Improving Language and Modality Transfer in Translation by Character-level Modeling

Ioannis Tsiamas[1,2] **David Dale**[1] **Marta R. Costa-jussà**[1]
[1]FAIR at Meta, Paris [2]Universitat Politècnica de Catalunya, Barcelona
Correspondence: ytsiamas@meta.com, daviddale@meta.com

Abstract

Current translation systems, despite being highly multilingual, cover only 5% of the world's languages. Expanding language coverage to the long-tail of low-resource languages requires data-efficient methods that rely on cross-lingual and cross-modal knowledge transfer. To this end, we propose a character-based approach to improve adaptability to new languages and modalities. Our method leverages SONAR, a multilingual fixed-size embedding space with different modules for encoding and decoding. We use a teacher-student approach with parallel translation data to obtain a character-level encoder. Then, using ASR data, we train a lightweight adapter to connect a massively multilingual CTC ASR model (MMS), to the character-level encoder, potentially enabling speech translation from 1,000+ languages. Experimental results in text translation for 75 languages on FLORES+ demonstrate that our character-based approach can achieve better language transfer than traditional subword-based models, especially outperforming them in low-resource settings, and demonstrating better zero-shot generalizability to unseen languages. Our speech adaptation, maximizing knowledge transfer from the text modality, achieves state-of-the-art results in speech-to-text translation on the FLEURS benchmark on 33 languages, surpassing previous supervised and cascade models, albeit being a zero-shot model with minimal supervision from ASR data.

1 Introduction

Translation has experienced a large growth in terms of language coverage in the last years, with models supporting 200-400 languages in text (NLLB, 2024; Kudugunta et al., 2023), and 100 in speech (SEAMLESS, 2025). Although impressive in terms of population coverage (90%), in terms of actual language coverage we stand at only 5%.[1] Moving towards expanding to the long-tail of low-resource languages in the world posses some serious challenges due to the increasingly scarce data sources. For text translation we have to rely on a few thousand parallel sentences, while chances are there are no parallel data for speech translation (ST). To ease the issue of data scarcity in low-resource settings, multilinguality for text (Johnson et al., 2017; Chang et al., 2024) and multimodality for speech (Tang et al., 2021a) can usually be beneficial. But how can we increase cross-lingual and cross-modal knowledge transfer from high-resource languages and modalities? Recent research suggests that character-level models exhibit better cross-lingual transfer in text translation, especially in low-resource scenarios (Edman et al., 2024). Furthermore, for speech translation, methods usually take advantage of a text-based encoder for semantic modeling (Tang et al., 2021a;

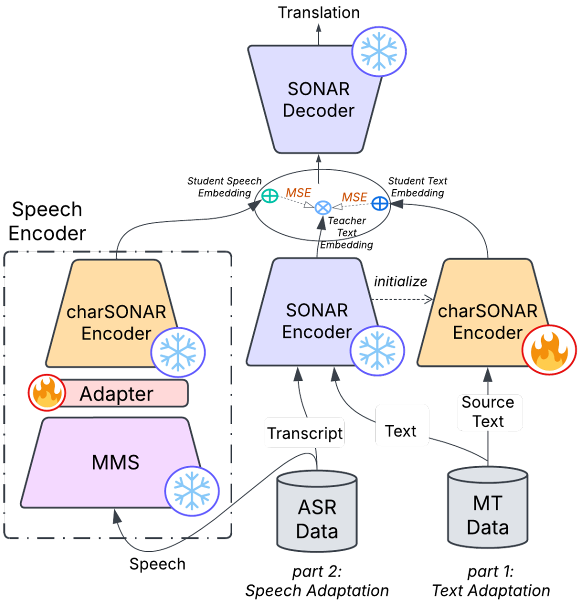

Figure 1: Approach for character-level and speech adaptation using the SONAR space.

[1]www.ethnologue.com

Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 20171–20187
July 27 - August 1, 2025

Zhang et al., 2023), but the subword-based tokenization is incompatible in terms of length and content with the acoustic representations, thus creating a *modality gap* that hinders knowledge transfer. Previous research mitigates this by either using a phoneme-based text encoder (Tang et al., 2021a; Le et al., 2023) or converting the acoustic representations to subword-like units (Tsiamas et al., 2024). But a phonemized input degrades performance due to ambiguity[2] and furthermore phonemizers for 1000+ languages might be infeasible (Zhao et al., 2024), while subword-based compression requires a substantial amount of data. To this end, we propose to shift towards character-based encoders, that could support data-efficient knowledge transfer both between languages and between text and speech. Our method is based on SONAR (Duquenne et al., 2023), which is an encoder-decoder with a fixed-size semantic embedding space that supports 200 languages, and on MMS (Pratap et al., 2023), which is a CTC-based ASR model that supports 1,000+ languages. Using a teacher-student approach, we obtain a character-based text encoder that embeds sentences in the SONAR space. Then, we propose an adapter that seamlessly connects the CTC output space of MMS to the character-level input space of our encoder, requiring minimal supervision from audio-transcription pairs (Fig. 1). Our experimental results in 75 languages on FLORES+ (NLLB, 2024), show that compared to traditional subword-based models, our multilingual character-level SONAR encoder exhibits better cross-lingual knowledge sharing between known languages and superior zero-shot generalizability to unseen languages. Furthermore, our speech adaptation of the character-based encoder, despite relying only on ASR data, can maximize knowledge transfer from text, and thus surpasses the previous best supervised system (SEAMLESS, 2025) and strong cascades with Whisper (Radford et al., 2022), achieving new state-of-the-art in FLEURS (Conneau et al., 2022).

2 Relevant Research

2.1 Character-level MT

Early works in machine translation investigated character-level approaches due their advantages in understanding and generating rare and unseen words, handling noise, having smaller vocabularies, and being simpler due to the removal of subword tokenization (Sennrich et al., 2016). Several methods using attention-based sequence-to-sequence models (Sutskever et al., 2014; Bahdanau et al., 2015) showed that character-level MT can reach or surpass subword-based approaches (Ling et al., 2015; Costa-jussà and Fonollosa, 2016; Lee et al., 2017; Cherry et al., 2018; Chung et al., 2016). Later, Xue et al. (2022) showed that ByT5, encoder-decoder multilingual language model operating on bytes, is more robust to noise and performs better in spelling-sensitive tasks, than its subword-based counterpart mT5 (Xue et al., 2021). Edman et al. (2024) fine-tuned the ByT5 and mT5 models for translation, and found that character-level modeling is particularity effective when parallel data are limited. Libovický et al. (2022) sought to answer why fully character-level MT has not been widely adopted, which was attributed to lower efficiency, and an inability to confirm previous findings that had been suggesting better domain and morphological generalization. In this work, we propose an encoder-only character-level approach (Cao, 2023) based on SONAR (Duquenne et al., 2023), and study the benefits of cross-lingual transfer in a large group of 75 languages, both in low-resource and in zero-shot settings. Several works have proposed methods that alleviate the additional computational costs stemming from the longer sequences that character- or byte-level models need to process (Clark et al., 2022; Tay et al., 2022; Pagnoni et al., 2024). Since our approach adopts character-level modeling only on the encoder side, and due to the fixed-size embedding bottleneck of SONAR, the computational overhead is minimal, and thus we do not study any architecture-based changes in this work.

2.2 Cross-modal Transfer in ST

Speech translation models have traditionally relied on cross-modal knowledge transfer from the more resourceful task of text translation to improve performance. Several works achieved this by using a multitasking framework of MT and ST, where they share the text modules between the two tasks, and the semantic text encoder accepts either acoustic representations or text embeddings as inputs (Liu et al., 2020; Ye et al., 2021; Tang et al., 2021b; Fang et al., 2022). Another line of work aims at bridging the modality gap by additionally minimizing the distance between the speech-text representations of the encoders (Tang et al., 2021a; Ye et al., 2022; Ouyang et al., 2023). ZeroSwot (Tsiamas et al., 2024) eliminated the dependency on parallel ST

[2] Homophones, loss of orthographic information, etc.

data, and relied only on minimizing the Wasserstein distance (Peyré and Cuturi, 2019) between the speech-text of representations of the encoders using ASR data. In our framework we also follow the paradigm of ZeroSwot, but due to the fixed-size encoder bottleneck of SONAR, our optimization is simpler, and minimizes the MSE distance. Another important consideration in maximizing knowledge transfer from text is unifying the tokenization of acoustic encoder's output and text encoder's input space, which are usually phoneme/characters for the the CTC (Graves et al., 2006) of the acoustic encoder, and subwords for the embedding layer of the text encoder. Previous works have either used a phoneme-based input for the text encoder (Tang et al., 2021a; Le et al., 2023), or a subword-based output for the acoustic encoder's CTC output (Liu et al., 2020; Yang et al., 2023), or more recently a character-to-subword compression adapter (Tsiamas et al., 2024). But phoneme-based text input degrades performance due to ambiguity in meaning. Then, for subword-based output in CTC, it is questionable whether it can scale to massively multilingual vocabularies of hundreds of thousand of tokens (NLLB, 2024). Finally the subword compression adapter requires a substantial amount of ASR data to learn, which can be problematic for the long tail of low-resource languages. Contrary, our approach is based on first modifying the text encoder to work with character-level inputs without degrading MT performance, and then learning a data-efficient and lightweight adapter that connects to it the character-based output space of a CTC acoustic encoder.

3 Methodology

We utilize the multilingual fixed-size embedding space of SONAR (Duquenne et al., 2023), in order to add new languages and modalities (speech) to it. We first obtain a character-level text encoder using a teacher-student approach with parallel translation data (§3.2), and then adapt it to work with CTC acoustic representations as inputs using a teacher-student approach with paired audio-transcriptions (§3.3).

