is not ideal. However, resource constraints prevented us from following the augmentation procedure that mitigates this, which we explain further in Section 3.4. Despite this limitation, manual analysis revealed that the ASR did not memorize the training data and produced similar N-best lists to those observed in the test conditions.

Following this, we fine-tuned the Mistral 7B Instruction-tuned LLM (Jiang et al., 2023) using QLoRA (Dettmers et al., 2024), to predict the gold reference based on the top candidates (see the prompt format below). Importantly, we chose not to shuffle the order of the top candidates when providing it in the prompt, as doing so would eliminate the ranking information provided to the LLM, which could be crucial for its performance.

```
Punctuate and Post-edit the hypothesis
based on the predictions:
Hyp 1 <SS> Hyp 2 <SS> Hyp 3 ..
Post-edited Hypothesis:
Gold Reference
```

3.3 Machine Translation

For building the MT system, we leverage the strong pre-trained model NLLB 200 3.3B (Costa-jussà et al., 2022) that is allowed in the constrained plus LLM track. We perform a two-step fine-tuning approach. Initially, we fine-tune the model on the *seed* data to adapt it to the spoken language domain. Subsequently, in the second step, we conduct in-domain fine-tuning on TED (in-domain) data, given its significance as one of the primary test sets in the offline track. Additionally, we implement checkpoint averaging to improve generalization with the last 3 checkpoints.

3.3.1 Restoring Punctuations

It is important to note that the ASR outputs lack punctuation. Therefore, we conducted experiments with two punctuators. First, we utilized the punctuations generated from the LLM ASR refinement process described in Section 3.2. Second, we employed a DeltaLM-based punctuation model, which was utilized in our previous year's submission (Liu

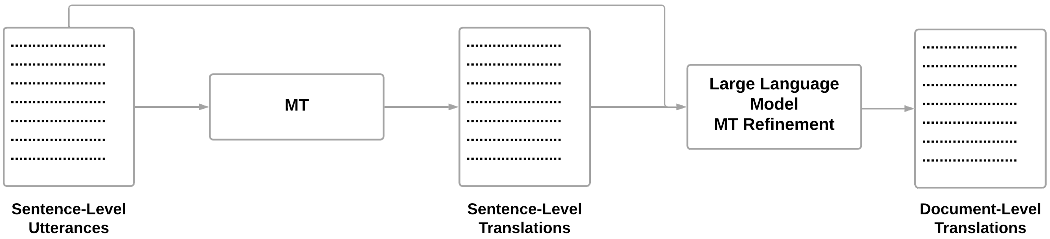
MT
Large Language Model MT Refinement
Sentence-Level Utterances
Sentence-Level Translations
Document-Level Translations

Figure 2: **Document Level MT Refinement**: The LLM trained to post-edit uses sentence-level transcripts and translations to generate a final document-level coherent and consistent translation.

et al., 2023). We observed that while the punctuations generated by the LLM were semantically correct, they often resulted in long sequences and led to a degradation in MT performance. As a result, we decided to opt for the second choice and segment the text into sentences using manually crafted rules.

3.4 Document-level Automatic Post-Editing

After translating the individual sentences with the fine-tuned NLLB, the outputs are not coherent as they are translated in isolation. Moreover, any ASR errors that might be fixed by observing the full document will be translated incorrectly. To mitigate this, we perform an additional step of document-level automatic post-editing using the source transcripts and sentence translations shown in Figure 2.

Similar to the situation outlined in Section 3.2, we encountered a lack of data for fine-tuning the LLM for document-level post-editing. Hence, we adopted the approach proposed by Koneru et al. (2023) to create the dataset. We divided the in-domain TED data into two halves, each containing English audio, English transcript, and German translation. Subsequently, we fine-tuned MT models on each half using the pre-trained models described in Sections 3.1 and 3.3. Following this, we conducted inference using the gold segmentation with our ASR and MT models trained on one half to the other half. This procedure generated a synthetic dataset with noisy ASR input, MT predictions, and corresponding gold references, leveraging the provided segmentation in the data.

We then use the synthetic dataset to create instances of document-level post-editing. We go through each talk and divide the transcripts into chunks, each chunk containing a maximum of 256 tokens corresponding to the LLM tokenizer. Then for each chunk, we use the transcript, hypothesis and reference to transform them into the format below and train the LLM to predict the gold reference given the noisy transcript and sentence-level hypothesis.

```
Noisy English Transcript:
ASR Hyp 1 <SS> ASR Hyp 2 <SS> ....
German Translations:
MT Hyp 1 <SS> MT Hyp 2 <SS> ....
Post-Edited German Translations:
Ref 1 <SS> Ref 2 <SS> ....
```

We use the delimiter "<SS>" to align with the input and perform sentence-level evaluation. Then, we again fine-tune the Mistral 7B Instruction-tuned LLM (Jiang et al., 2023) using QLoRA (Dettmers et al., 2024), training it to predict the gold reference given the noisy transcript and translations. We employ the sliding window with payload strategy during decoding as described in Koneru et al. (2023).

