SYSTRAN @ IWSLT 2025 Low-resource track

Marko Avila and Josep Crego
SYSTRAN by ChapsVision
5 rue Feydeau, 75002 Paris (France)

Abstract

SYSTRAN submitted systems for one language pair in the 2025 Low-Resource Language Track. Our main contribution lies in the tight coupling and light fine-tuning of an ASR encoder (Whisper) with a neural machine translation decoder (NLLB), forming an efficient speech translation pipeline. We present the modeling strategies and optimizations implemented to build a system that, unlike large-scale end-to-end models, performs effectively under constraints of limited training data and computational resources. This approach enables the development of high-quality speech translation in low-resource settings, while ensuring both efficiency and scalability. We also conduct a comparative analysis of our proposed system against various paradigms, including a cascaded Whisper+NLLB setup and direct end-to-end fine-tuning of Whisper.

1 Introduction

The goal of the IWSLT'2025 low-resource shared task is to benchmark and promote speech translation technology for a diverse range of dialects and low-resource languages. While significant research progress has been demonstrated recently on popular datasets, many of the world's dialects and low-resource languages lack the parallel data at scale needed for standard supervised learning. Thus, this share task requires creative approaches in leveraging disparate resources. The low-resource shared task will involve two tracks:

- Track 1: A "traditional" speech-to-text translation track focusing on XX typologically diverse language-pairs.
- Track 2: A data track, inviting participants to provide open-sourced speech translation datasets for under-resourced languages.

SYSTRAN participates exclusively in the "traditional" Tunisian Arabic-to-English speech translation track. Our system employs a tightly coupled architecture wherein the automatic speech recognition (ASR) encoder directly interfaces with the neural machine translation (NMT) encoder-decoder module. This end-to-end pipeline has demonstrated robust performance in prior evaluations under low-resource conditions. The primary objective is to build a high-performance speech-to-text translation (S2TT) system optimized for constrained computational environments and limited annotated data, while effectively leveraging the representational power of large-scale pretrained models.

In Section 2, we describe the corpora used in this study, as well as the pre-processing steps applied to improve their relevance and quality for the target tasks (see Section 3). Section 4 introduces the proposed system, which combines a speech encoder with neural machine translation components. Section 5 presents the experimental setup and reports the results obtained. Finally, Section 6 summarizes the main findings and concludes the paper.

2 Dataset Description

This work is conducted as part of a shared task aimed at advancing the state of the art in ASR and speech S2TT for low-resource dialects, with a particular focus on Tunisian Arabic. To ensure comparability and fairness, all experiments are conducted under the *constrained condition*, using exclusively the Tunisian-English resources provided by the Linguistic Data Consortium (LDC) for this challenge.

2.1 Corpus Overview

The dataset comprises manually transcribed and translated audio resources in two language varieties: Tunisian Arabic (TA) and Modern Standard Arabic (MSA). Although MSA content originates from broadcast news (BN), TA is represented through conversational telephone speech (CTS), offering rich linguistic variability. English transla-

Proceedings of the 22nd International Conference on Spoken Language Translation (IWSLT 2025), pages 324–332
July 31 - August 1, 2025 ©2025 Association for Computational Linguistics

tions are available for all MSA transcripts and for a large portion of TA segments, enabling the training of end-to-end speech translation models.

2.2 Audio Data

MSA Broadcast News (BN): This portion includes two single-channel recordings totaling approximately 1 hour of audio. The recordings consist of multi-speaker news broadcasts and interviews, sampled at 16 kHz and stored as FLAC-compressed MS-WAV files with 16-bit PCM encoding.

TA Conversational Telephone Speech (CTS): The core of the dataset consists of 387 hours of two-channel telephone conversations, distributed across 2,198 dialogues (4,396 single-channel files). These were collected in Tunisia via an automated robot operator that interfaced directly with the regional public telephone network. Each call involves:

- Side A: A "claque" speaker, a recruited participant tasked with initiating conversations.
- Side B: A callee, selected naturally by the claque from their personal contacts.

Claques completed 8 to 15 distinct calls, each lasting 8–10 minutes, and were encouraged to dominate the discourse to ensure informative linguistic content. The TA audio files are encoded in A-law format at 8 kHz with NIST SPHERE headers.