3.1 SONAR

The SONAR encoder is a Transformer (Vaswani et al., 2017) with N_t layers of dimensionality d_t, and a subword-based vocabulary $\mathcal{V}_t$. The final encoder representation is mean-pooled to obtain a sentence embedding $\mathbf{e} \in \mathbb{R}^{d_t}$. The SONAR decoder which also has N_t layers of dimensionality d_t, attends with cross-attention to $\mathbf{e}$, in order to predict the target sequence.

3.2 Character-level Text Encoder

Our character-level encoder (charSONAR) is initialized from the SONAR encoder, and thus has N_t layers of dimensionality d_t. As part of the character-based input vocabulary we only keep the tokens of $\mathcal{V}_t$ that are composed of single characters, thus having a vocabulary $\mathcal{V}_c \subset \mathcal{V}_t$.

Training Objectives. For training, we follow a student-teacher approach with the SONAR encoder as a teacher, where we minimize the MSE loss between the charSONAR embedding $\mathbf{c}$ and a SONAR embedding $\mathbf{e} \in \mathbb{R}^d$, using monolingual or parallel translation data. We consider three different MSE objectives:

- *Reconstruction*, where we learn from non-parallel data, and given a sentence x, we minimize $\mathcal{L}^{recon} = \text{MSE}(\mathbf{c}^x, \mathbf{e}^x)$.
- *Translation*, where we learn from parallel data, and given a sentence x with translation y, we minimize $\mathcal{L}^{trans} = \text{MSE}(\mathbf{c}^x, \mathbf{e}^y)$.
- *Interpolation*, where we also learn from parallel data, and given a sentence x with translation y, we minimize the distance from the 'average' teacher embedding for that pair (Eq. 1).

$$\mathcal{L}^{interpol} = \text{MSE}\left(\mathbf{c}^x, \frac{\mathbf{e}^x + \mathbf{e}^y}{2}\right) \tag{1}$$

Augmentations. We apply ASR-like augmentations to make the character-based encoder robust to the normalized and error-prone output of CTC ASR models and increase cross-modal transfer. Specifically with some probability p^{norm}, we normalize the source text input of the char-based encoder, removing casing and punctuation. Furthermore, with some small probability p^{noise}, we inject different noise perturbations to the text, such as character addition, deletion and replacement.

3.3 Speech Encoder

Our speech encoder is composed of an acoustic encoder, an adapter, and the charSONAR encoder.

3.3.1 Acoustic Encoder

The acoustic encoder consists of a series of strided convolutional layers, followed by a Transformer encoder with N_s layers of dimensionality d_s. It is initialized from MMS (Pratap et al., 2023), which was pretrained with the self-supervised objective of wav2vec 2.0 (Baevski et al., 2020) and fine-tuned with CTC (Graves et al., 2006) on 1,000+ languages. Each language i has its own CTC prediction head $\mathbf{W}^{(i)} \in \mathbb{R}^{d_s \times |\mathcal{B}_i|}$, where $\mathcal{B}_i$ is a language-specific character-based vocabulary (including the <blank> token), with $\mathcal{B}^{(i)} \subset \mathcal{V}_c$.

The acoustic encoder is kept frozen during training, and with it we extract the final encoder representation $\mathbf{H} \in \mathbb{R}^{m \times d_s}$. Next, we apply CTC-based compression (Gaido et al., 2021) to remove redundancy and obtain a representation that is similar in length as the character-based tokenization of our charSONAR encoder. We label each point j of $\mathbf{H}$ with its CTC prediction $\pi_j = \text{argmax}(\mathbf{W}^{(i)}\mathbf{h}_j)$, then average consecutive points corresponding to the same prediction, and drop points corresponding to <blank>. We thus obtain an acoustic representation $\mathbf{A} \in \mathbb{R}^{n \times d_s}$, where $n < m$.

3.3.2 Cross-modal Adapter

We use a cross-modal adapter to process the acoustic representation $\mathbf{A}$ into an embedding-like representation $\mathbf{E} \in \mathbb{R}^{n \times d_t}$, that aims to match as close as possible the character embedding expected at the input of the charSONAR encoder.

To maximize pretrained knowledge and obtain an adapter that could work out-of-the box in extremely low-resource settings, we propose a minimal (pretrained) two-layer architecture that is fully initialized from MMS and charSONAR. Specially, we use the CTC classification layer $\mathbf{W}^{(i)}$ of MMS to project $\mathbf{A}$ to logits, and with a softmax we obtain a probability distribution over the MMS vocabulary $\mathcal{B}^{(i)}$. Then, since $\mathcal{B}^{(i)} \subset \mathcal{V}_c$, we can connect the two spaces by doing a soft prediction over the charSONAR vocabulary using its embedding layer.

$$\mathbf{E}^{\text{pt}} = \text{softmax}\left(\mathbf{A}\mathbf{W}^{(i)}\right)\mathbf{Emb}^{(i)},$$

where $\mathbf{Emb}^{(i)} \in \mathbb{R}^{|\mathcal{B}^{(i)}| \times d_t}$ is the embedding layer of charSONAR, indexed by the entries of $\mathcal{B}^{(i)}$.

Due to the nature of its initialization, the hidden dimension of the pretrained cross-modal adapter is fixed and bound to the size $|\mathcal{B}^{(i)}|$ of the MMS vocabulary, which is relatively small, usually having 64 tokens. In order to be able to control, and increase, the capacity of the adapter, we also propose a dual cross-modal adapter that combines the pretrained one with another variable-sized adapter that is randomly initialized (Fig. 2).

$$\mathbf{E}^{\text{rnd}} = \text{ReLU}\left(\mathbf{A}\mathbf{U}^{in}\right)\mathbf{U}^{out},$$

where $\mathbf{E}^{\text{rnd}}$ is the output of the randomly-initialized adapter, $\mathbf{U}^{in} \in \mathbb{R}^{d_s \times d_h}$ and $\mathbf{U}^{out} \in \mathbb{R}^{d_h \times d_t}$ are learnable parameters, and d_h is a hyperparameter that we can control. We concatenate the individual outputs of the pretrained and randomly-initialized adapters and pass them through an MLP : $2d_t \to 1$, followed by a sigmoid function to obtain a vector of weights $\mathbf{v} \in (0,1)^n$. The final representation $\mathbf{E}^{\text{dual}} \in \mathbb{R}^{n \times d_t}$ is a weighted sum of $\mathbf{E}^{\text{pt}}, \mathbf{E}^{\text{rnd}}$.

$$\mathbf{v} = \sigma\left(\text{MLP}\left([\mathbf{E}^{\text{pt}}, \mathbf{E}^{\text{rnd}}]\right)\right)$$
$$\mathbf{E}^{\text{dual}} = \mathbf{v}\mathbf{E}^{\text{pt}} + (1-\mathbf{v})\mathbf{E}^{\text{rnd}}$$

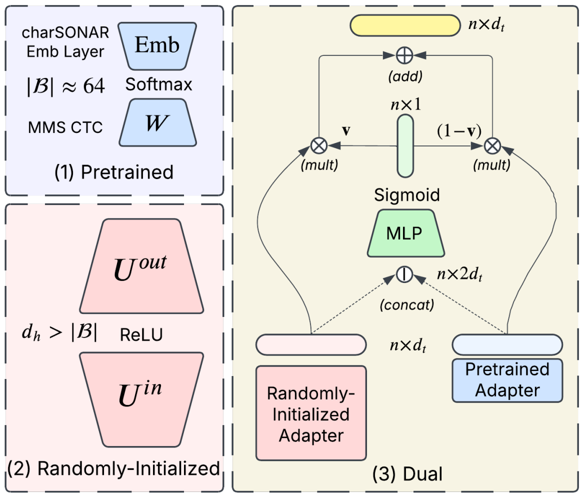

Figure 2: Cross-modal Adapters

To the output of the cross-modal adapter $\mathbf{E}$ we prepend the corresponding language token embedding, and append the embedding for end-of-sentence from the charSONAR embedding table. After adding positional encoding, $\mathbf{E}$ is passed through the transformer layers of the (frozen) charSONAR encoder to obtain a speech embedding $\mathbf{c}^z \in \mathbb{R}^{d_t}$. To train the adapter we use audio-transcription pairs, and minimize the MSE loss between $\mathbf{c}^z$ and the SONAR embedding for the transcription $\mathbf{e}^x$. For speech translation inference, we use the SONAR decoder to generate the translation from the speech embedding $\mathbf{c}^z$.

4 Experimental Setup

4.1 Data

Text. We construct a diverse group of 63 languages in terms of family and script, and with varying degrees of resourcefulness, that are already present in SONAR. We also add a group of 12 new languages, not present in SONAR, for which we have evaluation data in FLORES+.[3] For the 63 known languages, to train charSONAR we used a combination of human-labeled data and mined parallel data (NLLB, 2024), which were filtered with BLASER 2.0 (Dale and Costa-jussà, 2024), discarding pairs with score lower than 4. For the group of 12 new languages, we used various publicly available sources of parallel data. For validation and testing we used the `dev` and `devtest` splits of FLORES+.

Speech. For our experiments in speech we used the 33 source languages. Our criteria for choosing these languages where: (a) part of the text training; (b) supported by MMS; (c) included in the Common Voice (CV) (Ardila et al., 2020) dataset (ASR training data); and (d) included in the FLEURS (Conneau et al., 2022) dataset (ST evaluation data). For training we used the `train` split of the version 17.0 of CommonVoice,[4] and in some experiments the small `train` split of FLEURS which contains 2K examples for each language. Evaluation is done on the `dev` and `test` splits of FLEURS, which contain approximately 400 and 900 examples each.

Details regarding the languages and the amount of training data for both text and speech are available in Table 10 in the Appendix.