4 Results

4.1 Automatic Speech Recognition

To evaluate the benefit of the additional ASR refinement step described in Section 3.2, we compare the word error rate of our ASR system before and after post-editing, as shown in Table 1. The ASR performance improves in both cases, with a higher absolute improvement observed in the ACLdev set. The LLM is particularly beneficial in the ACLdev set, given that it contains terminology from the scientific domain where the LLM excels. We also observe a relative improvement of 10% in the TED talks, indicating that ASR refinement is beneficial.

	tst2019	ACLdev2023
KIT'23 (Liu et al., 2023) ASR	3.1	11.3
KIT'23 ASR + LLM Refine	2.8	10.6

Table 1: ASR word error rate scores on tst2019 and ACLdev2023 test sets. + LLM refine indicates that the N-best list was post-edited to generate the final hypothesis.

Model	*EPTV*	*ITV*	*Peloton*
KIT'23 ASR	26.43	37.83	**18.93**
KIT'23 ASR + Gold Seg	**16.84**	37.21	25.88
KIT'23 ASR + long-form	17.54	**30.98**	20.79
Seamless v2 (Barrault et al., 2023)	40.94	56.94	43.47

Table 2: ASR word error rate scores on the EPTV, Peloton and ITV dev set. Best scores for each set are highlighted in bold.

However, the performance of the same ASR system on the challenge set was below par. We conducted additional ablation studies and present the results in Table 2 for the challenge dev sets. We compared last year's ASR system with three conditions: providing gold segmentation, utilizing long-form decoding, and using the recently developed Seamless V2 (Barrault et al., 2023).

We observed that providing gold segmentation achieved a score of 16.84, demonstrating its crucial role in handling this challenging set for EPTV. Moreover, long-form decoding significantly narrowed the gap, decreasing the word error rate for both EPTV and ITV. Meanwhile, our ASR shows the best performance for Peloton without any modifications. Additionally, we evaluated Seamless to assess its robustness and found that its performance was severely lacking in comparison.

Based on these results, we use the ASR with standard segmentation for *TED* and *Peloton* test sets. For EPTV and ITV, we use the ASR system with long-form decoding. We found that the LLM cannot refine the N-best list given the poor WER of KIT'23 ASR for the latter test sets and generates long sequences with repetitions for most utterances. **Therefore, we do not perform any ASR or MT LLM refinements for *ITV* and *EPTV* sets and generate translations with a standard cascaded ST pipeline.**

4.2 Cascaded Speech Translation

In this section, we evaluate the final quality of our cascaded ST using the mwerSegmenter to realign the hypothesis with the reference segmentation. We report results with BLEU (Papineni et al., 2002) and Chrf2 (Popović, 2015) computed by Sacrebleu (Post, 2018). We also report the COMET (Rei et al., 2022) score using the default model[3].

Model	*tst2019*		
	BLEU	Chrf2	COMET
KIT'23 TED*	**28.4**	**58.8**	**78.87**
NLLB 3.3B	26.6	57.7	77.41
Seamless v2	25.5	57.0	76.65
NLLB 3.3B + Seed	26.9	57.9	77.87
NLLB 3.3B + Seed + TED	27.6	58.5	78.49

Table 3: MT scores using KIT'23 ASR as input calculated by resegmenting with mwerSegmenter. * indicates an unconstrained system that was trained on the same data sources but in more languages than what is allowed for IWSLT24. TED indicates the model adapted for TED and not ACLdev which was the official submission from KIT for IWSLT23

4.3 Two-step Fine-tuning

We presented a two-step fine-tuning approach to adapt our MT system in Section 3.3 to the target domain. We report the translation quality on tst2019 test set with this approach (last row) and other models for comparison in Table 3.

Firstly, we observe that Seamless performs inferiorly to NLLB across all translation metrics. Consequently, we proceeded with NLLB for further experiments.