2.3 Segmentation and Speaker Annotation

Broadcast News (BN). Manual segmentation and speaker turn identification were performed using the LDC-developed *XTrans* tool (Glenn et al., 2009). Speakers were identified by name when available; otherwise, anonymous labels indicating speaker and gender (e.g., *speaker1/male*) were used.

Conversational Telephone Speech (CTS). Segmentation followed a three-stage process: (i) automatic speech activity detection (SAD) with a minimum silence threshold of 0.5 seconds was applied to each single-channel audio file; (ii) segments longer than 15 seconds were re-segmented with a relaxed silence threshold of 0.3 seconds; (iii) final segment boundaries were manually verified and corrected in XTrans by expert annotators.

2.4 Transcription Protocol

BN transcripts were generated in Arabic script using XTrans. **CTS** segments, alternating between speakers A and B, were transcribed using a contextual navigation web interface. Transcripts were in Buckwalter transliteration, with some segments also featuring broad IPA transcriptions. A verification pass ensured alignment between orthographic and phonemic transcripts and enabled token-level annotation for MSA, foreign language, and uncertain items.

2.5 Transcription Statistics

Manual transcriptions were provided for both MSA and TA recordings. Table 4 summarizes the number of segments, duration of speech-only segments, and number of files per genre.

Table 1: Transcription statistics per genre.

Type	Segments	Hours	Files
MSA / BN	420	0.96	2
TA / CTS	398,064	323.73	4,396
Total	398,484	324.69	4,398

2.6 Translation Statistics

English translations are provided for the full set of MSA segments and a substantial subset of TA transcripts, supporting supervised ST model training. Table 2 reports the number of translated segments, duration of time-stamped speech, and corresponding files.

Table 2: Translation statistics per genre.

Type	Segments	Hours	Files
MSA / BN	420	0.96	2
TA / CTS	210,901	167.48	2,284
Total	211,321	168.44	2,286

In total, this release provides:

- **323.73 hours** of Tunisian Arabic CTS audio with manual transcriptions, suitable for ASR development.
- **167.48 hours** of translated Tunisian Arabic audio, enabling end-to-end ST modeling.
- **1 hour** of Modern Standard Arabic broadcast news audio, fully transcribed and translated.

This resource offers a rare and valuable foundation for research in dialectal ASR and ST, bridging the gap between underrepresented spoken varieties and high-resource translation targets.

3 Data Cleaning and Annotation

3.1 Token-Level Annotations and Markup

Arabic transcripts include token-level annotations to reflect linguistic variability:

- `M/` — Modern Standard Arabic
- `O/` — Foreign word
- `U/` — Uncertain token
- `UM/`, `UO/` — Combined uncertainty

BN transcripts use XML-style tags (`<non-MSA> ... </non-MSA>`) to flag non-MSA spans. These annotations were removed in our pre-processing pipeline to ensure clean input for downstream modeling.

3.2 Translation Annotations

English translations are aligned at the segment level. Annotation conventions include:

- `(())`: Uncertain words
- `%pw`: Partial word
- `#`: Untranslated foreign word
- `+`: Mispronunciation
- =: Typographic errors from the transcript
- *uh, um, eh, ah*: Filled pauses

All special symbols were removed in a pre-processing step.

3.3 Text Pre-processing and Token Filtering

To ensure consistency and reduce noise introduced by transcription and translation annotation artifacts, we applied language-specific filtering rules to clean both Arabic and English segments. These regular expressions were crafted based on the known annotation conventions of the dataset.

We defined the following regular expression used for Arabic transcripts:

$$re.compile(r'[OUM]+/*|\backslash u061F|\backslash ?|\backslash !|\backslash .')$$

This expression targets and removes annotation prefixes such as $O/$, $U/$, and $M/$, which denote foreign language tokens, uncertain words, and Modern Standard Arabic (MSA), respectively. It also eliminates punctuation marks including the Arabic question mark (Unicode $\backslash u061F$) as well as Western punctuation symbols (?, !, .), which are inconsistently used and not linguistically informative for model training.

For English translations, we defined the following filter:

$$re.compile(r'\backslash (|\backslash)|\backslash |\backslash +|\backslash =|\backslash ?|\backslash !|\backslash ;|\backslash .|\backslash ,|\backslash "|\backslash :')$$

This regular expression removes special characters and annotation markers such as $\#$ (foreign words), $+$ (mispronunciations), $=$ (typographical errors), and common punctuation symbols. These annotations were introduced during the manual translation process to capture spoken language phenomena but are not useful for token-level alignment or model training.