4.2 Model Architecture

The SONAR encoder (Duquenne et al., 2023) has $N_t = 24$ layers, with dimensionality of $d_t = 1024$, and an embedding table of size 256K (750M parameters in total). Our charSONAR encoder follows the same architecture, apart from the character-based embedding table with a size of 8K tokens (500M parameters in total). MMS (Pratap et al., 2023) has $N_s = 48$ layers with dimensionality $d_s = 1280$ (1B parameters).[5] It uses language-specific layers (Houlsby et al., 2019) and CTC classification layers. The vocabulary is different for each language, usually having around 64 tokens. Since the size of the vocabulary is also the hidden dimension of our pretrained adapter, this adapter has approximately 200K parameters. The randomly-initialized adapter uses a hidden dimension of $d_h = 1024$ (2.2M parameters). For the dual adapter we use an MLP with an inner dimension of 64 (100K parameters) to predict the weight vector, thus having a total of 2.5M parameters. To generate translations, either from text or speech, we couple the encoder with the SONAR decoder which has 24 layers. For X→Eng generation we use the normal SONAR decoder,[6] while for all other generation tasks we use the finetuned decoder,[7] which according to Duquenne et al. (2023) and our observations here, performs better.

4.3 Training Details

Text. We use AdamW (Loshchilov and Hutter, 2019) with a learning rate of 4e-4, inverse square root scheduler with warmup, a batch size of 12K examples, dropout of 0.1, and train for 128K steps. We up-sample languages with a temperature of 0.5 (NLLB, 2024). We apply ASR-like text normalization by un-casing and removing punctuation to a source sentence with $p^{norm} = 0.25$, and inject character-based noise with $p^{noise} = 0.125$. Specifically, each character in the source sentence can be deleted, replaced, or a new character is further added, each with a probability of 0.0025. These values were tuned in a small validation set of CV to approximate the character-error-rate of MMS (Pratap et al., 2023). For replacement and addition we sample a new character from the character distribution of that language.

Speech. Both MMS and charSONAR remain frozen during speech training, and only the adapter is finetuned. We minimize the MSE distance with the original SONAR as a teacher. The learning rate is set to 2e-4, the batch size 500 examples, and the adapter dropout to 0.1 (0.3 for the randomly-initialized adapter).

4.4 Evaluation

We apply checkpoint averaging according to the `dev` set performance, and generate with a beam search of 5. We evaluate primary on two tasks: translation and similarity search. Translation quality is measured with xCOMET-XL (Guerreiro et al., 2024)[8]. When the target language is not supported, we use case-sensitive detokenized BLEU (Post, 2018) and chrF++ (Popović, 2017). For similarity search we measure xSIM++ (Chen et al., 2023) error rates, by augmenting the English parts of FLORES or FLEURS with 40K hard negatives.

[3] huggingface.co/datasets/openlanguagedata/flores_plus
[4] datasets/mozilla-foundation/common_voice_17_0
[5] huggingface.co/facebook/mms-1b-all
[6] dl.fbaipublicfiles.com/SONAR/sonar_text_decoder.pt
[7] dl.fbaipublicfiles.com/SONAR/finetuned_decoder.pt
[8] huggingface.co/Unbabel/XCOMET-XL

5 Text Results

Here we present our results in text translation and similarity search with charSONAR, and investigate its capacity for cross-lingual knowledge transfer.

5.1 Initial Exploration

Before the main experiments we conduct an exploration regarding the training objectives and augmentations. We used a small subgroup of 15 languages, with 3 languages from the Uralic family, and 12 languages that use the Cyrillic script (Table 10). In the upper part of Table 1 we show that the proposed interpolated MSE objective surpasses both the reconstruction and translation MSE objectives, and additionally their combination. This shows that SONAR embeds sentences in sub-optimal regions, while there are regions in between the languages that are better suited for both translation and similarity search. Further motivation is provided by our results of Table 2, where we show that for a pair of non-English languages Lang_1 and Lang_2, decoding from their average SONAR embedding $\text{Emb}_{\text{AVG}} = (\text{Emb}_1 + \text{Emb}_2)/2$ into English, is better than decoding from each individual embedding, with low-resource languages benefiting the most.[9] This finding indicates that our charSONAR encoder can benefit from learning to map sentences to the interpolated or 'average' space existing between languages.

In the lower part of Table 1, we find that pretraining charSONAR with the reconstruction MSE before the interpolated MSE is beneficial, since it decouples learning character-level modeling and optimizing the embedding space. Finally, we see that the normalization and noise augmentation do not have an impact in performance. This is expected due to the ground truth source text, but as we show later in the initial exploration for the speech experiments (§6.1), these augmentations are beneficial, as the input to charSONAR is error-prone.

5.2 Scaling to 75 Languages

Next, we present our findings from scaling-up the language coverage of charSONAR to 75 languages. We use the interpolated MSE objective, with reconstruction MSE pretraining and ASR-like text augmentations. For new languages, which are not supported by SONAR, we use the translation MSE objective. We compare against SONAR-200 (Duquenne et al., 2023), and an NLLB-200 (NLLB, 2024) topline, which is not restricted by a bottleneck encoder representation. We also train a comparable subword-based model by further fine-tuning SONAR on the 75 languages with the same setup as we did for charSONAR. We report text translation and cross-lingual similarity search (X→Eng) results, and group results by language resourcefulness according to the amount of our training data (Table 10). Our results of Table 3 show the clear advantage of our character-based encoder, where charSONAR-75 outperforms the comparable SONAR-75, and additionally the NLLB topline in translation. The gains are more evident in the group of 21 low-resource languages, where cross-lingual transfer can be more impactful.

Model				COMET	xSIM++
SONAR-200				0.925	8.5
charSONAR-Ural/Cyrl					
Objective	**Pretrain**	**Norm**	**Noise**		
recon	✗	✗	✗	0.929	7.4
trans	✗	✗	✗	0.924	6.6
recon+trans	✗	✗	✗	0.929	6.8
interpol	✗	✗	✗	0.931	6.6
interpol	✓	✗	✗	**0.934**	**6.4**
interpol	✓	✓	✗	**0.934**	**6.4**
interpol	✓	✓	✓	**0.934**	6.5

Table 1: Ablations on training objectives and augmentations for the Ural/Cyrl language group (15 langs). Text translation COMET scores and cross-lingual xSIM++(↓) error rates on FLORES dev (X→Eng).

Pairs		**COMET**			**Advantage**	
Lang_1	Lang_2	Emb_1	Emb_2	Emb_{AVG}	Lang_1	Lang_2
Low	Low	0.788	0.795	0.864	+0.076	+0.069
Low	High	0.793	0.937	0.920	+0.137	-0.017
High	High	0.939	0.937	0.944	+0.005	+0.007

Table 2: COMET scores of translating from average (interpolated) embeddings, compared to translating from individual embeddings, for different pairs based on resourcefulness. Results in X→Eng FLORES dev averaged over 50 randomly-sampled pairs in each row.

5.3 Zero-shot Generalization

In our next experiment, we only train on the 63 known languages, and evaluate zero-shot on the 12 new ones. SONAR and NLLB encoders require a language tag to be prepended in the source sequence, which is problematic if we want to encode a sentence from a language not seen during train-

[9] High-resource are negatively impacted when paired with low-resource, but this reflects only a very small fraction of their data.

Model	COMET (↑)					xSIM++ (↓)				
	Low (21)	Med (21)	High (21)	All (63)	New (12)	Low (21)	Med (21)	High (21)	All (63)	New (12)
Previous Works (trained on 200, not including the 12 new)										
NLLB-200	0.877	**0.914**	**0.949**	0.913	0.454†	-	-	-	-	-
SONAR-200	0.851	0.894	0.944	0.897	0.450†	13.1	10.1	7.3	10.2	52.7†
This work (trained on 63 known + 12 new languages)										
SONAR-75	0.882	0.909	0.948	0.913	0.859	9.0	7.7	5.8	7.5	12.6
charSONAR-75	**0.889**	**0.914**	**0.949**	**0.917**	**0.863**	8.4	**7.2**	**5.5**	**7.0**	**12.3**
Δ	0.007	0.005	0.001	0.004	0.004	0.6	0.5	0.3	0.5	0.3
This work (trained only on the 63 known languages)										
SONAR-63	0.882	0.909	0.947	0.913	0.517†	8.8	7.7	5.8	7.5	42.9†
charSONAR-63	**0.889**	**0.914**	**0.949**	**0.917**	<u>0.530</u>†	**8.3**	**7.2**	**5.5**	**7.0**	<u>42.2</u>†
Δ	0.007	0.005	0.002	0.004	0.013	0.5	0.5	0.3	0.5	0.7

Table 3: Translation COMET scores and cross-lingual similarity search xSIM++ error rates on FLORES `devtest` (X→Eng), grouped by All(Low/Med/High) and New languages. † indicates zero-shot evaluation. **bold**: best overall; <u>underlined</u>: best zero-shot. All models have the same number of parameters (1.3B). Δ refers to the difference between charSONAR-N and SONAR-N models.

Model	BLEU	chrF++
NLLB-200	17.4	45.3
SONAR-200	15.6	44.0
SONAR-Eng	**15.9**	**44.8**
charSONAR-Eng	15.8	44.7

Table 4: Text translation (Eng→200) BLEU and chrF++ scores on FLORES `devtest`.

