NAVER LABS Europe Submission to the Instruction-following 2026 Short Track

Marcely Zanon Boito[1] **Hemant Yadav**[2] **Jean-Luc Meunier**[1] **Ioan Calapodescu**[1]

[1] NAVER LABS Europe, France [2] IIIT Delhi, India

contact: marcely.zanon-boito@naverlabs.com

Abstract

In this paper, we describe NAVER LABS Europe's submission to the instruction-following speech processing short track at IWSLT 2026. We participate again in the constrained setting, developing systems capable of jointly performing ASR, ST, and SQA from English speech into Chinese, Italian, and German. Building on our previous submission, ranked first in last year's short track, we update our multi-stage training pipeline by replacing the speech projector with `SpeechMapper`, a method for learning a speech-to-LLM embedding projector using only ASR data. In addition, we introduce a synthetic SQA dataset, *fakACL*, composed of artificially generated scientific presentations. This dataset is built by prompting the LLM backbone, segmenting the generated talks, and synthesizing speech with `SeamlessM4T-v2-large`. The combination of an improved speech projection mechanism and domain-specific synthetic data allows our model to outperform last year's best short-track system, while being considerably more compact and relying on a weaker LLM backbone.

1 Introduction

Fueled by the progress of text-only LLMs, a growing number of speech-based assistants have recently been proposed, addressing both semantic (Rubenstein et al., 2023; Tang et al., 2023; Défossez et al., 2024; Hu et al., 2024; Ambilduke et al., 2025; Mohapatra et al., 2026) and acoustic speech tasks (Huang et al., 2024; Nguyen et al., 2025; Xu et al., 2025). In this context, the second edition of the *IWSLT Instruction-following Speech Processing Track* (Adelani et al., 2026) provides a common benchmark for developing speech assistants capable of performing semantic tasks directly from speech. The challenge proposes to leverage LLMs and speech foundation models (SFMs) to build systems that can perform multilingual tasks from English speech input, guided by textual multilingual instructions.

NAVER LABS Europe (NLE) again participates in the constrained setting of the *short track*, which includes automatic speech recognition (ASR), speech translation (ST), multilingual spoken question answering (SQA), and a surprise instruction-following task revealed only at test time. The target languages for ST and SQA are Chinese, Italian, and German. The target languages for the surprise task were German and Chinese. Participants are allowed to use the `SeamlessM4T-v2-large` (Barrault et al., 2023) speech backbone and the `Qwen3-4B-Instruct-2507` (Team, 2025) LLM for both training and data generation. Building on our previous winning submission (Lee et al., 2025), we adopt a similar multi-stage training pipeline to develop a speech LLM capable of following multilingual instructions.

Our approach trains two components in parallel: (1) a speech-to-text projector that maps averaged speech representations from the `SeamlessM4T-v2-large` encoder into the embedding space of a frozen LLM, and (2) text-only LoRA adapters (Hu et al., 2022) applied to the same frozen LLM. These components are subsequently combined through a (3) short SFT stage on multimodal multilingual data. Compared to last year, we introduce two main improvements. First, we replace the speech projector with an updated version of `SpeechMapper` (Mohapatra et al., 2026), which enables learning a speech-to-embedding mapping without requiring LLM forward passes, substantially lowering computational and hardware requirements. Second, we construct a synthetic scientific dataset, *fakACL*, to reduce the domain gap between training and evaluation data.

This system paper is organized as follows. Section 2 describes the preprocessing applied to the data used in this challenge. Section 3 presents the updated `SpeechMapper` model we leverage in

Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026), pages 150–163

this work. Sections 4 and 5 describe our training pipeline and experimental settings, respectively. Section 6 presents our experiments and discussion. Section 7 presents the submitted system. Section 8 concludes the paper.

2 Data

We leverage all available data from the constrained setting for training, including CoVoST2 (Wang et al., 2020), EuroParlST (Iranzo-Sánchez et al., 2020), GigaST (Ye et al., 2023), and LibriSQA (Zhao et al., 2025). This data is also synthetically augmented to cover additional target languages via `SeamlessM4T-v2-large` MT. In addition, we introduce *fakACL*, a synthetic dataset of scientific presentations generated by prompting the LLM backbone and synthesizing the outputs with `SeamlessM4T-v2-large` TTS. The validation sets of EuroParlST, CoVoST2, and LibriSQA are used for model validation, while MCIF (Papi et al., 2026) is used for final model selection.

We now present our data preprocessing (Section 2.1) and prompt format (Section 2.2).

2.1 Data Preprocessing

We create both speech-to-text and text-to-text instructions to train our systems. Appendix Table 5 lists our splits and number of samples. In all cases where synthetic translation is generated from English text, this procedure is done via `SeamlessM4T-v2-large` MT followed by filtering of the result using reference-free COMET[1] (Rei et al., 2022) to filter out all translations that did not score at least 0.85. Below we detail dataset-specific preprocessing.

GigaST, CoVoST2 and EuroParlST CoVoST2 and GigaST cover English to German and Simplified Chinese language directions. EuroParlST covers English to German and Italian. ASR splits for CoVoST2 and EuroParlST datasets were built by merging the existing language splits and deduplicating the audio files. For all, language-specific ST and MT splits were created by aligning translations to English speech and reference transcriptions, respectively. For GigaST, and because our LLM backbone is particularly weak in Italian (see Table 1), we produce additional synthetic Italian translations.