This pre-processing step allowed us to normalize the text, reduce vocabulary sparsity, and ensure cleaner input for downstream Automatic Speech Recognition (ASR) and Speech Translation (ST) tasks.

After filtering, preprocessing, and splitting the data according to the partitions provided by the organizers, we obtained the following subsets: training, development, and test.

Table 3: Transcription statistics per genre after filtering for train/dev/test.

Type	Segments	Hours
MSA / BN	410	0.94
TA / CTS	390,021/3833/4220	317.19/3.12/3.43
Total	390,431/3833/4220	318.13/3.12/3.43

Table 4: Translation statistics per genre after filtering for train/dev/test.

Type	Segments	Hours
MSA / BN	409	0.937
TA / CTS	202,504/3833/4204	160.81/3.12/3.42
Total	202,913/3833/4204	161.747/3.12/3.42

After filtering, the dataset comprises 318.13 hours of transcribed Tunisian Arabic audio for ASR training, with 3.12 and 3.42 hours for development and test, respectively. Additionally, it includes 161.75 hours of parallel audio-translation pairs for ST training, with the same dev/test splits.

3.4 Known Issues

- **Partial Call Coverage:** Some CTS calls are only partially annotated due to transcription kit omissions.

- **Stranded Diacritic Marks:** 158 instances of diacritic-prefixed tokens (e.g., a, i, o) persist in 134 files.
- **Empty Segments:** 714 CTS segments contain only a hyphen ("-"), signaling rejected or unusable segments.
- **Missing Translations:** 10 BN segments lack translations due to English speech in the source audio.

4 Coupling Whisper and NLLB

This work introduces a hybrid solution designed for parameter-efficient training in low-resource language scenarios inspired by the integration strategy presented in (Avila and Crego, 2025) , integrating speech representation features from a pre-trained speech model into a multilingual NMT system. Our approach integrates speech representation features from a pre-trained speech model encoder such as Whisper into a multilingual Neural Machine Translation system such as NLLB, enabling both ASR and S2TT capabilities.

4.1 Motivation and Context

The primary goal of this shared task is to benchmark and foster advancements in speech translation technologies for a wide spectrum of dialects and low-resource languages. In particular, this initiative focuses on improving automatic speech transcription and translation for the Tunisian dialect, a variety of Arabic that remains significantly underrepresented in existing resources.

Low-resource conditions such as those encountered with Tunisian Arabic pose substantial challenges for conventional speech translation pipelines, which typically rely on large-scale annotated corpora. In this context, pre-trained models like Whisper, despite their multilingual design, lack direct support for Tunisian. Conversely, the NLLB model provides explicit support for Tunisian text and English, enabling translation in both directions.

This complementary nature of Whisper and NLLB forms the foundation of our hybrid approach. By leveraging Whisper for robust audio feature extraction and NLLB for multilingual text translation, we bridge the gap between speech and text modalities. The integration of high-quality speech representations into a powerful text-based multilingual translation model allows us to address the limitations of current systems in low-resource environments.

4.2 Speech Representation via Whisper

In our hybrid approach, Whisper encoder is kept frozen and used to generate speech representations, which substitute the input word embedding of the NLLB network.

The speech representations X consist of the outputs after the K lower encoder layers:

$$\texttt{Whisper}_{ENC}^{K}(a) = X, \text{ with } X \in \mathcal{R}^{N \times M}$$

with a the audio signal, N the sequence length and M the embedding dimension.

Whisper[1] (Radford et al., 2023) is a speech recognition model tailored for multilingual recognition, translation, and language identification. Its Transformer-based architecture integrates multiple speech processing tasks into a single, unified model.

We use two variants of Whisper (**Medium** and **Large-v3**) to evaluate the impact of model scale on representation quality. Both models take 30-second segments of audio resampled at 16kHz and convert them into 80-channel log-magnitude Mel spectrograms. The Whisper encoder outputs are extracted from the final Transformer layer: the K=24th layer for **Medium** ($M = 1024$) and the K=32nd for **Large-v3** ($M = 1280$). The output is a fixed-length sequence of $N = 1500$ vectors.