Model	# Tokens	Inference Time (s)
SONAR	49	0.84
charSONAR	158 (×3.2)	0.94 (×1.1)

Table 5: Average number of tokens and average inference time in FLORES `dev`.

ing. To achieve better encoding for these unseen languages, we propose the use of family tokens according to the linguistic family subgroup of each language. Specifically, during training we replace the language token with the corresponding subgroup token with a 20% probability. On inference, we encode a new language, with the appropriate subgroup token. The subgroup tokens are trainable and are initialized from the average of the all the language tokens of each family.[10] In the last part of Table 3 we observe that charSONAR-63 can generalize better than a subword-based encoder to languages not seen during training, achieving an improvement of 0.013 points in COMET and 0.7 in xSIM++. We also notice a sharp increase for both our encoders, compared to original SONAR-200, showing the benefits of expanding language tokens to subgroup tokens.[11]

5.4 Are the gains due to language transfer or more compute?

An implicit side-effect of character-level modeling is that sequences are on average 3× longer, which means that the charSONAR encoder is using more FLOPs than the SONAR encoder. To further investigate the source of the advantage shown in Table 3 we conduct an experiment where we train SONAR and charSONAR on only one language, specifically on English. The results of Table 4 show that in the single-language setting, there is no advantage for the character-based model, being slightly behind the subword-based one. This finding indicates that character-level modeling is beneficial due to better cross-lingual knowledge transfer, rather than due to increased compute.

5.5 Efficiency Analysis

To assess the degree of computational overhead due to the longer sequences, we measure the average inference time for the charSONAR and SONAR models in FLORES `dev` using a batch size of 1. The results of Table 5 show that although sequences

[10]Information about the linguistic families are available at Table 10 in the Appendix.

[11]For encoding a new language with SONAR-200 and NLLB-200, we do not use a language tag.

are 3.2× longer for charSONAR, the inference time is only 1.1× longer. This is due to the encoder bottleneck, which decouples the decoder from the source sequence length. Results with batching are available in Table 16 in the Appendix.

6 Speech Results

To investigate the cross-modal benefits of character-level modeling, we present results in zero-shot speech translation and speech-text similarity search with our charSONAR-based speech encoder.

6.1 Initial Exploration

In Table 6 we present zero-shot ST X→Eng results in FLEURS dev for four languages of different families and of varying degree of resourcefulness, ranging from 3K examples (Estonian) to 330K (Spanish). We observe that the pretrained cross-modal adapter (PRETR), despite being significantly smaller, outperforms the large ($d_h = 1024$), yet randomly initialized, adapter (RND). Although for the high-resource Spanish, we notice that the difference is rather small, which indicates that with more data it can be beneficial to increase the capacity. Indeed, our proposed dual adapter (DUAL), with large dimensionality in the random branch, surpasses them both. Finally, we notice further gains when we switch to a robust charSONAR version that was trained with ASR-like Norm/Noise augmentations (§3.2).

Adapter			Encoder	COMET				
Type	Dim	Train	Norm/Noise	Est	Rus	Tur	Spa	Avg
PRETR	∼64	✗	✗ / ✗	0.845	0.849	0.828	0.837	0.840
PRETR	∼64	✓	✗ / ✗	0.901	0.910	0.877	0.890	0.894
RND	256	✓	✗ / ✗	0.837	0.872	0.831	0.878	0.854
RND	1024	✓	✗ / ✗	0.882	0.889	0.869	0.889	0.882
DUAL	256	✓	✗ / ✗	0.914	0.912	0.889	0.888	0.901
DUAL	1024	✓	✗ / ✗	0.911	0.909	0.894	**0.905**	0.905
DUAL	1024	✓	✓ / ✗	0.914	0.910	0.891	0.897	0.903
DUAL	1024	✓	✓ / ✓	**0.915**	**0.923**	**0.906**	**0.905**	**0.912**

Table 6: Ablations in speech adaptation. Speech Translation (X→Eng) results on FLEURS dev.

6.2 Zero-shot Speech Translation

Next, we train adapters for 33 languages using the charSONAR-75 encoder, and compare against strong supervised E2E models, Whisper (Radford et al., 2022) and SeamlessM4T (SEAMLESS, 2025), cascades with MMS/Whisper and NLLB/SONAR, and our own cascades with SONAR-75/charSONAR-75. We report results by grouping languages according to number of examples in CommonVoice (CV) (Table 7). The first version of our system (11) using the pretrained adapter, can work out-of-the-box and without any training, even outperforming Whisper by a large margin, and particularly for low-resource languages. This indicates that the input space of our character-based encoder is fully compatible with the output space of MMS given the initialization of our adapter. Following, by training this adapter with ASR data (12), we surpass the previous state-of-the-art SeamlessM4T-Large-v2. The benefits of cross-modal transfer from charSONAR are evident for some extremely low-resource languages such as Asturian, where with only 400 examples it surpasses SeamlessM4T by 0.1 COMET (Table 12 in Appendix). Furthermore, we observe additional gains when using the proposed dual adapter (13) for medium/high-resource languages, where there are enough data to learn the large, but randomly initialized, branch. Finally, we show that by adding only 2K additional examples from FLEURS train, we can achieve further important gains across all categories (14-15).

Apart from the strong cross-modal transfer showcased by our speech adaptation of charSONAR, significant gains are also observed for the cascade systems that employ it. Specifically a cascade of MMS and charSONAR (9) outperforms all other cascades (3-8) and is on par with SeamlessM4T-Large-v2 in low/medium-resource settings.

6.3 Similarity Search

In Table 8 we present results on cross-lingual and cross-modal similarity search on FLEURS test. We compare our character-based speech encoder against several cascades and the original SONAR speech encoders (Duquenne et al., 2023) that are based on w2v-BERT (Chung et al., 2021), and thus do not transfer knowledge from an acoustic model (MMS) nor the text modality (charSONAR). We observe that our minimal adapters trained on CV outperform the previous SONAR speech encoders and all cascades apart from the charSONAR-based one. Still, by using more data (FLEURS train) and/or more parameters (dual adapter), the proposed encoder surpasses the charSONAR cascades.

6.4 Adapters for Subword-based Encoders

To understand how essential is the character-based encoder in our proposed speech adaptation, we experiment with replacing it with a subword-based

id	Model	Text Encoder Tokenization	Total Params	Adapter Train Params	Adapter Train Data	Low (11)	Med (11)	High (11)	All (33)
Supervised E2E ST (previous)									
1	WHISPER-LARGE-v3	/	1.5B	/	/	0.598	0.754	0.790	0.714
2	SEAMLESSM4T-LARGE-v2	/	2.3B	/	/	0.829	0.889	0.901	0.873
Cascade ST (previous)									
3	MMS + NLLB-200	subwords	2.3B	/	/	0.786	0.834	0.822	0.814
4	WHISPER + NLLB-200	subwords	2.8B	/	/	0.717	0.870	0.863	0.817
5	MMS + SONAR-200	subwords	2.3B	/	/	0.757	0.839	0.824	0.807
6	WHISPER + SONAR-200	subwords	2.8B	/	/	0.684	0.869	0.861	0.804
Cascade ST (ours)									
7	MMS + SONAR-75	subwords	2.3B	/	/	0.811	0.870	0.854	0.845
8	WHISPER + SONAR-75	subwords	2.8B	/	/	0.721	0.871	0.865	0.819
9	MMS + charSONAR-75	chars	2.3B	/	/	0.833	0.889	0.875	0.866
10	WHISPER + charSONAR-75	chars	2.8B	/	/	0.755	0.893	0.882	0.843
Zero-shot E2E ST (ours)									
11	Speech-charSONAR-75 - PRETR	chars	2.3B	0	/	0.772	0.833	0.831	0.812
12	Speech-charSONAR-75 - PRETR	chars	2.3B	0.2M	CV	0.837	0.893	0.894	0.875
13	Speech-charSONAR-75 - DUAL	chars	2.3B	2.5M	CV	0.615	0.899	0.902	0.805
14	Speech-charSONAR-75 - PRETR	chars	2.3B	0.2M	CV+FLEURS	0.852	0.900	0.901	0.884
15	Speech-charSONAR-75 - DUAL	chars	2.3B	2.5M	CV+FLEURS	**0.853**	**0.906**	**0.910**	**0.889**

Table 7: Speech Translation (33 → Eng) COMET scores on FLEURS `test`. Low/Med/High each contain 11 languages, according to amount of Common Voice data. Underlined are the previous best scores. Highlighted are our scores with char-based models that are at least on par with the previous best. In **bold** are best overall.

Model	avg26	avg33
SONAR Speech (Duquenne et al., 2023)	14.3	/
MMS + SONAR-200	15.7	17.2
MMS + SONAR-75	12.7	13.8
MMS + charSONAR-75	10.7	11.5
Speech-charSONAR-75 - PRETR	11.6	12.9
↪ w/ FLEURS `train`	10.0	10.7
↪ w/ DUAL	**9.4**	**10.2**

Table 8: Cross-modal and cross-lingual retrieval. xSIM++ error rates (↓) on FLEURS `test` (X→Eng). avg26 is the languages supported by SONAR Speech (Duquenne et al., 2023) and our models.

one. To achieve this we mean-pool the indices of the compressed acoustic representation of MMS that belong to the same subword, as predicted by the CTC. The pretrained adapter version is not possible in this setting and thus we experiment only with the randomly-initialized adapter. The results of Table 9 indicate that we can learn an adapter to connect MMS and (subword-based) SONAR, although the quality is limited, and only works in high resource settings (Turkish, Spanish), while still being several points behind the character-based model. This highlights both the data-efficiency and cross-modal adaptability of our proposed method.

Model	Oci # 0.3k	Est # 3k	Tur # 30k	Spa # 330k
Speech-SONAR-75 - RND	0.199	0.223	0.795	0.841
Speech-charSONAR-75 - RND	0.202	0.877	0.868	0.912
Speech-charSONAR-75 - PRETR	**0.795**	0.910	0.885	0.917
Speech-charSONAR-75 - DUAL	0.707	**0.912**	**0.903**	**0.920**

Table 9: Speech Translation COMET (X→Eng) on FLEURS `test` for subword-based vs char-based encoders with adapters. Underlined: best among RND adapters; **bold**: best overall; #: ASR examples.