[1] `Unbabel/wmt22-cometkiwi-da`

LibriSQA LibriSQA is built on the 300-hour split of LibriSpeech (Panayotov et al., 2015). The dataset is divided in *PartI* and *PartII*, which share the same speech content, but differ in the format of the questions and answers. *PartI* consists of direct questions to be the answered by the system, while *PartII* presents multiple-choice questions (MCQ) corresponding to those in *PartI*. We performed the following modifications to this dataset:

- **MCQ answer expansion:** For *PartII*, answers are provided as the letter corresponding to the correct option. This results in a very limited number of tokens contributing to the loss, which we found to be suboptimal in past experiments. To address this, we expand the target answers to include the full option text (e.g., instead of predicting "A", the model is trained to produce "A <option A text>").
- **Multilingual SQA/QA:** We translate the English text into the three target languages. For *PartI*, we retain only question-answer pairs whose *both* translations exceed the quality threshold. For *PartII*, we translate only the questions and their associated answer options, since the correct answer is implicitly defined by the options. We then apply a parsing step to discard examples where the A–B–C–D structure is altered or lost during translation. This parser also recovers the correct answer by aligning the translated options with the original references.
- **MT split:** We create an MT split for each target language by exploiting the alignment between questions and answers obtained during the creation of our SQA split. We do not create a corresponding ST split, as we consider this dataset to be significantly out-of-domain for the challenge, particularly in terms of acoustic conditions, and including more of it could be detrimental during training.
- **Similarity-based Invalid splits:** Some questions in LibriSQA are relatively broad (e.g., "How did the person in the text act?"), making it difficult to reliably mismatch them with unrelated audio segments. To address this, we construct mismatched pairs by selecting question–transcript combinations with low semantic similarity. We first shuffle and randomly pair transcripts and questions, then compute

sentence-level representations using a Sentence Transformer[2] to measure cosine similarity between them. We retain only pairs with low similarity scores.[3] This process yields 48,346 and 58,682 mismatched pairs for *PartI* and *PartII*, respectively.

- **Qwen-based Invalid splits:** To address potential noise introduced by our similarity-based invalid splits we created, we include a more general-purpose set of invalid questions by prompting `Qwen3-4B-Instruct-2507` (see Appendix A.1 for details). After post-processing, this yields 62,040 English questions, which are then translated and filtered for quality using the same procedure described above. Finally, these questions are randomly paired with speech audio segments to construct the invalid splits.

fakACL In both this and the previous edition of the challenge, the validation data is drawn from scientific paper presentations at ACL conferences. Motivated by this, we constructed a synthetic dataset targeting this domain to reduce the mismatch between training and test conditions, and to provide additional relevant QA data, as the constrained SQA dataset is relatively distant from the test setting. We generated fakACL ASR/ST/SQA data by first producing short presentation scripts for NLP papers using `Qwen3-4B-Instruct-2507`. These scripts were segmented into chunks of two to three sentences and synthesized into speech using `SeamlessM4T-v2-large` TTS. Each segment transcript was then fed back into the LLM to generate two questions answerable from the corresponding content. Appendix A.2 provides full details of the dataset creation process.

2.2 Prompt Format

The goal of the short track of this challenge is to produce a model that is able to 1) transcribe English speech; 2) translate English speech into Italian, German and Chinese; 3) Answer multilingual questions using English speech as input. In this setting, the language of the question must match the language of the answer.

Last year we designed an unified prompt with consistent structure: regardless of the task (ASR, ST, or SQA), the user turn begins by encapsulating the speech embeddings within textual tags. This is followed on a new line by a task-specific instruction formulated as a question in the target language, and finally, another line containing a common suffix. Our last year results (Lee et al., 2025) highlighted that by forcing the instruction (i.e. question) to be in the target language, we were able to reduce confusion between the tasks of ASR/ST, while also helping the system to generalize to languages it was not originally trained for.[4] Therefore, this year we only experiment with this setting, in which the instruction is given in the target language.

We additionally include a template for multiple choice questions (MCQ), for the surprise task, and for zero-shot generation (LLM backbone and `SpeechMapper` in projector-only mode). The list of updated templates used is available in Appendix Table 7.

3 Speech Projector: `SpeechMapper`

In this work we use an improved version of the original `SpeechMapper` projector (Mohapatra et al., 2026). SpeechMapper is a speech-to-LLM embedding projector for semantic information, and training it only requires an LLM's tokenizer and its embedding layer. The benefits are twofold: (1) the size of LLM does not increase GPU memory required for training;[5] (2) it mitigates prompt overfitting. We refer the reader to the original paper for further details.

The decision to replace last year's simple transformer based projector with the more parameter-heavy `SpeechMapper` is primarily motivated by practical considerations. In preliminary experiments, we attempted to reproduce the pipeline from Lee et al. (2025), but observed very poor performance for projector-only models (exceeding 200% WER). We attribute this low-performance compared to last year to the lower dimensionality of the embedding space in this year's LLM. We hypothesize that projecting into a smaller embedding space requires higher precision, and therefore greater model capacity, which `SpeechMapper` is better able to provide.

Below, we describe our main modification to

[2] `sentence-transformers/multi-qa-MiniLM-L6-cos-v1`

[3] Based on manual assessment, we set this to ≤ 0.1.

[4] For instance, in the SpeechMapper paper (Mohapatra et al., 2026), our IWSLT25 system generalizes to Spanish and French ST via multilingual prompting, despite the fact that its LoRA weights were never trained on these languages.

[5] The only increase in computational cost comes from the embedding dimensionality, which is typically not proportional to the LLM size, keeping the approach computationally efficient.

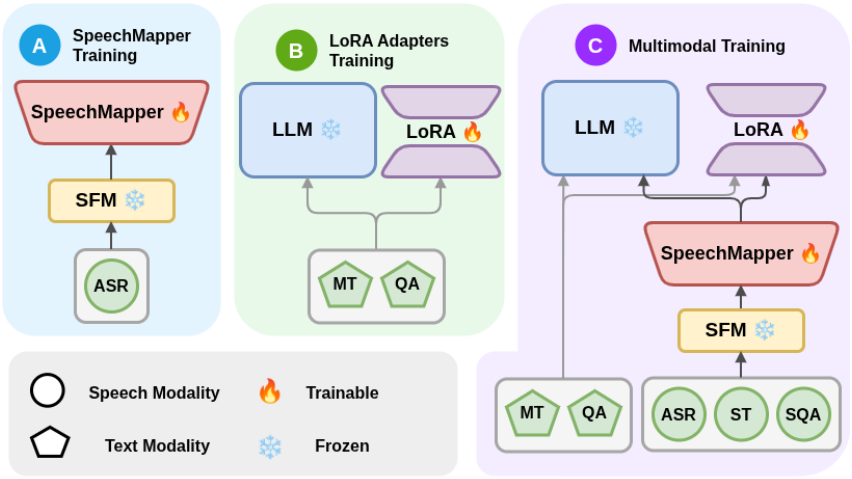

Figure 1: Our training pipeline. A speech projector (SpeechMapper) (A) and text LoRA adapters (B) are trained in parallel using ASR and text-to-text data, respectively. These modules are then integrated during a brief multimodal adaptation step (C).