To align Whisper outputs with the NLLB encoder input we employ a Reshape module consisting of:

- A **convolutional layer** with kernel size = 3 and stride = 1 is used to reduce the sequence length from 1500 to 100.
- A **linear projection layer** ($M \times 2048$) is applied to match the expected embedding dimension of the NLLB 3.3B encoder.

4.3 Neural Machine Translation with NLLB

We employ NLLB[2] (team et al., 2022), a multilingual NMT model developed by Meta AI, designed to support direct translation between more than 200 languages, including many low-resource and underrepresented languages. Based on a Transformer architecture, NLLB employs language-specific tokens and dense representations to handle diverse

[1] https://huggingface.co/openai/whisper-medium,https://huggingface.co/openai/whisper-large-v3

[2] https://huggingface.co/facebook/nllb-200-3.3B

linguistic structures. Its 3.3B parameter version, used in this work, provides strong performance across a wide range of language pairs, making it well-suited for multilingual and low-resource translation tasks.

In NLLB, we prepend a special token $\langle \text{lang}_{src} \rangle$ at the beginning of the source sentence to specify the source language and another special token $\langle \text{lang}_{tgt} \rangle$ to specify the target language. During inference, this last token guides the decoder to produce output in the desired language.

The NLLB encoder is partially fine-tuned during training, specifically the lower L layers, while the higher layers remain frozen to retain multilingual generalization. The Whisper encoder remains completely frozen and is used purely for speech feature extraction.

4.4 Language Conditioning and Token Embeddings

To handle multilingual input and output, we append the source language token $\langle \text{lang}_{src} \rangle$ to the reshaped speech representation and use $\langle \text{lang}_{tgt} \rangle$ in the decoder. Both tokens are embedded using NLLB's embedding layer. This token-based control mechanism enables seamless switching between languages during both training and inference.

Source and target training pairs are formatted as follows:

$$\begin{aligned} source &= \langle lang_{src} \rangle \ src\_sentence \ \langle eos \rangle \\ target &= \langle bos \rangle \ \langle lang_{tgt} \rangle \ tgt\_sentence \ \langle eos \rangle \end{aligned}$$

4.5 Hybrid Architecture

This hybrid configuration transforms the multilingual NLLB 3.3B model into a multi-functional system capable of both ASR and S2TT. The architecture leverages pre-trained speech representations from Whisper (specifically the Medium and Large-v3 variants) and integrates them into the NLLB framework. This design enables the system to operate in low-resource settings with minimal parameter updates. In this setup, high-level audio features are extracted from a frozen Whisper encoder, which serves solely as a feature extractor. These representations are then reshaped to align with the input format expected by the NLLB encoder. Crucially, this reshaped output replaces the traditional word embedding layer in the NLLB encoder, allowing the model to process audio input instead of text, and the efficiency of parameter training is achieved by only modifying the parameters of reshape module and the lower layers of the NLLB encoder. The architecture consists of three main components:

- A frozen Whisper encoder (either Medium or Large-v3),
- A reshape module that projects the audio embeddings into the required format,
- A multilingual NLLB 3.3B encoder-decoder model.

Figure 1 (right block) illustrates the complete hybrid S2TT architecture. Speech representations X, visualized as black squares, are generated by the Whisper encoder. These are subsequently reshaped X' and passed to the NLLB encoder, which processes them and generates translations from the outputs Z by applying a linear projection followed by a softmax function. By limiting fine-tuning to only the lower layers of the NLLB encoder and the reshape module, the model achieves parameter-efficient training while retaining multilingual capabilities.

The Whisper encoder outputs high-dimensional speech representations that are reshaped to match the input format expected by the NLLB encoder. This replaces the word embedding layer in NLLB with audio-derived embeddings. Mor formally:

$$\begin{aligned} X &= \texttt{Whisper}^{K}_{ENC}(a) && (1) \\ X' &= \texttt{EMB}(\langle lang_{src} \rangle) \cdot \texttt{Reshape}(X) && (2) \\ Y &= \texttt{NLLB}_{ENC}(X') && (3) \\ Z &= \texttt{NLLB}_{DEC}(Y) && (4) \end{aligned}$$

Here, a is the input audio signal, X is the speech representation, and X' is the concatenated input embedding. Y and Z represent the encoded and decoded outputs, respectively.