7 Conclusions

We presented a methodology based on character-level modeling that increases cross-lingual transfer and cross-modal transfer in text and speech tasks. For text, our character-based encoder surpasses comparable subword-based encoders, especially in low-resource settings, while exhibiting better zero-shot generalization to unseen languages. For speech, our proposed minimal adapter seamlessly connects an ASR CTC encoder to our character-based encoder, surpassing previous state-of-the-art models. Furthermore it requires minimal supervision from ASR data, and can even work out-of-the-box without any training, surpassing models like Whisper. Future research will focus on target-side cross-lingual and cross-modal transfer, and expanding to more languages.

Limitations

In this work we focused on source-side cross-lingual and cross-modal transfer, leaving target-side transfer for future research. We hypothesize that character-level modeling can be beneficial for target-side, although decoding on the character-level can be problematic and relatively more inefficient than encoding on the character-level. We still believe this is an interesting direction for future work.

Furthermore, we decided to focus on adapting a specific model, SONAR, to work with character-level input. Although the encoder bottleneck reduces the computational overhead in generation, and allowed us to simplify the teacher-student training by using an MSE objective, it also reduces the capacity of the model. We hypothesize that similar gains can be achieved by adapting a traditional encoder-decoder, like NLLB (NLLB, 2024), to work with characters, either by back-propagating the translation signal through the (frozen) decoder or using similar objectives to ZeroSwot (Tsiamas et al., 2024). Also, as discussed in our Relevant Research (§2), we did not experiment with any specific architectural changes in the encoder that are better suited for character-level modeling (Clark et al., 2022; Tay et al., 2022; Pagnoni et al., 2024), as we aimed to study character-based vs subword-based modeling within the same architecture. We believe that by using such techniques further gains in performance and efficiency can be achieved.

Additionally, our proposed methodology for speech adaptation is limited by the language-specific CTC layers of MMS. This forced us to train language-specific cross-modal adapters, which does not allow the speech encoder to generalize to more languages, other than the ones for which we have ASR data. To go around this issue, we carried some experiments with the zero-shot version of MMS (Zhao et al., 2024) that uses a unified model for all languages, but due to decreased ASR quality compared to MMS-1B (Pratap et al., 2023), translation quality was also lagging behind. Still, in the future, and given a supervised MMS-like acoustic model with a unified architecture, our proposed cross-modal adapter could enable generalized speech understand and translation with it.

Acknowledgments

We thank Pere Lluís Huguet Cabot for his helpful comments and suggestions.

References

Rosana Ardila, Megan Branson, Kelly Davis, Michael Kohler, Josh Meyer, Michael Henretty, Reuben Morais, Lindsay Saunders, Francis Tyers, and Gregor Weber. 2020. Common Voice: A Massively-Multilingual Speech Corpus. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4218–4222, Marseille, France. European Language Resources Association.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A Framework for Self-Supervised Learning of Speech Representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural Machine Translation by Jointly Learning to Align and Translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Kris Cao. 2023. What is the best recipe for character-level encoder-only modelling? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5924–5938, Toronto, Canada. Association for Computational Linguistics.

Tyler A. Chang, Catherine Arnett, Zhuowen Tu, and Ben Bergen. 2024. When Is Multilinguality a Curse? Language Modeling for 250 High- and Low-Resource Languages. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 4074–4096, Miami, Florida, USA. Association for Computational Linguistics.

Mingda Chen, Kevin Heffernan, Onur Çelebi, Alexandre Mourachko, and Holger Schwenk. 2023. xSIM++: An Improved Proxy to Bitext Mining Performance for Low-Resource Languages. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 101–109, Toronto, Canada. Association for Computational Linguistics.

Colin Cherry, George Foster, Ankur Bapna, Orhan Firat, and Wolfgang Macherey. 2018. Revisiting Character-Based Neural Machine Translation with Capacity and Compression. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4295–4305, Brussels, Belgium. Association for Computational Linguistics.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A Character-level Decoder without Explicit Segmentation for Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1693–1703, Berlin, Germany. Association for Computational Linguistics.

Yu-An Chung, Yu Zhang, Wei Han, Chung-Cheng Chiu, James Qin, Ruoming Pang, and Yonghui Wu. 2021. w2v-BERT: Combining Contrastive Learning and Masked Language Modeling for Self-Supervised Speech Pre-Training. In *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 244–250.

Jonathan H. Clark, Dan Garrette, Iulia Turc, and John Wieting. 2022. Canine: Pre-training an Efficient Tokenization-Free Encoder for Language Representation. *Transactions of the Association for Computational Linguistics*, 10:73–91.

Alexis Conneau, Min Ma, Simran Khanuja, Yu Zhang, Vera Axelrod, Siddharth Dalmia, Jason Riesa, Clara Rivera, and Ankur Bapna. 2022. FLEURS: Few-shot Learning Evaluation of Universal Representations of Speech. *arXiv preprint arXiv:2205.12446*.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based Neural Machine Translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 357–361, Berlin, Germany. Association for Computational Linguistics.

David Dale and Marta R. Costa-jussà. 2024. BLASER 2.0: A Metric for Evaluation and Quality Estimation of Massively Multilingual Speech and Text Translation. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16075–16085, Miami, Florida, USA. Association for Computational Linguistics.

Paul-Ambroise Duquenne, Holger Schwenk, and Benoît Sagot. 2023. SONAR: Sentence-Level Multimodal and Language-Agnostic Representations. *Preprint*, arXiv:2308.11466.

Lukas Edman, Gabriele Sarti, Antonio Toral, Gertjan van Noord, and Arianna Bisazza. 2024. Are Character-level Translations Worth the Wait? Comparing ByT5 and mT5 for Machine Translation. *Transactions of the Association for Computational Linguistics*, 12:392–410.

Qingkai Fang, Rong Ye, Lei Li, Yang Feng, and Mingxuan Wang. 2022. STEMM: Self-learning with Speech-text Manifold Mixup for Speech Translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7050–7062, Dublin, Ireland. Association for Computational Linguistics.

Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi. 2021. CTC-based Compression for Direct Speech Translation. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 690–696, Online. Association for Computational Linguistics.

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*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

Nuno M. Guerreiro, Ricardo Rei, Daan van Stigt, Luisa Coheur, Pierre Colombo, and André F. T. Martins. 2024. xCOMET: Transparent Machine Translation Evaluation through Fine-grained Error Detection. *Transactions of the Association for Computational Linguistics*, 12:979–995.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-Efficient Transfer Learning for NLP. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 2790–2799. PMLR.

Melvin Johnson, Mike Schuster, Quoc V. Le, Maxim Krikun, Yonghui Wu, Zhifeng Chen, Nikhil Thorat, Fernanda Viégas, Martin Wattenberg, Greg Corrado, Macduff Hughes, and Jeffrey Dean. 2017. Google's Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation. *Transactions of the Association for Computational Linguistics*, 5:339–351.

Sneha Kudugunta, Isaac Rayburn Caswell, Biao Zhang, Xavier Garcia, Derrick Xin, Aditya Kusupati, Romi Stella, Ankur Bapna, and Orhan Firat. 2023. MADLAD-400: A Multilingual And Document-Level Large Audited Dataset. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Phuong-Hang Le, Hongyu Gong, Changhan Wang, Juan Pino, Benjamin Lecouteux, and Didier Schwab. 2023. Pre-training for Speech Translation: CTC Meets Optimal Transport. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Jason Lee, Kyunghyun Cho, and Thomas Hofmann. 2017. Fully Character-Level Neural Machine Translation without Explicit Segmentation. *Transactions of the Association for Computational Linguistics*, 5:365–378.

Jindřich Libovický, Helmut Schmid, and Alexander Fraser. 2022. Why don't people use character-level machine translation? In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2470–2485, Dublin, Ireland. Association for Computational Linguistics.

Wang Ling, Isabel Trancoso, Chris Dyer, and Alan W Black. 2015. Character-based Neural Machine Translation. *Preprint*, arXiv:1511.04586.

Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. 2020. Bridging the Modality Gap for Speech-to-Text Translation. *Preprint*, arXiv:2010.14920.

Ilya Loshchilov and Frank Hutter. 2019. Decoupled Weight Decay Regularization. In *International Conference on Learning Representations*.

NLLB. 2024. Scaling Neural Machine Translation to 200 Languages. *Nature*, 630:841–846.

Siqi Ouyang, Rong Ye, and Lei Li. 2023. WACO: Word-Aligned Contrastive Learning for Speech Translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3891–3907, Toronto, Canada. Association for Computational Linguistics.

Artidoro Pagnoni, Ram Pasunuru, Pedro Rodriguez, John Nguyen, Benjamin Muller, Margaret Li, Chunting Zhou, Lili Yu, Jason Weston, Luke Zettlemoyer, Gargi Ghosh, Mike Lewis, Ari Holtzman, and Srinivasan Iyer. 2024. Byte Latent Transformer: Patches Scale Better Than Tokens. *Preprint*, arXiv:2412.09871.

Gabriel Peyré and Marco Cuturi. 2019. Computational Optimal Transport: With Applications to Data Science. *Foundations and Trends® in Machine Learning*, 11(5-6):355–607.

Maja Popović. 2017. chrF++: Words Helping Character N-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. 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, Belgium, Brussels. Association for Computational Linguistics.

Vineel Pratap, Andros Tjandra, Bowen Shi, Paden Tomasello, Arun Babu, Sayani Kundu, Ali Elkahky, Zhaoheng Ni, Apoorv Vyas, Maryam Fazel-Zarandi, Alexei Baevski, Yossi Adi, Xiaohui Zhang, Wei-Ning Hsu, Alexis Conneau, and Michael Auli. 2023. Scaling Speech Technology to 1,000+ Languages. *Preprint*, arXiv:2305.13516.