SpeechMapper relative to Mohapatra et al. (2026): the training objective.

3.1 New SpeechMapper Training Objective

Our updated version of SpeechMapper is trained using four loss functions designed to align speech to LLM input embeddings. Speech is encoded into a sequence of embeddings $Z_s \in \mathbb{R}^{T\times d}$ using the frozen SFM and fed to the SpeechMapper. The corresponding target sentence is processed to a sequence of LLM token embeddings $Z_t \in \mathbb{R}^{T'\times d}$ using the frozen LLM embedding layer. The resulting speech sequence length T is significantly larger than the text sequence length T'. To mitigate this length mismatch, we pad the target embedding sequence with the LLMs [PAD] token embedding to match the speech sequence length (Eq. 1).

$$Z_t = [z_t^{(1)}, \ldots, z_t^{(T')}, z_{\text{pad}}, \ldots, z_{\text{pad}}] \in \mathbb{R}^{T\times d} \tag{1}$$

Therefore, the output of SpeechMapper is semantic embeddings followed by pad embeddings, implicitly capturing sequence length. We optimize the following objectives.

L1 Alignment Loss The element wise L1 distance between speech and text embeddings explicitly enforce feature-level alignment between the two modalities (Eq. 2). This loss replaces the MSE loss from Mohapatra et al. (2026).

Cosine Similarity Loss To encourage angular alignment in the embedding space, we keep the original cosine similarity loss from the original paper, and minimize the cosine distance between corresponding speech and text representations (Eq. 3).

Softmax Contrastive Loss To enforce separation from non target tokens, we use softmax function to maximize similarity to the positive class embeddings and minimize similarity to all other embeddings. Let $E \in \mathbb{R}^{V\times d}$ denotes the LLM's embedding layer, where V is the vocabulary size. For each speech representation $z_s^{(t)}$, we compute cosine similarity scores with all embeddings and use these as logits, with $s_t \in \mathbb{R}^V$ and $y_t \in \mathbb{R}^V$ denoting respectively the logits and the one-hot vector corresponding to the target token at position t (Eq. 4). The negative log likelihood encourages SpeechMapper output embeddings to be close to its correct token embedding while remaining well separated from all other tokens in the vocabulary.

$$\mathcal{L}_{\text{L1}} = \frac{1}{Td}\sum_{t=1}^{T}\left\| z_s^{(t)} - z_t^{(t)} \right\| \tag{2}$$

$$\mathcal{L}_{\cos} = \frac{1}{T}\sum_{t=1}^{T}\left(1 - \frac{z_s^{(t)}\cdot z_t^{(t)}}{\|z_s^{(t)}\|_2 \|z_t^{(t)}\|_2}\right) \tag{3}$$

$$\mathcal{L}_{\text{softmax}} = -\frac{1}{T}\sum_{t=1}^{T} y_t^{\top}\log(\text{softmax}(s_t)) \tag{4}$$

Connectionist Temporal Classification (CTC) We found that adding CTC (Graves et al., 2006) loss ($\mathcal{L}_{\text{ctc}}$) at an intermediate layer helps in stabilizing SpeechMapper's training and output embeddings quality.

Final Objective The overall training objective is a weighted sum of the losses:

$$\mathcal{L} = \mathcal{L}_{\text{L1}} + \mathcal{L}_{\text{cos}} + 0.1 * \mathcal{L}_{\text{softmax}} + \mathcal{L}_{\text{ctc}} \quad (5)$$

4 Training Pipeline

Our training pipeline is illustrated in Figure 1. We first train two components in parallel: (A) a SpeechMapper speech-to-embedding projector using ASR data, and (B) text LoRA weights using MT and QA data. These components are subsequently reloaded and jointly adapted on a mixture of speech and text tasks (C). In this section, we describe the key components of this training pipeline.

Foundation Models For speech, we leverage `SeamlessM4T-v2-large` model, extracting speech representations for all our audio data from its 24th speech encoder layer (i.e. the last layer). Prior to training, we average every two consecutive frame vectors, reducing significantly the sequence length. We highlight a minor difference from last year's setup: we average two consecutive frame vectors instead of three. Prior experiments with `SpeechMapper` motivated this choice, indicating that a lighter compression before the projector's CNN layers yields better performance. All our models are built on top of a frozen `Qwen3-4B-Instruct-2507`.

SpeechMapper Settings Our `SpeechMapper` follows the architecture described in Mohapatra et al. (2026), comprising two consecutive blocks, each consisting of a CNN, N self-attention layers, and a feed-forward projection to a higher dimensional space. To accommodate a CTC head at the end of the first block, we modify the convolutional strides from the original setting of 2 in both blocks to 1 and 4 for the first and second blocks, respectively. We train the model with $N = 6$, using initial, intermediate, and output dimensionalities of 1024, 2048, and 2560, respectively.

LoRA Adapters LoRA adaptation (Hu et al., 2022) is applied to both the self-attention (Q/K values, output projection) and feed-forward modules, and across all LLM layers. We use $rank = 8, \alpha = 16$. We do not use dropout.

Data Sampling Strategy For training the models in Figure 1 (B) and (C), we define an epoch as X steps across the dataset, where $X = \frac{|\text{speech\_examples}|}{\text{batch\_size}}$. To construct the data for each epoch, batches are sampled by first applying the predefined task-level sampling ratios, followed by sampling according to domain-level splits within each task. In the multimodal training setting (speech and text tasks mixed), we consider speech as our *main* modality, using it for defining epoch size and task sampling ratio. Whenever a sampled speech task and language pair has a textual counterpart (e.g., ST corresponds to MT; SQA to QA), we also sample a batch from the corresponding textual task. In practice, this results in alternating batches, such that a sampled ST en-de batch is followed by an MT en-de batch. We find that incorporating textual data in this manner consistently improves the final model performance.