Subsequently, fine-tuning the seed parallel data improved quality across all metrics, notably in-

[3] Unbabel/wmt22-comet-da

Model	*tst2019*			ACLdev2023		
	BLEU	Chrf2	COMET	BLEU	Chrf2	COMET
NLLB 3.3	26.6	57.7	77.41	35.0	63.9	74.83
NLLB 3.3 Seed	26.9	57.9	77.87	34.7	63.8	75.67
ASR Refine + NLLB 3.3 Seed	27.3	58.3	78.32	36.1	**65.0**	77.59
ASR Refine + NLLB 3.3 Seed + TED	28.3	58.8	78.98	34.8	63.7	77.25
ASR Refine + NLLB 3.3 Seed + TED + Doc APE	**28.7**	**59.1**	**79.63**	**36.4**	64.5	**78.64**

Table 4: MT scores using KIT'23 ASR as input calculated by resegmenting with mwerSegmenter. ASR Refine indicates an additional ASR refinement step with the LLM. Seed and TED indicate fine-tuning the NLLB 3.3 with seed alone or a two-step process with additional fine-tuning on TED. Doc APE indicates an LLM post-editing refinement to generate a coherent and consistent document. Best scores in each metric and test set are highlighted in bold.

creasing the score from 77.41 to 77.87 in COMET. Following this, with the assistance of second-step fine-tuning, we observed further improvements, resulting in scores reaching 78.49. However, it is important to note that this system still lags behind last year's submission, which was specifically adapted to the TED domain. Nevertheless, it's worth highlighting that this system was trained across multiple languages, placing it in the unconstrained condition for IWSLT24. Moreover, we could not replicate a similar adaptation process for NLLB due to resource and time constraints.

4.4 LLM Refinement

We proposed improving the ASR outputs and converting sentence-level to document-level translations using fine-tuned LLMs. We evaluate the benefits of the individual steps and report the results of our final cascaded ST system in Table 4 on tst2019 and ACLdev2023 test sets.

First, the benefits of two-step fine-tuning, ASR refinement, and document post-editing complement each other. Using KITs 23 ASR with NLLB 3.3 B as a baseline, we obtained 77.41 COMET in tst2019 test set. However, including all enhancements led to a total improvement of 2.23 COMET points. Furthermore, the improvements are consistent with both lexical and neural metrics.

Next, we observed that integrating LLMs provides significant benefits in the ACLdev set compared to the TED dev sets. This is plausible due to scientific terminology and accented speakers in the ACLdev set. Both of these challenges are well-suited for LLMs, as the quality of the initial systems is sufficient to utilize context and rectify mistakes reliably.

5 Conclusion

This system paper presented KIT's submission for the offline track in the constrained + LLMs condition, focusing on the English-to-German translation direction. Using modular techniques, we successfully integrated LLMs into any cascaded ST pipeline. Additionally, we highlighted the benefits of long-form decoding in scenarios involving noisy and overlapping speech.

For future work, we aim to explore robust techniques for integrating LLMs that can effectively handle challenging scenarios where ASR quality is sub-par. Furthermore, the translation's latency is quite high as it needs to call the LLM twice. However, integrating quality estimation techniques to decide when we need the LLM can limit the effects of the high latency problem.

Acknowledgments

This work is partly supported by the Helmholtz Programme-oriented Funding, with project number 46.24.01, named AI for Language Technologies, funding from the pilot program Core-Informatics of the Helmholtz Association (HGF). It also received partial support from the European Union's Horizon research and innovation programme under grant agreement No 101135798, project Meetween (My Personal AI Mediator for Virtual MEETtings BetWEEN People). The work was partly performed on the HoreKa supercomputer funded by the Ministry of Science, Research and the Arts Baden-Württemberg and by the Federal Ministry of Education and Research.

References

Milind Agarwal, Sweta Agarwal, Antonios Anastasopoulos, Luisa Bentivogli, Ondřej Bojar, Claudia Borg, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Mingda Chen, et al. 2023. Findings of the iwslt 2023 evaluation campaign. Association for Computational Linguistics.

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner. 2021. FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 1–29, Bangkok, Thailand (online). Association for Computational Linguistics.

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis M. Tyers, and Gregor Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference, LREC 2020, Marseille, France, May 11-16, 2020*, pages 4218–4222. European Language Resources Association.

Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, et al. 2023. Seamlessm4t-massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*.

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. *Advances in neural information processing systems*, 33:1877–1901.

Chen Chen, Yuchen Hu, Chao-Han Huck Yang, Sabato Marco Siniscalchi, Pin-Yu Chen, and Eng-Siong Chng. 2024. Hyporadise: An open baseline for generative speech recognition with large language models. *Advances in Neural Information Processing Systems*, 36.

Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. 2022. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518.