4.6 Parameter-Efficient Training Scenarios

We consider two training scenarios for low-resource adaptation:

- **Zero-shot:** Whisper and NLLB are used as is, without fine-tuning. Figure 1 illustrates this scenario.
- **Domain adaptation:** Parameter-efficient fine-tuning is performed:
  - Whisper is fine-tuned over Tunisian audio/transcription examples obtained from LDC in-domain data (LDC ASR).

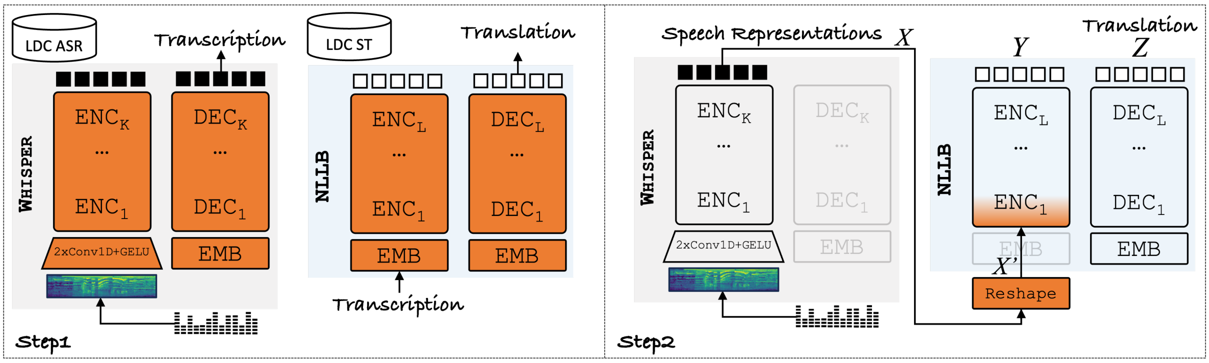

Figure 1: Overview of the Hybrid Whisper+NLLB Approach in a parameter-efficient domain adaptation scenario. The Whisper encoder/decoder is fine-tuned using LDC ASR data, while NLLB is fine-tuned on both transcription and translation text from LDC ASR and S2TT datasets (step1). Both models are then coupled to enable hybrid processing (step2). Red color indicates model weights being updated (the rest are kept freezed).

- NLLB is fine-tuned using in-domain transcription/translation examples.

Figure 1 illustrates the adaptation in domain for Whisper ASR tunisian and NLLB adaption to translate english or tunisian in LDC domain. In both cases, a last adaptation for coupling these two models is achieved by updating only a small subset of model parameters (e.g., the reshape module and lower layers of the NLLB encoder), enabling effective learning from limited resources.

5 Experimental work

5.1 Networks

Our **Coupling Hybrid** models are trained using a single NVIDIA H100 GPU (80GB) during up to 20 epochs, with a maximum batch size of 64 utterances and updates of the model after accumulating 64 batches. We validate every $1,000$ updates and perform early stopping on a separate validation set excluded from the training set. We use the lazy Adam algorithm (Kingma and Ba, 2014) for optimization. In inference, we use a beam size of 5.

5.2 Results

Table 5 summarizes the results obtained across various model configurations and architectures. We report BLEU scores (Post, 2018) and word error rates (WER)[3] as evaluation metrics for S2TT and ASR, respectively. WER is computed on normalized transcriptions[4].

BLEU and WER results are indicated over internal development and test sets, as provided by the task organizers. These splits are considered in our analysis. Best scores for each development/test set are highlighted in bold.

Columns *Whisper Inf Enc* and *Dec* indicate the number of encoder/decoder layers used during inference by Whisper. Similarly, *NMT Opt Enc* and *Dec* specify the number of encoder and decoder layers fine-tuned in the NLLB model. Note that we always use the 3.3B parameter version of NLLB.

During inference, NLLB consistently employs all its encoder/decoder layers. The *Size* column reports the total number of parameters used by each system during inference.

System **`Whisper M`** indicates the original Whisper Medium model, used for both ASR and S2TT tasks. Without fine-tuning the model obtains very poor transcription and translation scores. This is mainly because Whisper was pre-trained in modern standard Arabic (MSA) and lacks exposure to the Tunisian dialect, which severely limits its ability to handle dialectal input.