Alec Radford, Jong Wook Kim, Tao Xu, Greg Brockman, Christine McLeavey, and Ilya Sutskever. 2022. Robust Speech Recognition via Large-Scale Weak Supervision. *Preprint*, arXiv:2212.04356.

SEAMLESS. 2025. Joint Speech and Text Machine Translation for up to 100 Languages. *Nature*, 637:587–593.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to Sequence Learning with Neural Networks. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'14, page 3104–3112, Cambridge, MA, USA. MIT Press.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021a. Improving Speech Translation by Understanding and Learning from the Auxiliary Text Translation Task. 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 4252–4261, Online. Association for Computational Linguistics.

Yun Tang, Juan Pino, Changhan Wang, Xutai Ma, and Dmitriy Genzel. 2021b. A General Multi-Task Learning Framework to Leverage Text Data for Speech to Text Tasks. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6209–6213.

Yi Tay, Vinh Q. Tran, Sebastian Ruder, Jai Gupta, Hyung Won Chung, Dara Bahri, Zhen Qin, Simon Baumgartner, Cong Yu, and Donald Metzler. 2022. Charformer: Fast Character Transformers via Gradient-based Subword Tokenization. In *International Conference on Learning Representations*.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2024. Pushing the Limits of Zero-shot End-to-End Speech Translation. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 14245–14267, Bangkok, Thailand. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Linting Xue, Aditya Barua, Noah Constant, Rami Al-Rfou, Sharan Narang, Mihir Kale, Adam Roberts, and Colin Raffel. 2022. ByT5: Towards a Token-Free Future with Pre-trained Byte-to-Byte Models. *Transactions of the Association for Computational Linguistics*, 10:291–306.

Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2021. mT5: A Massively Multilingual Pre-trained Text-to-Text Transformer. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 483–498, Online. Association for Computational Linguistics.

Jichen Yang, Kai Fan, Minpeng Liao, Boxing Chen, and Zhongqiang Huang. 2023. Towards Zero-shot Learning for End-to-end Cross-modal Translation Models. In *Findings of the Association for Computational*

Linguistics: EMNLP 2023, pages 13078–13087, Singapore. Association for Computational Linguistics.

Rong Ye, Mingxuan Wang, and Lei Li. 2021. End-to-End Speech Translation via Cross-Modal Progressive Training. In *Interspeech 2021*, pages 2267–2271.

Rong Ye, Mingxuan Wang, and Lei Li. 2022. Cross-modal Contrastive Learning for Speech Translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5099–5113, Seattle, United States. Association for Computational Linguistics.

Yuhao Zhang, Chen Xu, Bei Li, Hao Chen, Tong Xiao, Chunliang Zhang, and Jingbo Zhu. 2023. Rethinking and Improving Multi-task Learning for End-to-end Speech Translation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 10753–10765, Singapore. Association for Computational Linguistics.

Jinming Zhao, Vineel Pratap, and Michael Auli. 2024. Scaling A Simple Approach to Zero-Shot Speech Recognition. *Preprint*, arXiv:2407.17852.

A Data

In Table 10 we provide details about the languages and the amounts of data used in our experiments. The numbers for the MT data indicate the amount after filtering with BLASER 2.0 (Dale and Costa-jussà, 2024). The resourcefulness label (low, medium, or high) of each language is separate for each modality, and indicates in which of the three percentile of the data distribution it belongs.

B Additional Results

In Table 11 we present the per-language text translation results for the models of Table 3.

In Table 12, we present the per-language speech translation results for some of the models of Table 7. We also add results from the SONAR Speech Encoders (Duquenne et al., 2023), which were excluded from the main table since they do not support all the languages with which we experiment. Furthermore, Table 12 includes more than 33 languages, since for ease of presentation, in the main results we presented the ones that were both supported by Whisper and our models. We indicate the languages not taken into account for the results of Table 7.

In Tables 14 and 13, we present the per linguistic subgroup and script results of our text encoders in translation and cross-lingual similarity search. We observe that for known languages, the charSONAR encoder outperforms the subword-based encoder

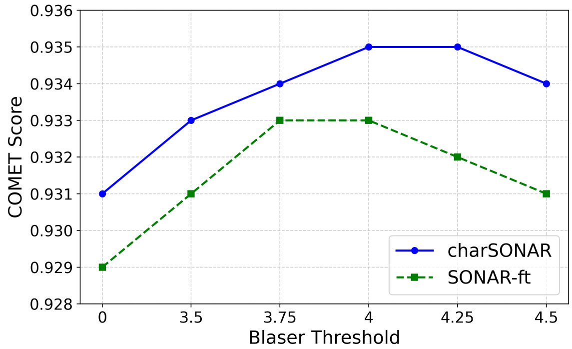

Figure 3: COMET scores vs. BLASER 2.0 filtering threshold for charSONAR and SONAR in FLORES dev. Results with the Ural/Cyrl group of 15 languages. COMET scores are average of X→Eng for all the 15 languages in the group.

in all categories, apart from the single group that contains the Greek language, and only in cross-lingual similarity search. For the new languages, we notice that charSONAR performs better in all categories for translation, but the subword-based model is better for the Turkic and Uralic subgrouping, and Cyrillic script in cross-lingual similarity search.

In Table 15 we present the text translation results for the three encoders that were used in the initial exploration with the four languages for the speech adaptation (Table 6). We used the Uralic/Cyrillic encoder for Estonian and Russian, the Turkic for Turkish, and the Romance for Spanish.

In Figure 3 we present our ablation for deciding the BLASER 2.0 filtering threshold. To speed-up experimentation and use less data, we filtered with 4.5 for the initial exploration, but for the main experiments we used a threshold of 4.

Finally, in Table 16 we provide an efficiency analysis for SONAR and charSONAR models, similar to the results of §5.5, but now with batching. We use length-based bucketing and a batch size of 5K tokens, which results in 8 batches for SONAR and 31 batches for charSONAR. The results here confirm the findings of Table 5, showing that the impact of the char-based tokenization is minimal with respect to the additional computational overhead.