Marta R Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, et al. 2022. No language left behind: Scaling human-centered machine translation. *arXiv preprint arXiv:2207.04672*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2024. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019. MuST-C: a Multilingual Speech Translation Corpus. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia A. Tomashenko, and Yannick Estève. 2018. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer - 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18-22, 2018, Proceedings*, volume 11096 of *Lecture Notes in Computer Science*, pages 198–208. Springer.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Yuchen Hu, Chen Chen, Chao-Han Huck Yang, Ruizhe Li, Dong Zhang, Zhehuai Chen, and Eng Siong Chng. 2024. Gentranslate: Large language models are generative multilingual speech and machine translators. *arXiv preprint arXiv:2402.06894*.

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*.

Philipp Koehn. 2005. Europarl: A parallel corpus for statistical machine translation. In *Proceedings of Machine Translation Summit X: Papers*, pages 79–86, Phuket, Thailand.

Sai Koneru, Miriam Exel, Matthias Huck, and Jan Niehues. 2023. Contextual refinement of translations: Large language models for sentence and document-level post-editing. *arXiv preprint arXiv:2310.14855*.

Pierre Lison and Jörg Tiedemann. 2016. OpenSubtitles2016: Extracting large parallel corpora from movie and TV subtitles. In *Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16)*, pages 923–929, Portorož, Slovenia. European Language Resources Association (ELRA).

Danni Liu, Thai Binh Nguyen, Sai Koneru, Enes Yavuz Ugan, Ngoc-Quan Pham, Tuan-Nam Nguyen, Tu Anh Dinh, Carlos Mullov, Alexander Waibel, and Jan Niehues. 2023. Kit's multilingual speech translation system for iwslt 2023. *arXiv preprint arXiv:2306.05320*.

Yinhan Liu, Jiatao Gu, Naman Goyal, Xian Li, Sergey Edunov, Marjan Ghazvininejad, Mike Lewis, and Luke Zettlemoyer. 2020. Multilingual denoising pre-training for neural machine translation. *Transactions of the Association for Computational Linguistics*, 8:726–742.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. `https://github.com/huggingface/peft`.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North*. Association for Computational Linguistics.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An ASR corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP 2015, South Brisbane, Queensland, Australia, April 19-24, 2015*, pages 5206–5210. IEEE.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Maja Popović. 2015. chrF: character n-gram F-score for automatic MT evaluation. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 392–395, Lisbon, Portugal. Association for Computational Linguistics.

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

Jie Pu, Thai-Son Nguyen, and Sebastian Stüker. 2023. Multi-stage large language model correction for speech recognition. *arXiv preprint arXiv:2310.11532*.

Ricardo Rei, José G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André F. T. Martins. 2022. COMET-22: Unbabel-IST 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4512–4525, Online. Association for Computational Linguistics.

Elizabeth Salesky, Kareem Darwish, Mohamed Al-Badrashiny, Mona Diab, and Jan Niehues. 2023. Evaluating Multilingual Speech Translation Under Realistic Conditions with Resegmentation and Terminology. In *Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023)*. Association for Computational Linguistics.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. In *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC'12)*, pages 2214–2218, Istanbul, Turkey. European Language Resources Association (ELRA).

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

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 993–1003, Online. Association for Computational Linguistics.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

A Appendix

We use the transformers library (Wolf et al., 2019) for fine-tuning our ASR and LLM and the fairseq toolkit (Ott et al., 2019) for fine-tuning NLLB 3.3B. For the ASR training, we set the *batch size* to 384, resulting in approximately 128 minutes per batch. We employ a *warmup strategy* over $2,000$ steps and a total of $100,000$ training steps. The *learning rate* is initialized to $1e-4$.

For the NLLB fine-tuning experiments, we use a *learning rate* set to $5e-5$, *label smoothing* to 0.1, *drop out* to 0.1, *attention drop-out* to 0.1. We use the *Adam optimizer* with *betas* to $(0,9,0.98)$ and the remaining optimizer parameters to default. We used a *batch size* of maximum 3096 making one step, *update-freq* to 16 and validating on the dev set after every epoch. We stopped the training after the dev loss did not increase after 10 epochs. For fine-tuning the LLM with QLoRA we use the peft (Mangrulkar et al., 2022) along with the transformers library. We add LoRA adapters to the target modules [*q_proj, k_proj, v_proj, o_proj, gate_proj, up_proj, down_proj*]. We set the *adapter rank* to 16, *alpha* to 32 and *lora dropout* to 0.1. We use a *batch size of* 8, *learning rate* of $5e-5$ with other parameters set to default. After every 200 steps, we validate and terminate the training if it does not improve 10 consecutive times.

During inference, we use beam search for all ASR, MT and LLM components. The ASR and MT decode with *beam size* of 5, whereas the LLM does it with *beam size* of 3.