Systems **`Whisper M`**^{FT} and **`Whisper L`**^{FT} involve full fine-tuning of `Whisper Medium` and `Whisper Large v3` for ASR using the complete cleaned speech-transcription training data introduced in Section 2. These are the only two configurations in which Whisper is fine-tuned, resulting in considerably longer training times (nearly 2 days). Although their BLEU scores remain very low, similar to those of the baseline Whisper model, their ASR performance improves significantly after fine-tuning. Compared to the baseline **`Whisper M`**, which was not fine-tuned, both fine-tuned systems show significant improvements in ASR per-

[3] `https://huggingface.co/spaces/evaluate-metric/wer`

[4] Normalization is performed by `BasicTextNormalizer` from the `transformers.models.whisper` module.

Model	Data	Whisper Inf		NLLB Opt		Size	BLEU↑		WER↓	
		Enc	*Dec*	*Enc*	*Dec*		*dev*	*tst*	*dev*	*tst*
Whisper M	-	24	24	-	-	769M	1.18	1.14	157.28	168.23
Whisper *fine-tunned*										
Whisper M^{FT}	ASR	24	24	-	-	769M	1.17	1.15	44.31	53.41
Whisper L^{FT}	ASR	32	32	-	-	1550M	2.13	1.86	**43.70**	**50.23**
Cascade										
Whisper M^{FT} + NLLB	ASR	24	24	0	0	4.07B	5.45	4.64	44.31	53.41
Whisper L^{FT} + NLLB	ASR	32	32	0	0	4.85B	5.68	5.07	43.70	50.23
Whisper M^{FT} + NLLB^{FT}	ASR+MT	24	24	24	24	4.07B	19.25	16.44	44.31	53.41
Whisper L^{FT} + NLLB^{FT}	ASR+MT	32	32	24	24	4.85B	**19.77**	17.39	43.70	50.23
Hybrid										
Whisper M + NLLB	ST	24	-	2	0	4.07B	12.39	9.92	-	-
Whisper M + NLLB	ASR+ST	24	-	2	0	4.07B	9.10	7.44	77.71	85.07
Whisper M^{FT} + NLLB^{FT}	ST	24	-	2	0	4.07B	19.22	16.62	126.57	121.41
Whisper L^{FT} + NLLB^{FT}	ST	32	-	2	0	4.35B	19.37	**17.52**	149.31	139.48

Table 5: Translation (BLEU) and recognition (WER) results across various model configurations. The column *Data* shows data used for each configuration, the column *Whisper Inf* specifies the number of Whisper encoder/decoder layers used during inference, while *NMT Opt* shows the number of NLLB encoder/decoder layers optimized during training. The *Size* column denotes the total number of parameters used during inference.

formance. Specifically, **Whisper M**^{FT} achieves WERs of 44.31 and 53.41 on the dev and test sets, while **Whisper L**^{FT} further improves to 43.70 and 50.23. These results demonstrate the effectiveness of fine-tuning even without changes to the model architecture. However, BLEU scores remain low for in both cases as these models are not explicitly optimized for translation. The slight increase in BLEU for the larger model is likely due to more accurate transcriptions feeding into the implicit translation process, but overall, these scores confirm that fine-tuning Whisper solely for ASR is insufficient for reliable S2TT performance.

In the *Cascade* setup, systems **Whisper M**^{FT}**+NLLB** and **Whisper L**^{FT}**+NLLB** combine fine-tuned Whisper models (for ASR) with the base NLLB model (for MT). In this approach, Whisper is first fine-tuned on the LDC ASR dataset to generate transcriptions, which are then passed to the unadapted NLLB model for translation. These configurations do not yield strong translation performance, primarily due to the mismatch between the transcription domain and the NLLB training data. However, performance could be improved through domain adaptation of the NLLB component. When adapting the NLLB model with the available in-domain datasets, systems **Whisper M**^{FT}**+NLLB**^{FT} and **Whisper L**^{FT}**+NLLB**^{FT} clearly improve their translation performance. The NLLB model is fine-tuned on transcription-translation pairs from the LDC ASR and ST datasets. Thus, transcriptions produced by Whisper are then translated using the adapted NLLB network. These latter systems demonstrate the effectiveness of adapting NLLB to the ASR/ST domain using LDC transcriptions and translations. However, despite the improved accuracy, the inherent latency introduced by cascading models makes them less suitable for real-time or industrial applications, where efficiency is critical. WER scores remain constant across all cascade systems because the Whisper component, responsible for transcription, is identical within each Whisper variant. This consistency further confirms that BLEU gains are due solely to the adaptation of the translation model.