5 Experimental Settings

Codebase We train our models using an internal fork of `torchtune` (torchtune maintainers and contributors, 2024), which allows us to process interleaved representations of text and high-dimensional vectors within the user turn during instruction tuning. We also implement our updated version of `SpeechMapper` on this codebase. For multimodal training, the high-dimensional vectors pass through `SpeechMapper`, while the text prefix and suffix user prompts are processed by the LLM embedding layer. The obtained speech and text embeddings are both concatenated and fed into the first layer of the LLM which is trained on the masked input with standard cross-entropy loss. Different learning rate schedulers and optimizers are employed for `SpeechMapper` and the LoRA weights, allowing for more controlled and effective training of these distinct model components.

Inference Settings We perform inference using `torchtune`, with a batch size of 1, greedy decoding, and with the maximum number of new tokens limited to 100. This decoding strategy was consistently applied across all experimental settings.

Evaluation Metrics We evaluate our models on speech (ASR, ST, SQA) and text (MT, QA) tasks when relevant. For ASR, we score word error rate (WER) using HuggingFace evaluate library with default settings and MMS normalization (Pratap et al., 2024). For ST/MT we score COMET (Rei et al., 2022).[6] For SQA/QA, we use LLM-as-a-judge evaluation scripts from the

[6] `Unbabel/wmt22-comet-da`

bergen library[7] (Rau et al., 2024). We use their "yes/no" quality assessment evaluation format including the reference question and answer, and generated output. We report average accuracy across three LLMs: `EuroLLM-9B-Instruct` (Martins et al., 2024), `Gemma3-12B-Instruct`, `Gemma3-27B-Instruct` (Team et al., 2025).

Baselines We compare our results with both backbones we use for training. We evaluate MT using the reference transcripts and `Qwen3-4B-Instruct-2507` in zero-shot settings, and we evaluate `SeamlessM4T-v2-large` for ASR, ST and MT. Additionally, we present results for last year's best short track system, referring to it as **BEST-IWSLT25-IF** (Lee et al., 2025).

6 Experiments

We now present our results for ASR, ST, and SQA. Section 6.1 introduces the models used in our experiments, followed by results and discussion in Section 6.2.

6.1 Our Models

`SpeechMapper` We train `SpeechMapper` on ASR data from CoVoST2, EuroParlST, GigaST, and LibriSQA, without applying any up-sampling. Training is performed for 500k steps using AdamW with a learning rate of $1e-4$ and 50k warm-up steps. We employ dynamic batching with gradient accumulation set to 2. As `Qwen3-4B-Instruct-2507` does not provide a dedicated padding token, we use the reserved (untrained) token 151664 for padding. The model is trained on $4\times$A100-80GB GPUs for approximately two days.

B. Text-only LoRA (MT/QA) The LoRA weights (B) are trained on all available real and synthetic data from Appendix Table 5, with task-level sampling ratios of 0.6 and 0.4 for MT and SQA, respectively. For MT, language sampling ratios are set to 0.4/0.3/0.3 for de/it/zh, while for QA they are 0.2/0.3/0.3/0.2 for en/de/it/zh. We train for 30k steps using AdamW with learning rate of $3e-4$, weight decay of 0.1, and 100 warm-up steps. Batch size of 16, and gradient accumulation of 8 is used. This model trains for approximately 4 days in a single A100-80GB.

C. Multimodal (A + B) We restart training by reloading both modules described above. We explore various combinations of the available datasets and obtain the best performance with the configuration detailed in Appendix Table 6. We use task-level sampling ratios of 0.3/0.4/0.3 for ASR, ST, and SQA, respectively. For ST/MT, language sampling ratios are set to 0.4/0.4/0.2 for de/it/zh. For SQA/QA, we use uniform language sampling over the four languages, assigning a ratio of 0.2 each for the valid set and 0.05 each for the invalid set. We use learning rate of $5e-5$ for the `SpeechMapper`, and of $1e-5$ for the LoRA weights (setup 1) or $5e-5$ for the LoRA weights (setup 2). We use a batch size of 8, and gradient accumulation of 6. This model trains for 3K steps, approximately 2 hours in a single A100-80GB.

6.2 Results and Discussion

Table 1 present results for ASR, ST/MT and SQA/QA for all of our models and baselines over the available test splits. For ASR, we evaluate using EuroParl, CoVoST and Librispeech clean/other. For ST/MT, we evaluate on EuroParl and CoVoST, excluding GigaST which we find to be particularly noisy. SQA/QA results cover only English, and are an average over scores obtained for the different LLM-as-judge models. Table 2 present MCIF scores (ASR, ST and multilingual SQA) obtained for the speech models. Below we discuss our main findings.

MT Results Looking at the MT results for text-only models (top portion of Table 1), we observe that this year's backbone performs substantially worse than `SeamlessM4T-v2-large` and `Llama-3.1-8B-Instruct`. This is expected, as the backbone has roughly half the parameters of last year's LLM. Results are particularly poor for Italian, motivating us to up-sample this language during fine-tuning. Finally, we also present results for our best LoRA setting, which considerably improves performance across both MT and QA tasks.