Language	Code	FLORES+	MMS	Family	Subgrouping	Script	MT		ASR	
							# (M) / Resource		# (K) / Resource	
Aragonese	arg_Latn	✓	✗	Indo-European	Italic	Latin	0.1	new	0.0	-
Asturian	ast_Latn	✗	✓	Indo-European	Italic	Latin	0.2	low	0.4	-
Awadhi	awa_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	0.4	low	0.0	-
South Azerbaijani	azb_Arab	✗	✓	Turkic	Common Turkic	Arabic	0.3	low	0.0	-
North Azerbaijani	azj_Latn	✗	✓	Turkic	Common Turkic	Latin	9.4	med	0.1	low
Bashkir	bak_Cyrl	✗	✓	Turkic	Common Turkic	Cyrillic	1.7	med	119.2	-
Belarusian	bel_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	11.5	med	347.6	high
Bhojpuri	bho_Deva	✗	✗	Indo-European	Indo-Aryan	Devanagari	0.6	low	0.0	-
Bosnian	bos_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	21.8	high	0.0	-
Boro	brx_Deva	✓	✗	Sino-Tibetan	Tibeto-Burman	Devanagari	0.1	new	0.0	-
Bulgarian	bul_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	39.3	high	4.8	med
Catalan	cat_Latn	✗	✓	Indo-European	Italic	Latin	10.1	med	1146.2	high
Valencian	cat_Latn_vale1252	✓	✗	Indo-European	Italic	Latin	0.0	new	0.0	-
Czech	ces_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	52.3	high	20.1	med
Chuvash	chv_Cyrl	✓	✓	Turkic	Oghuric	Cyrillic	1.2	new	1.4	-
Central Kurdish	ckb_Arab	✗	✓	Indo-European	Iranian	Arabic	1.7	med	7.7	-
Crimean Tatar	crh_Latn	✗	✓	Turkic	Common Turkic	Latin	0.2	low	0.0	-
Dogri	dgo_Deva	✓	✓	Indo-European	Indo-Aryan	Devanagari	0.1	new	0.0	-
Greek	ell_Grek	✗	✓	Indo-European	Graeco-Phrygian	Greek	52.6	high	1.9	low
Estonian	est_Latn	✗	✓	Uralic	Finnic	Latin	16.9	high	3.2	med
Finnish	fin_Latn	✗	✓	Uralic	Finnic	Latin	32.6	high	2.1	low
French	fra_Latn	✗	✓	Indo-European	Italic	Latin	144.9	high	558.1	high
Friulian	fur_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	0.0	-
Galician	glg_Latn	✗	✓	Indo-European	Italic	Latin	6.5	med	25.2	high
Konkani	gom_Deva	✓	✗	Indo-European	Indo-Aryan	Devanagari	0.1	new	0.0	-
Hindi	hin_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	35.6	high	4.7	med
Chhattisgarhi	hne_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	0.3	low	0.0	-
Croatian	hrv_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	17.1	high	0.0	-
Hungarian	hun_Latn	✗	✓	Uralic	–	Latin	32.6	high	37.1	high
Italian	ita_Latn	✗	✓	Indo-European	Italic	Latin	95.7	high	169.8	high
Karakalpak	kaa_Latn	✓	✓	Turkic	Kipchak	Latin	0.3	new	0.0	-
Kashmiri	kas_Deva	✗	✗	Indo-European	Indo-Aryan	Devanagari	0.1	low	0.0	-
Kazakh	kaz_Cyrl	✗	✓	Turkic	Common Turkic	Cyrillic	5.6	med	0.5	low
Halh Mongolian	khk_Cyrl	✗	✓	Mongolic-Khitan	Mongolic	Cyrillic	0.5	low	2.2	low
Kyrgyz	kir_Cyrl	✗	✓	Turkic	Common Turkic	Cyrillic	2.7	med	1.8	-
Northern Kurdish	kmr_Latn	✗	✓	Indo-European	Iranian	Latin	0.7	med	5.1	-
Ligurian	lij_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	1.6	-
Lithuanian	lit_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	14.0	high	7.3	med
Lombard	lmo_Latn	✗	✗	Indo-European	Italic	Latin	0.3	low	0.0	-
Latgalian	ltg_Latn	✗	✗	Indo-European	Balto-Slavic	Latin	0.3	low	3.7	-
Standard Latvian	lvs_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	2.8	med	11.4	med
Magahi	mag_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	0.3	low	0.0	-
Maithili	mai_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	0.4	low	0.0	-
Marathi	mar_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	11.8	med	2.2	low
Meadow Mari	mhr_Cyrl	✓	✓	Uralic	Finno-Ugric	Cyrillic	0.4	new	185.9	-
Macedonian	mkd_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	6.8	med	1.7	low
Erzya	myv_Cyrl	✓	✓	Uralic	Mordvinic	Cyrillic	0.1	new	1.2	-
Nepali	npi_Deva	✗	✓	Indo-European	Indo-Aryan	Devanagari	4.5	med	0.3	low
Occitan	oci_Latn	✗	✓	Indo-European	Italic	Latin	0.2	low	0.3	low
Aranese	oci_Latn_aran1260	✓	✗	Indo-European	Italic	Latin	0.0	new	0.0	-
Southern Pashto	pbt_Arab	✗	✗	Indo-European	Iranian	Arabic	0.9	med	0.0	-
Western Persian	pes_Arab	✗	✓	Indo-European	Iranian	Arabic	15.0	high	28.9	high
Polish	pol_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	60.4	high	20.7	med
Portuguese	por_Latn	✗	✓	Indo-European	Italic	Latin	116.8	high	22.0	med
Dari	prs_Arab	✗	✗	Indo-European	Iranian	Arabic	0.9	med	0.0	-
Romanian	ron_Latn	✗	✓	Indo-European	Italic	Latin	59.9	high	5.1	med
Russian	rus_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	89.0	high	26.4	high
Sanskrit	san_Deva	✗	✗	Indo-European	Indo-Aryan	Devanagari	0.3	low	0.0	-
Sicilian	scn_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	0.0	-
Slovak	slk_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	29.7	high	3.3	med
Slovenian	slv_Latn	✗	✓	Indo-European	Balto-Slavic	Latin	20.8	high	1.4	low
Sindhi	snd_Deva	✓	✓	Indo-European	Indo-Aryan	Devanagari	0.0	new	0.0	-
Spanish	spa_Latn	✗	✓	Indo-European	Italic	Latin	202.0	high	336.8	high
Sardinian	srd_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	0.5	-
Serbian	srp_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	5.5	med	1.9	low
Silesian	szl_Latn	✗	✗	Indo-European	Balto-Slavic	Latin	0.4	low	0.0	-
Tatar	tat_Cyrl	✗	✓	Turkic	Common Turkic	Cyrillic	2.1	med	9.3	-
Tajik	tgk_Cyrl	✗	✓	Indo-European	Iranian	Cyrillic	1.1	med	0.0	-
Turkmen	tuk_Latn	✗	✓	Turkic	Common Turkic	Latin	0.6	low	0.8	-
Turkish	tur_Latn	✗	✓	Turkic	Common Turkic	Latin	47.4	high	35.1	high
Tuvan	tyv_Cyrl	✓	✗	Turkic	Common Turkic	Cyrillic	0.2	new	0.0	-
Uyghur	uig_Arab	✗	✓	Turkic	Common Turkic	Arabic	0.8	med	9.7	-
Ukrainian	ukr_Cyrl	✗	✓	Indo-European	Balto-Slavic	Cyrillic	12.2	med	25.1	med
Northern Uzbek	uzn_Latn	✗	✓	Turkic	Common Turkic	Latin	4.1	med	48.5	high
Venetian	vec_Latn	✗	✗	Indo-European	Italic	Latin	0.2	low	0.0	-

Table 10: Details about the languages used in our experiments.

	NLLB-200	SONAR-200	SONAR-63	charSONAR-63	SONAR-75	charSONAR-75
arg_Latn	0.867†	0.860†	0.910†	0.912†	**0.931**	0.917
ast_Latn	0.918	0.909	0.920	**0.928**	0.920	0.927
awa_Deva	**0.918**	0.861	0.894	0.896	0.898	0.898
azb_Arab	0.632	0.513	**0.698**	**0.698**	0.681	0.696
azj_Latn	**0.910**	0.741	0.863	0.868	0.865	0.870
bak_Cyrl	0.916	0.900	0.915	**0.920**	0.916	0.918
bel_Cyrl	**0.928**	0.905	0.922	0.926	0.920	0.926
bho_Deva	0.907	0.879	0.902	**0.908**	0.904	0.906
bos_Latn	**0.961**	0.953	0.960	0.960	0.959	0.960
brx_Deva*	0.208†	0.203†	0.217†	0.216†	0.841	**0.851**
bul_Cyrl	**0.954**	0.952	0.952	0.953	0.952	**0.954**
cat_Latn	0.954	0.949	0.952	**0.956**	0.952	**0.956**
cat_Latn_vale1252	0.903†	0.886†	0.952†	0.955†	0.952†	0.955†
ces_Latn	0.954	0.953	0.955	**0.956**	0.955	**0.956**
chv_Cyrl	0.240†	0.251†	0.274†	0.268†	0.851	**0.855**
ckb_Arab	0.881	0.872	0.882	0.889	0.884	**0.890**
crh_Latn	0.907	0.890	0.909	**0.920**	0.914	0.918
dgo_Deva*	0.606†	0.567†	0.575†	0.590†	0.895	**0.899**
ell_Grek	**0.940**	0.932	0.938	0.938	0.937	0.939
est_Latn	0.940	0.935	0.942	**0.945**	0.942	**0.945**
fin_Latn	0.942	0.936	0.945	0.946	0.944	**0.947**
fra_Latn	**0.965**	0.961	0.961	0.961	0.959	0.961
fur_Latn	0.931	0.930	0.937	**0.938**	0.936	**0.938**
glg_Latn	**0.957**	0.951	0.954	**0.957**	0.954	**0.957**
gom_Deva*	0.511†	0.477†	0.607†	0.635†	0.869	**0.880**
hin_Deva	**0.939**	0.935	0.934	0.935	0.934	0.938
hne_Deva	0.911	0.902	0.913	0.915	0.912	**0.916**
hrv_Latn	0.947	0.949	**0.954**	**0.954**	**0.954**	**0.954**
hun_Latn	0.946	0.940	0.945	**0.948**	0.945	0.947
ita_Latn	**0.957**	0.954	0.951	0.954	0.952	0.954
kaa_Latn	0.349†	0.359†	0.698†	0.767†	0.917	**0.924**
kas_Deva	0.657	0.600	0.674	0.704	0.681	**0.710**
kaz_Cyrl	**0.918**	0.908	0.912	0.917	0.909	0.917
khk_Cyrl	0.852	0.838	0.863	0.873	0.865	**0.876**
kir_Cyrl	0.908	0.898	0.911	**0.915**	0.909	**0.915**
kmr_Latn	0.784	0.776	0.789	**0.804**	0.789	0.800
lij_Latn	0.907	0.896	0.912	**0.914**	0.910	**0.914**
lit_Latn	0.930	0.921	0.933	**0.938**	0.934	0.937
lmo_Latn	0.866	0.833	0.884	**0.900**	0.885	0.898
ltg_Latn	0.888	0.866	0.900	**0.917**	0.900	0.916
lvs_Latn	0.928	0.915	0.932	**0.938**	0.932	0.937
mag_Deva	0.931	0.927	0.926	**0.935**	0.929	0.930
mai_Deva	**0.930**	0.881	0.907	0.905	0.906	0.906
mar_Deva	**0.929**	0.920	0.920	0.925	0.921	0.927
mhr_Cyrl*	0.262†	0.278†	0.268†	0.307†	0.896	**0.901**
mkd_Cyrl	0.946	0.942	0.946	**0.952**	0.947	0.951
myv_Cyrl	0.245†	0.243†	0.251†	0.259†	**0.851**	0.846
npi_Deva	**0.926**	0.881	0.900	0.901	0.900	0.901
oci_Latn	0.956	0.952	0.958	0.958	0.957	**0.959**
oci_Latn_aran1260	0.505†	0.500†	0.569†	0.576†	0.566†	0.571†
pbt_Arab	0.866	0.855	0.870	0.872	0.872	**0.874**
pes_Arab	0.924	0.918	0.922	**0.926**	0.925	**0.926**
pol_Latn	**0.950**	0.946	0.946	0.949	0.948	0.948
por_Latn	**0.963**	0.960	0.961	0.962	0.960	0.962
prs_Arab	0.901	0.900	0.908	**0.910**	0.908	**0.910**
ron_Latn	**0.964**	0.961	0.963	**0.964**	**0.964**	**0.964**
rus_Cyrl	**0.950**	0.942	0.944	0.946	0.944	0.946
san_Deva	0.749	0.702	0.731	0.737	0.732	0.742
scn_Latn	0.896	0.877	0.904	**0.915**	0.905	0.914
slk_Latn	0.955	0.951	0.953	**0.956**	0.952	0.955
slv_Latn	0.944	0.943	0.948	**0.951**	0.948	**0.951**
snd_Deva*	0.466†	0.464†	0.512†	0.497†	0.860	**0.869**
spa_Latn	**0.950**	0.949	0.946	0.948	0.946	0.948
srd_Latn	0.909	0.902	0.917	**0.918**	0.915	0.916
srp_Cyrl	0.949	0.943	0.950	**0.954**	0.950	**0.954**
szl_Latn	0.930	0.920	0.934	**0.942**	0.933	0.940
tat_Cyrl	0.927	0.915	0.925	0.927	0.925	**0.928**
tgk_Cyrl	0.915	0.903	0.915	**0.925**	0.916	0.923
tuk_Latn	0.893	0.882	0.898	0.910	0.904	**0.912**
tur_Latn	**0.946**	0.940	0.943	**0.946**	0.945	0.945
tyv_Cyrl	0.281†	0.313†	0.366†	0.381†	0.880	**0.884**
uig_Arab	0.863	0.853	0.859	**0.868**	0.859	0.867
ukr_Cyrl	0.948	0.942	0.945	0.948	0.946	**0.949**
uzn_Latn	**0.931**	0.914	0.918	0.924	0.920	0.926
vec_Latn	0.920	0.907	0.931	0.933	0.930	**0.936**

Table 11: Text Translation COMET (X→Eng) scores in FLORES `devtest`. ∗ indicates translation is evaluated on `dev` split. † indicates that the result is zero-shot. Underlined is the best among the zero-shot for each language, if any. In **bold** is the best for supervised results for each language, if any.