The next set of results pertains to our hybrid systems. We utilize fine-tuned versions of Whisper (Medium and Large v3) tightly coupled with NLLB as detailed in section 4. Similarly to the cascade setup, the first two systems use the original, pre-trained Whisper and NLLB models, while the latter two are hybrid systems that combine Whisper and NLLB models which have been previously fine-tuned.

One key advantage of the hybrid models lies in their compactness: they require significantly fewer parameters than the cascade counterparts. Furthermore, coupling optimization is computationally efficient. The Whisper speech encoder is kept frozen, while only 2 out of 24 layers in the NLLB encoder are fine-tuned. This strategy drastically reduces training time and computational cost. Fine-tuning with the LDC ST dataset required only 1 to 3 days, depending on the configuration and number of trainable parameters.

The first two hybrid systems, where Whisper and NLLB models are used without any fine-tuning, output moderate improvements over the raw Whisper model but significantly lower performance than domain-adapted cascade approaches. The system trained on both ASR and ST objectives (ASR+ST) exhibits a significant drop in both translation and transcription quality compared to the version trained solely on the ST objective (ST). This suggests that, in the absence of domain adaptation, multitask training may lead to interference between the tasks.

When hybridizing the adapted networks (last two rows), where both Whisper and NLLB are fine-tuned using in-domain LDC data, systems attain BLEU scores nearly equivalent to the best-performing cascade systems. These results validate the effectiveness of our lightweight hybrid fine-tuning strategy, which freezes most Whisper and NLLB layers, optimizing only a minimal subset. Notably, these hybrid models operate with lower parameter counts and exhibit superior latency characteristics compared to their cascade counterparts. WER scores, however, are higher in the hybrid domain-adapted models (ranging from 121 to 149), reflecting a trade-off in ASR accuracy potentially introduced by tighter integration and shared optimization. This is also partly due to the fact that the hybrid models were exclusively fine-tuned using speech translation (ST) data, without direct supervision on ASR objectives. As a result, while the models are optimized for generating accurate translations, their raw transcription outputs may be less precise, contributing to higher WER.

As expected, the hybrid models achieve S2TT performance comparable to the cascade systems. For example, the best hybrid domain adaptation configuration attains BLEU scores of 19.37 and 17.52 on the development and test sets, respectively. Importantly, these hybrid models offer superior latency characteristics, making them more suitable for deployment in real-time or resource-constrained environments compared to their cascade counterparts.

Finally, it is important to note that the results submitted for the evaluation of this task were obtained several epochs prior to the final version of the model. At that stage, the model achieved a BLEU score of 18.96 on the development set and 16.94 on the test set. The current version of our model outperforms the submitted one by approximately 0.5 BLEU points.

6 Conclusions and further work

We presented SYSTRAN's submitted systems for the 2025 Low-Resource Language Track, targeting the task of Tunisian Arabic to English speech translation. Our approach combines an ASR encoder (Whisper) with a neural machine translation decoder (NLLB), using light fine-tuning to create an efficient and compact speech translation pipeline. The resulting Speech-to-Text Translation system is designed to operate with minimal computational resources and limited training data. We evaluated our system against several alternative configurations, including a cascaded Whisper+NLLB setup and direct end-to-end fine-tuning of Whisper. Our results demonstrate that it is possible to achieve high translation quality under low-resource constraints, enabling broader accessibility without the need for large-scale infrastructure.

Acknowledgments

This work has been funded by the French Ministry of Defense through the DGA-RAPID 2022190955, COMMUTE project.

References

Marko Avila and Josep Crego. 2025. Leveraging large pre-trained multilingual models for high-quality speech-to-text translation on industry scenarios. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 7624–7633, Abu Dhabi, UAE. Association for Computational Linguistics.

Meghan Lammie Glenn, Stephanie Strassel, and Haejoong Lee. 2009. Xtrans: a speech annotation and transcription tool. In *Interspeech*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

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.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2023. Robust speech recognition via large-scale weak supervision. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Nllb team, Marta Ruiz Costa-jussà, James Cross, Onur cCelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Alison Youngblood, Bapi Akula, Loïc Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *ArXiv*, abs/2207.04672.