Projector-only Results We present results for our best `SpeechMapper` configuration trained to produce `Qwen3-4B-Instruct-2507` embeddings. Overall, this setting proves challenging: the LLM appears particularly sensitive to noise in the input embeddings, especially compared to larger backbones tested in Mohapatra et al. (2026). Our best `SpeechMapper` setup performs correctly in in-domain settings (Table 1), but we find it to underperform considerably in the MCIF dataset (Table 2), where it handles poorly named entities. In addition, zero-shot usage of `Qwen3-4B-Instruct-2507` is

[7] https://github.com/naver/bergen

Model	ASR (WER)	ST/MT (COMET)			SQA/QA (LLM-as-judge)	
		en-de	en-it	en-zh	Part I	Part II
Text-only Models (zero-shot LLMs, SFM and LoRA)						
`SeamlessM4T-v2-large` (MT)	-	81.4	86.7	80.8	-	-
`Llama-3.1-8B-Instruct`	-	**81.9**	84.1	77.0	86.6	70.8
`Qwen3-4B-Instruct-2507`	-	71.0	67.7	74.3	89.1	70.2
`Qwen3-4B-Instruct-2507` + LoRA (B)	-	80.7	**86.9**	**84.7**	**89.9**	**82.6**
SFM and Projector-only Model						
`SeamlessM4T-v2-large` (ASR/ST)	**5.9**	**78.3**	76.9	78.0	-	-
`SpeechMapper` (A)	14.2	73.5	**80.1**	**79.7**	84.4	72.1
Multimodally Trained Models						
BEST-IWSLT25-IF	**7.3**	**77.3**	84.2	80.2	82.0	63.0
`SpeechMapper` + LoRA (C) setup 1	8.2	75.1	83.2	80.6	86.2	**82.5**
`SpeechMapper` + LoRA (C) setup 2	7.4	76.3	**84.4**	**81.3**	**87.9**	80.2

Table 1: Results for ASR (LibriSpeech clean/other, EuroParlST, CoVoST2), ST/MT (CoVoST2, EuroParlST), and SQA/QA (LibriSQA PartI and PartII). ASR and ST/MT scores are reported as weighted averages across datasets, proportional to the number of samples, while SQA/QA results are averaged across judges and reported as accuracy.

Model	ASR	ST	SQA
BEST-IWSLT25-IF	12.6	0.743	0.417
`SpeechMapper` (A)	32.2	0.711	0.225
`SpeechMapper` + LoRA (C) setup 1	12.0	0.772	**0.428**
`SpeechMapper` + LoRA (C) setup 2	**10.5**	**0.781**	0.400

Table 2: Averaged MCIF scores for all speech models.

difficult, as the model reorganizes and/or rephrases the transcription text in many instances, or responds in an overly verbose manner, both cases negatively affect evaluation scores. Qualitative examples illustrating these behaviors are provided in Appendix Table 8.

Multimodal Models We experiment with a wide range of dataset combinations and sampling strategies, and report results only for our best-performing multimodal models (bottom portion of Table 1 and Table 2). These models are able to match the performance of last year's system, despite relying on a smaller LLM backbone. Overall, we find that improving performance jointly on ASR and SQA is more challenging this year, as gains in one task often come at the expense of the other. This trade-off is illustrated by Setups 1 and 2: while Setup 1 achieves better results on LibriSQA Part II (MCQ) and MCIF's SQA, Setup 2 yields stronger performance on ASR and ST.

7 Submitted Model

We select our model based on its MCIF performance. Overall, we were unable to produce a single run that achieves the best score across all tasks, as improvements in one task tend to degrade performance in others. Therefore, we prioritize SQA as our primary task and use it as the key criterion for model selection. Based on Table 2, we submit setup 1 as our main submission. We also resubmit BEST-IWSLT-IF as a contrastive, unconstrained system. Our competition results are discussed in Appendix Section A.3.

8 Conclusion

In this paper, we presented NLE's submission to the instruction-following speech processing short track at IWSLT 2026 under the constrained setting. We developed multimodal models capable of jointly performing ASR, ST, and SQA from English speech into Chinese, German, and Italian. Building on our previous pipeline (Lee et al., 2025), we replaced the transformer-based speech projector with an updated `SpeechMapper` projector (Mohapatra et al., 2026). We also introduced a synthetic dataset of scientific talks, *fakACL*, to mitigate domain mismatch between training and evaluation. Despite relying on a substantially smaller LLM backbone, our final system outperforms last year's best submission in the short track.

References

David Ifeoluwa Adelani, Antonios Anastasopoulos, Victor Agostinelli, Luisa Bentivogli, Ondřej Bojar, Sebastien Bratières, Marine Carpuat, Roldano Cattoni, Mauro Cettolo, Lizhong Chen, Marcello Federico, Marco Gaido, Mahendra Gupta, HyoJung Han, Ali Hatami, David Javorský, Yejin Jeon, Marek Kasztelnik, Danni Liu, and 26 others. 2026. Speech translation and metrics in 2026: Findings of the iwslt campaign. In *Proceedings of the 23rd International Conference on Spoken Language Translation (IWSLT 2026)*, San Diego, California, US. Association for Computational Linguistics.

Kshitij Ambilduke, Ben Peters, Sonal Sannigrahi, Anil Keshwani, Tsz Kin Lam, Bruno Martins, Marcely Zanon Boito, and André FT Martins. 2025. From tower to spire: Adding the speech modality to a text-only llm. *arXiv preprint arXiv:2503.10620*.

Loïc Barrault, Yu-An Chung, Mariano Cora Meglioli, David Dale, Ning Dong, Paul-Ambroise Duquenne, Hady Elsahar, Hongyu Gong, Kevin Heffernan, John Hoffman, and 1 others. 2023. Seamlessm4t: Massively multilingual & multimodal machine translation. *arXiv preprint arXiv:2308.11596*.

Alexandre Défossez, Laurent Mazaré, Manu Orsini, Amélie Royer, Patrick Pérez, Hervé Jégou, Edouard Grave, and Neil Zeghidour. 2024. Moshi: a speech-text foundation model for real-time dialogue. *arXiv preprint arXiv:2410.00037*.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, pages 369–376.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Shujie Hu, Long Zhou, Shujie Liu, Sanyuan Chen, Lingwei Meng, Hongkun Hao, Jing Pan, Xunying Liu, Jinyu Li, Sunit Sivasankaran, and 1 others. 2024. Wavllm: Towards robust and adaptive speech large language model. *arXiv preprint arXiv:2404.00656*.

Rongjie Huang, Mingze Li, Dongchao Yang, Jiatong Shi, Xuankai Chang, Zhenhui Ye, Yuning Wu, Zhiqing Hong, Jiawei Huang, Jinglin Liu, and 1 others. 2024. Audiogpt: Understanding and generating speech, music, sound, and talking head. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 23802–23804.

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 *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233.