	E2E ST			charSONAR Cascades		Speech-charSONAR (CV)			+ FLEURS	
Language	**SONAR**	**Whisper**	**SeamlessM4T**	**w/ MMS**	**w/ Whisper**	**Random**	**Pretrained**	**Dual**	**Pretrained**	**Dual**
ast_Latn*	-	-	0.752	0.821	-	0.204	0.816	0.223	0.840	**0.849**
azj_Latn	-	0.642	**0.837**	0.762	0.777	0.197	0.788	0.194	0.788	0.764
bel_Cyrl	0.823	0.676	0.885	0.864	0.842	0.856	0.864	0.868	0.874	**0.886**
bos_Latn	0.882	0.829	**0.919**	0.899	0.911	-	-	-	-	-
bul_Cyrl	0.852	0.816	0.902	0.884	0.891	0.879	0.891	0.898	0.892	**0.907**
cat_Latn	0.886	0.874	0.929	0.893	0.929	0.903	0.900	0.922	0.913	**0.930**
ces_Latn	0.873	0.811	0.910	0.889	0.904	0.888	0.890	0.904	0.902	**0.911**
ckb_Arab*	-	-	0.766	0.786	-	0.692	0.800	0.798	0.802	**0.818**
ell_Grek	-	0.747	0.868	0.860	0.874	0.216	0.866	0.866	0.872	**0.889**
est_Latn	0.805	0.629	0.898	0.908	0.902	0.877	0.910	0.912	**0.918**	0.914
fin_Latn	0.799	0.768	0.887	0.900	0.926	0.229	0.902	0.904	**0.906**	0.905
fra_Latn	0.870	0.891	0.910	0.884	**0.935**	0.888	0.906	0.914	0.914	0.922
glg_Latn	-	0.830	0.917	0.897	0.908	0.859	0.908	**0.913**	0.906	0.910
hin_Deva	0.770	0.746	0.856	0.856	0.834	0.854	0.876	0.875	0.880	**0.882**
hrv_Latn*	0.866	0.816	0.895	0.903	**0.912**	-	-	-	-	-
hun_Latn	-	0.724	0.878	0.867	0.886	0.870	0.866	0.882	0.884	**0.892**
ita_Latn	0.892	0.898	0.927	0.915	**0.943**	0.914	0.924	0.926	0.932	0.939
kaz_Cyrl	-	0.349	0.846	0.844	0.774	0.198	0.849	0.198	0.866	**0.876**
khk_Cyrl	-	0.207	0.748	0.731	0.342	0.229	0.732	0.213	0.763	**0.764**
kir_Cyrl*	-	-	0.854	0.837	-	0.197	0.855	0.853	0.861	**0.865**
lit_Latn	0.766	0.579	0.832	0.880	0.853	0.858	0.881	0.885	0.884	**0.897**
lvs_Latn	0.848	0.589	0.885	0.899	0.888	0.879	0.905	0.909	0.908	**0.910**
mar_Deva	0.734	0.503	0.821	0.812	0.684	0.734	0.836	0.843	0.840	**0.846**
mkd_Cyrl	0.887	0.808	0.917	0.919	0.912	0.246	0.927	0.925	**0.928**	**0.928**
npi_Deva	0.675	0.538	**0.826**	0.790	0.652	0.196	0.810	0.798	0.801	0.799
oci_Latn	-	0.483	0.568	0.747	0.583	0.202	0.707	0.707	0.795	**0.805**
pes_Arab	0.810	0.666	0.887	0.867	0.851	0.875	0.890	0.884	0.887	**0.899**
pol_Latn	0.860	0.856	0.893	0.888	**0.923**	0.876	0.888	0.899	0.898	0.908
por_Latn	0.878	0.906	0.897	0.902	**0.941**	0.885	0.908	0.917	0.918	0.922
ron_Latn	0.856	0.867	0.909	0.895	**0.919**	0.855	0.900	0.904	0.906	0.905
rus_Cyrl	0.878	0.893	**0.912**	0.883	0.934	0.883	0.898	0.908	0.903	0.910
slk_Latn	0.885	0.822	0.914	0.911	0.924	0.876	0.909	0.917	0.919	**0.921**
slv_Latn	0.843	0.672	0.879	0.871	0.852	0.201	0.871	0.200	**0.883**	0.875
snd_Deva*	0.360	0.360	0.443	**0.698**	0.423	-	-	-	-	-
spa_Latn	0.888	0.893	0.908	0.899	**0.934**	0.912	0.917	0.920	0.922	0.930
srp_Cyrl	0.891	0.856	0.924	0.924	**0.929**	0.235	0.922	0.913	0.927	0.928
tgk_Cyrl*	-	0.523	0.858	**0.874**	0.658	-	-	-	-	-
tur_Latn	0.743	0.827	0.888	0.878	**0.924**	0.868	0.885	0.903	0.899	0.909
ukr_Cyrl	0.858	0.865	0.912	0.890	**0.931**	0.887	0.903	0.904	0.911	0.915
uzn_Latn	0.736	0.326	0.846	0.753	0.527	0.798	0.839	0.843	0.846	**0.854**

Table 12: Speech Translation COMET scores (X→Eng) on FLEURS `test`. The 6 languages with ∗ where not part of the main results of Table 7, since they were not supported either by our models or by Whisper.

	Subgrouping				Script		
	Indic	**Romance**	**Turkic**	**Uralic**	**Devanagari**	**Latin**	**Cyrillic**
# Languages	4	3	3	2	4	4	4
COMET							
SONAR	0.428	0.749	0.305	0.260	0.428	0.651	0.269
SONAR-75	0.866	0.813	0.882	0.869	0.866	0.839	0.867
charSONAR-75	**0.875**	**0.814**	**0.888**	**0.874**	**0.875**	**0.842**	**0.871**
XSIM++							
SONAR-200	53.8	29.3	63.5	70.2	53.8	36.1	68.6
SONAR-75	8.7	21.6	**9.6**	**10.5**	8.7	17.7	**11.0**
charSONAR-75	**8.2**	**20.9**	10.1	10.8	**8.2**	**16.9**	11.8

Table 13: Text translation (COMET) and text retrieval (xSIM++) results per language subgroup and script for the 12 newly added languages. Results in FLORES `devtest` (X→Eng).

	Subgrouping								Script				
	Balto-Slavic	**Romance**	**Turkic**	**Indic**	**Iranian**	**Uralic**	**Mongolic**	**Greek**	**Latin**	**Cyrillic**	**Devanagari**	**Arabic**	**Greek**
# Languages	16	15	11	10	6	3	1	1	34	12	10	6	1
COMET													
SONAR-200	0.934	0.926	0.850	0.849	0.871	0.937	0.838	0.932	0.917	0.916	0.849	0.818	0.932
SONAR-75	0.942	0.935	0.884	0.870	0.881	0.944	0.864	0.936	0.929	0.925	0.870	0.852	0.936
charSONAR-75	**0.946**	**0.940**	**0.892**	**0.877**	**0.887**	**0.946**	**0.876**	**0.939**	**0.934**	**0.930**	**0.877**	**0.860**	**0.939**
XSIM++													
SONAR	8.1	8.0	13.4	13.5	10.7	7.1	13.3	9.2	8.3	9.7	13.5	16.1	9.2
SONAR-75	6.2	5.5	10.0	9.7	8.2	5.7	11.0	**6.7**	6.1	7.7	9.7	11.6	**6.7**
charSONAR	**5.8**	**5.2**	**9.4**	**9.1**	**7.7**	**5.3**	**10.1**	6.9	**5.7**	**7.1**	**9.1**	**11.0**	6.9

Table 14: Text translation (COMET) and text retrieval (xSIM++) results per language subgroup and script for the 63 known training languages. Results in FLORES `devtest` (X→Eng).

Model	**Ural/Cyrl**	**Turkic**	**Romance**	**Avg**
SONAR-200	0.925	0.857	0.932	0.905
SONAR-group	0.930	0.879	0.942	0.917
charSONAR-group	**0.934**	**0.881**	0.946	**0.920**
↪ w/ Norm	**0.934**	0.877	0.946	0.919
↪ w/ Norm & Noise	**0.934**	0.878	**0.947**	**0.920**

Table 15: Text Translation COMET scores (X→Eng) in FLORES `dev`. Each encoder was trained on the respective group of languages.

Model	**# Tokens**	**Inference Time (s)**
SONAR	49	127
charSONAR	158 (×3.2)	142 (+10%)

Table 16: Average number of tokens and average inference time in FLORES `dev` with batching (5K tokens per batch).