Beomseok Lee, Marcely Zanon Boito, Laurent Besacier, and Ioan Calapodescu. 2025. NAVER LABS Europe submission to the instruction-following track. In *Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025)*, pages 186–200, Vienna, Austria (in-person and online). Association for Computational Linguistics.

Pedro Henrique Martins, Patrick Fernandes, João Alves, Nuno M. Guerreiro, Ricardo Rei, Duarte M. Alves, José Pombal, Amin Farajian, Manuel Faysse, Mateusz Klimaszewski, Pierre Colombo, Barry Haddow, José G. C. de Souza, Alexandra Birch, and André F. T. Martins. 2024. Eurollm: Multilingual language models for europe. *Preprint*, arXiv:2409.16235.

Biswesh Mohapatra, Marcely Zanon Boito, and Ioan Calapodescu. 2026. Speechmapper: Speech-to-text embedding projector for llms. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*.

Tu Anh Nguyen, Benjamin Muller, Bokai Yu, Marta R Costa-Jussa, Maha Elbayad, Sravya Popuri, Christophe Ropers, Paul-Ambroise Duquenne, Robin Algayres, Ruslan Mavlyutov, and 1 others. 2025. Spirit-lm: Interleaved spoken and written language model. *Transactions of the Association for Computational Linguistics*, 13:30–52.

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)*, pages 5206–5210.

Sara Papi, Maike Züfle, Marco Gaido, Beatrice Savoldi, Danni Liu, Ioannis Douros, Luisa Bentivogli, and Jan Niehues. 2026. MCIF: Multimodal crosslingual instruction-following benchmark from scientific talks. In *The Fourteenth International Conference on Learning Representations*.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, and 1 others. 2024. Scaling speech technology to 1,000+ languages. *Journal of Machine Learning Research*, 25(97):1–52.

David Rau, Hervé Déjean, Nadezhda Chirkova, Thibault Formal, Shuai Wang, Stéphane Clinchant, and Vassilina Nikoulina. 2024. BERGEN: A benchmarking library for retrieval-augmented generation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 7640–7663, Miami, Florida, USA. Association for Computational Linguistics.

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.

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, and 1 others. 2023. Audiopalm: A large language model that can speak and listen. *arXiv preprint arXiv:2306.12925*.

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)*, pages 62–78, Toronto, Canada (in-person and online). Association for Computational Linguistics.

Changli Tang, Wenyi Yu, Guangzhi Sun, Xianzhao Chen, Tian Tan, Wei Li, Lu Lu, Zejun Ma, and Chao Zhang. 2023. Salmonn: Towards generic hearing abilities for large language models. *arXiv preprint arXiv:2310.13289*.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Qwen Team. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

torchtune maintainers and contributors. 2024. torchtune: Pytorch's finetuning library.

Changhan Wang, Anne Wu, and Juan Pino. 2020. Covost 2: A massively multilingual speech-to-text translation corpus. *Preprint*, arXiv:2007.10310.

Jin Xu, Zhifang Guo, Hangrui Hu, Yunfei Chu, Xiong Wang, Jinzheng He, Yuxuan Wang, Xian Shi, Ting He, Xinfa Zhu, and 1 others. 2025. Qwen3-omni technical report. *arXiv preprint arXiv:2509.17765*.

Rong Ye, Chengqi Zhao, Tom Ko, Chutong Meng, Tao Wang, Mingxuan Wang, and Jun Cao. 2023. GigaST: A 10,000-hour Pseudo Speech Translation Corpus. In *Interspeech 2023*, pages 2168–2172.

Zihan Zhao, Yiyang Jiang, Heyang Liu, Yu Wang, and Yanfeng Wang. 2025. Librisqa: A novel dataset and framework for spoken question answering with large language models. *IEEE Transactions on Artificial Intelligence*, 6(11):2884–2895.

A Appendix

A.1 Qwen-based SQA Invalid Split

We aim to generate high-quality invalid questions for training our speech LLM. To ensure this quality, the generated questions must be unrelated to the speech content. Since LibriSQA is an audiobook-based dataset, we therefore focus on creating questions grounded in broad conversational topics.

We first prompted `Qwen3-4B-Instruct-2507` to produce a large set of conversation topics using:

> "*List about 1000 conversation topics, without numbering or adding any comment of explanation.*"

After post-processing the output, we retained 413 topics. Next, for each one, we prompted the LLM with:

> "*Generate 30 words related to **TOPIC**. Produce a comma-separated list of words, without any explanation.*"

Then, using the resulting list, we finally prompted the model with:

> "*Ask 25 different questions about **WORD**. Do not repeat the question, include no comments, and output only the questions, one per line.*"

The output was automatically filtered to remove sentences that did not end with a question mark, resulting in 62,040 English questions. Some examples are shown below:

- *What role do animals play in traditional rites?*
- *How does still air interact with thermal radiation?*
- *Can a revoked loan be reapproved?*
- *What is the role of the saw blade in determining cut quality?*
- *How do social media platforms influence the perception of friendship and connection?*
- *What is the significance of putrefaction in forensic science?*

A.2 fakACL Creation

Our goal in creating *fakACL* is to mimic ACL60-60 (Salesky et al., 2023) and MCIF. These datasets consist of oral presentations that typically begin with the authors introducing themselves and stating the title of the paper, followed by an overview of the main findings.

Our dataset creation process consists of three stages: (1) script creation, (2) segmentation and speech synthesis, and (3) QA generation. We now describe these stages.

A.2.1 Script Creation

The goal of this stage of the dataset creation process is to generate high-quality presentation scripts on NLP-related topics. We started the process by first prompting `Qwen3-4B-Instruct-2507` to:

> "*List 60 sub-fields related to ACL/NLP conferences, one subfield par line.*"

This resulted in a list of 56 ACL conference topics, with entries such as *Healthcare NLP*, *Legal document analysis* and *Education*. We do not check for duplicated conference topics.

Next, we generated a collection of possible paper titles for each one of the topics. We prompted the LLM with:

> "*List 40 paper titles from ACL conferences related to: **SUB-FIELD**. Output only the paper titles, one per line. Nothing else.*"

The result was a collection of 2,527 paper titles. Examples of generated paper titles were "*Phonological and Morphological Development in Multilingual Children*" and "*Emotion-Driven Text-to-Speech with Multimodal Inputs*".

Finally, we prompted the LLM with:

> "*Write a 12 sentences oral script for presenting the ACL paper entitled: **TITLE**.*"

After manual processing, this resulted in 2,497 scripts. An example of a generated script is given at Table 3.

A.2.2 Segmentation and Speech Synthesis

After the scripts are created, the next step consists of segmenting the content into blocks to be synthesized by `SeamlessM4T-v2-large`. This stage includes preprocessing steps such as number and symbol normalization (e.g., "30%" becomes "thirty percent"), as well as text normalization.

Synthetic Script from *fakACL*
Good morning, everyone. Today, I'm excited to present our paper titled Cross-Modal Retrieval using Cross-Modal Semantic Matching and Contextual Awareness. In traditional retrieval systems, matching content across different modalities like text and images has been challenging due to inherent semantic gaps. Our approach tackles this by introducing a novel framework that combines cross-modal semantic matching with contextual awareness. We leverage deep neural networks to learn rich, shared semantic representations between modalities, ensuring that queries from one modality can accurately retrieve relevant content from another. Crucially, we incorporate contextual awareness to understand the environment and user intent such as time, location, or user preferences before generating retrieval results. This allows the system to go beyond literal matches and deliver more relevant, human-like responses. We validate our method on benchmark datasets including CORD-19 and MS-COCO, demonstrating significant improvements over baseline models in precision and recall. Our experiments show that contextual signals improve retrieval performance by up to 18% in real-world scenarios. Additionally, we conduct ablation studies that confirm the importance of both semantic matching and contextual components. The framework is scalable and can be adapted to various applications from medical image search to e-commerce product recommendations. We believe this work bridges the gap between technical accuracy and real-world usability in cross-modal systems. Finally, we envision future work exploring dynamic context modeling and multimodal feedback loops to further refine retrieval. Thank you for your attention. I'd be happy to take any questions.
Synthetic Questions and Answers from *fakACL*
Context from script: "Crucially, we incorporate contextual awareness to understand the environment and user intent such as time, location, or user preferences before generating retrieval results." **Question 1:** "What does the system use to understand user intent?" **Answer 1:** "Contextual awareness including time, location, and user preferences." **Question 2:** "How does the system enhance retrieval results?" **Answer 2:** "By aligning retrieval with time, location, and user preferences."

Table 3: Example of a synthetic presentation script and QA produced by `Qwen3-4B-Instruct-2507`.

We use the Python library `spaCy` to split the scripts into sentences, normalize the text, and discard sentences containing fewer than 10 characters or fewer than 6 words. This process results in 21,400 sentences. We then synthesize the sentences using `SeamlessM4T-v2-large` TTS with random speakers. The average utterance length is 8.6 seconds.

A.2.3 QA Generation

Lastly, after obtaining the segmented and normalized scripts, our next goal was to generate valid question–answer pairs. For this stage, we discarded the first three and the last two sentences of each script, as these tended to be overly generic and usually corresponded to greetings or acknowledgments.

For each text segment, we prompted `Qwen3-4B-Instruct-2507` with:

> "*Generate 2 pairs of question and very short answer. Generate the question in one line, the answer in the next line, prefix the question by "Q:" and the answer by "A:". Generate strictly and solely based on the content:* ***CONTENT***."

We automatically filtered the outputs based on the question–answer prefixes and subsequently performed manual verification to remove clear hallucinations. This process yielded a final set of 38,968 questions.

A.3 IWLST 2026 IF Short Results

Table 4 presents the official results for our primary (Setup 1 from Table 1) and contrastive (BEST-IWSLT25-IF) submissions. Overall, the results confirm that we were able to build a stronger system than last year's submission, despite relying on a smaller and weaker LLM backbone.

Our primary submission outperforms BEST-IWSLT25-IF across all languages for ST, while lagging by only 0.2% WER on ASR. For QA, this year's model again improves performance across all languages except Italian, for which we found the backbone to be particularly weak.

Finally, the inclusion of MCQ training allowed our model to perform the surprise task of Quality Estimation (QE-accuracy) while respecting the proposed zero-shot prompt (QE-format-accuracy). In contrast, last year's solution struggled to follow the required format and often generated answers instead of selecting options, making it incapable of correctly answering cases where the correct option did not correspond to the language of the prompt.

Model	Lang	TRANS-COMET	QA-BERTScore	QE-accuracy	QE-format-accuracy	ASR-WER
NLE_IWSLT26_IF_SHORT_CONSTRAINED_PRIMARY	en	–	**0.531**	–	–	0.136
NLE_IWSLT26_IF_SHORT_CONSTRAINED_PRIMARY	it	**0.763**	0.456	–	–	–
NLE_IWSLT26_IF_SHORT_CONSTRAINED_PRIMARY	de	**0.765**	**0.470**	**0.786**	**0.997**	–
NLE_IWSLT26_IF_SHORT_CONSTRAINED_PRIMARY	zh	**0.794**	**0.487**	**0.894**	**1.000**	–
NLE_IWSLT26_IF_SHORT_UNCONSTRAINED_CONSTRASTIVE	en	–	0.501	–	–	**0.134**
NLE_IWSLT26_IF_SHORT_UNCONSTRAINED_CONSTRASTIVE	it	0.733	**0.514**	–	–	–
NLE_IWSLT26_IF_SHORT_UNCONSTRAINED_CONSTRASTIVE	de	0.749	0.462	0.333	0.005	–
NLE_IWSLT26_IF_SHORT_UNCONSTRAINED_CONSTRASTIVE	zh	0.755	0.466	0.500	0.014	–

Table 4: Official results for our systems submitted to IWSLT 2026 IF short track.

Dataset	Task	Language	# Samples
CoVoST2	**ASR**	en	289,413
	ST/MT	en-de	289,413
		en-zh	289,413
EuroParlST	**ASR**	en	35,372
	ST/MT	en-de	32,628
		en-it	29,552
GigaST	**ASR**	en	7,645,641
	ST/MT	en-de	7,645,641
		en-it*	3,004,804
		en-zh	7,645,641
LibriSQA	**ASR**	en	104,014
	MT	en-de*	36,707
		en-it*	75,378
		en-zh*	22,211
	SQA/QA (valid)	en	208,038
		en-de*	59,274
		en-it*	127,295
		en-zh*	29,385
	SQA/QA (invalid)	en	169,068
		en-de*	79,578
		en-it*	123,632
		en-zh*	58,092
fakACL	**ASR**	en†	21,400
	ST/MT	en†-de*	9,722
		en†-it*	14,999
		en†-zh*	4,659
	SQA/QA (valid)	en†	38,968
		en†-de*	8,763
		en†-it*	22,290
		en†-zh*	5,341
	SQA/QA (invalid)	en†	38,968
		en†-de*	21,192
		en†-it*	31,333
		en†-zh*	19,549

Table 5: Training sets statistics by task. For ST/MT sets, target side is duplicated. * denotes synthetic text obtained via `SeamlessM4T-v2-large` MT; † indicates splits generated with `SeamlessM4T-v2-large` TTS. For fakACL, all English textual content is produced via `Qwen3-4B-Instruct-2507`prompting.

	CoVoST			EuroParlST			GigaST			LibriSQA			fakeACL			
	ASR	**ST**	**MT**	**ASR**	**ST**	**MT**	**ASR**	**ST**	**MT**	**ASR**	**MT**	**SQA/QA**	**ASR**	**ST**	**MT**	**SQA/QA**
A `SpeechMapper` (ASR)	✓	✗	✗	✓	✗	✗	✓	✗	✗	✓	✗	✗	✗	✗	✗	✗
B Text-only LoRA (MT/QA)	✗	✗	✓	✗	✗	✓	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓
C Multimodal model (ASR/ST/MT/SQA/QA)	✓	✓	✓	✓	✓	✓	✗	✗	✓	✗	✓	✓	✗	✗	✓	✓

Table 6: List of datasets and splits used for training our presented models. Statistics are presented in Table 5.

	User Prompt
Speech Prefix	Content: <speech>[**SPEECH EMBEDDINGS**]</speech>\n
Text Prefix	Content: <text>[**SPEECH TRANSCRIPTION**]</text>\n
ASR	Question: Can you transcribe the Speech content into English text?\n
ST/MT (de)	Question: Können Sie den Inhalt der Rede in den deutschen Text übersetzen?\n
ST/MT (it)	Question: Puoi tradurre il contenuto del discorso in testo italiano?\n
ST/MT (zh)	Question: 你能把演讲内容翻译成中文吗?\n
SQA/QA (Free format)	Question: [**QUESTION**]\n
SQA/QA (MCQ)	Question: [**QUESTION+OPTIONS**]\n
Surprise Task	Question: [**SURPRISE INSTRUCTION**]\n
Suffix	\nYour answer:
Suffix (zero-shot ASR)	Do not add anything else to your answer. \nYour answer:
Suffix (zero-shot ST de)	Antworten Sie nur mit der Übersetzung. Fügen Sie Ihrer Antwort nichts weiter hinzu.\nYour answer:
Suffix (zero-shot ST it)	Rispondi solo con la traduzione. Non aggiungere altro alla tua risposta.\nYour answer:
Suffix (zero-shot ST zh)	仅回复翻译。请勿在答案中添加任何其他内容。\nYour answer:

Table 7: The user turn prompt template used for training and/or evaluating models. For speech tasks, the user prompt is given by **Speech Prefix+Task+Suffix**, for textual tasks, the user prompt is given by **Text Prefix+Task+Suffix**. **ST/MT instructions** were obtained by translating the instruction "*Can you translate the Speech content into **[German/Italian/Chinese]** text?*" and "*Answer only with the translation. Do not add anything else to your answer.*" to corresponding target languages using `SeamlessM4T-v2-large`. **SQA/QA instructions** are either *free format* (direct answer) or *multiple-choice questions* (MCQ). In the case of MCQ, options are included in the same line of the question, and both are translated to the target language in case of multilingual SQA. **Surprise task instructions** are included after replacing the "\n" by spaces.

Reference	Generation
This is a joint work with John Gauthier, Aaron Mueller, Kanishka Misra, Karen Fences, Roger Levy, and Adina Williams.	It's a joint work with John Gautier, Aaron Mueller, Kanishka Misra, Karen Fentos, Roger Le, and Andina Will. Will. Will.
Mr President, what worries me most is the suggestion that the European Investment Bank has a significant role to play in solving the financial crisis – that it can somehow act as the Europe-wide body to provide stimulus where Member States have singularly failed to coordinate their own stimuli.	Mr. President, what worries me most is the suggestion that the European Investment Bank has a significant role to play in solving the financial crisis. It is claimed that the bank could somehow act as the Europe-wide body to provide stimulus where member states have singularly failed to coordinate their own stimuli.
From the experience I have as a member of the national parliament and talking with many people for a long time, I would really be in favour of smart sanctions – targeted, for example, at specific members of the Revolutionary Guard, putting them on a veto list for visits, or at other specific persons.	From the experience I have as a member of the national parliament and having discussed this with many people over a long time, I would really support smart sanctions targeting, for example, specific members of the Revolutionary Guard, in order to target or punish specific individuals.
C. Valor and fortune	The speech states: *"each party hurrying to a battle where value and and fortune chiefly the the success."* \n\nThis indicates that the primary factors determining battle success were **valor** (bravery) and **fortune** (luck).\n\nLooking at the options:\n\n- A. Experience and knowledge → not mentioned \n- B. Strategic planning → not mentioned \n- C. Valor and fortune → directly supported by the speech \n- D. Ignorance and impetuous conduct → mentioned as something historians may have overlooked, not as a factor of success \n\n**Correct answer: C. Valor and fortune**

Table 8: Some examples of generations for `Qwen3-4B-Instruct-2507` using `SpeechMapper` (A) embeddings. In the top, an example of hallucination related to unknown named entities at MCIF; in the middle two examples of ASR rephrasing with EuroParlST; and in the bottom an example of extremely verbose SQA output for LibriSQA.