Multi-lingual Speech to Speech Translation for Under-Resourced Languages

Anthony Larcher, Yannick Estève, Mickael Rouvier, Natalia Tomashenko, Jarod Duret, Gaelle Laperriere, Santosh Kesijaru, Marek Sarvas, Renata Kohlova, Henry Li, et al.

Esperanto
Exchanges for SPEech
ReseArch aNd TechnOlogies
Horizon 2020 project

Exchanges for SPEech ReseArch aNd TechnOlogies

Deliverable D6.1

Multi-lingual Speech to Speech Translation for Under-Resourced Languages

Funding Instrument: MSCA RISE
Call: H2020-MSCA-RISE-2020

Project Start: 1 January 2021
Project Duration: 48 months

Beneficiary in Charge: LMU

Dissemination Level		
PU	Public	✓
PP	Restricted to other programme participants (including the Commission Services)	
RE	Restricted to a group specified by the Consortium (including the Commission Services)	
CO	Confidential, only for members of the Consortium (including the Commission Services)	

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101007666

Deliverable Information

Document Administrative Information	
Project Acronym:	ESPERANTO
Project Number:	101007666
Deliverable Number:	D6.1
Deliverable Full Title:	Multi-lingual Speech to Speech Translation for Under-Resourced Languages
Deliverable Short Title:	Report on 2022 Workshop
Beneficiary in Charge:	LMU
Report Version:	v1.0
Contractual Date:	26/07/2022
Report Submission Date:	26/07/2022
Dissemination Level:	PU
Nature:	Report
Lead Author(s):	Anthony Larcher (LMU)
Co-author(s):	[Names of co-authors (partners short names)]
Keywords:	Community fostering, Under-resourced languages
Status:	_x_ draft, __ final, __ submitted

Table of Contents

This report describe the research done during the first ESPERANTO/JSALT workshop from the 13^{th} June 2022 to the 5^{th} of August 2022.

1 Participants

Table 1: List of participants to the JSALT 2022 workshop in ESPERANTO team and authors of this report.

Institution		Partner	First name	Name	Status
AU	Avignon University	✓	Yannick	ESTEVE	Senior
			Mickael	ROUVIER	Senior
			Natalia	TOMASHENKO	Senior
			Jarod	DURET	Junior
			Gaelle	LAPERRIERE	Junior
BUT	Brno University of Technology	✓	Santosh	KESIJARU	Senior
			Marek	SARVAS	Junior
			Renata	KOHLOVA	Admin
JHU	Johns Hopkins University	✓	Henry	LI	Junior
LMU	Le Mans University	✓	Anthony	LARCHER	Leader
			Antoine	LAURENT	Senior
			Thibault	GAUDIER	Junior
			Valentin	PELLOIN	Junior
			Thomas	THEBAUD	Junior
			Emmanuelle	BILLARD	Admin
LNE	Laboratoire National de Metrologie et d'Essai	✓	Olivier	GALIBERT	Senior
			Swen	RIBEIRO	Senior
	Microsoft		Harshita	DIDDEE	Junior
MIT	Massachusetts Institute of Technology		Sameer	KHURANA	Junior
	Naver Labs		Laurent	BESACIER	Senior
			Ioan	CALAPODESCU	Senior
Omilia	OMILIA	✓	Themos	STAFYLAKIS	Senior
PHON	PHONEXIA	✓	Tomas	PAVLICECK	Senior
UGA	University of Grenoble Alpes	✓	Cecile	MACAIRE	Junior
	University of Pittsburgh		Alejandro	CIUBA	Junior
USFD	University of Sheffield	✓	Peter	VICKERS	Junior
UNIZAR	University of Zaragoza	✓	Luis	VICENTE	Senior
			Victoria	MINGOTE	Senior
			Pablo	GIMENO	Junior

2 Global scope of the project

Seamless communication between people speaking different languages is a long term dream of humanity. Artificial intelligence aims at reaching this goal. Despite recent huge improvements made for Machine Translation, Speech Recognition and Speech Translation, Speech to Speech Translation (SST) remains a central problem in natural language processing, especially for under-resourced languages. A solution to this problem is to gather and share information across modalities and large resource languages to create a **Common multi-modal multi-lingual representation space** that could then be used to process under-resourced one through transfer learning, as depicted in Figure 1.

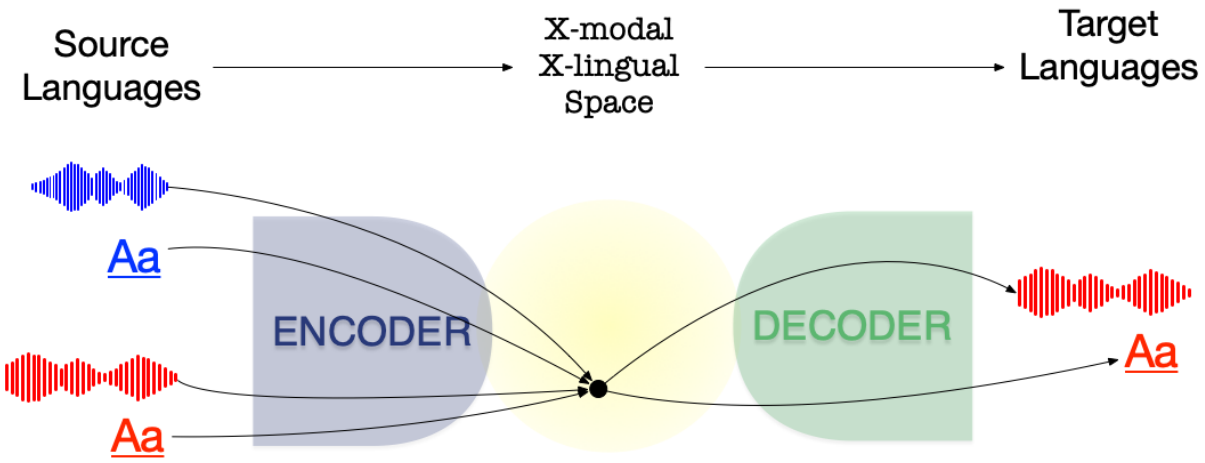


Figure 1: Block diagram of the multi-modal / multi-lingual translation.

The main goal of our project is twofold

1. Develop a fully multi modal system for which speech and text can be the input or output modalities in any considered language.
2. Adapt to under-resourced language
   - avoid catastrophic forgetting
   - transfer learning with incomplete data (when not all modalities are available)

A fully multi-modal/multilingual system, as depicted in Figure 1, allows to train with multiple objectives (driven by the available data). Tasks such as machine translation (text to text), speech translation (speech to target language text), speech recognition (speech to source language text) and text to speech will be considered as auxiliary tasks to drive the learning of the joint space as well as to ensure a good transfer for the under-resourced languages (where by definition, labeled data in all modalities are not always available).

The following issues have been addressed during the workshop:

- **Project speech and text modalities in the same space**
  Currently, SSL models exist for multi-modal mono-lingual representations in English [3] but are difficult to produce in many languages. On the other hand, mono-modal multi-lingual representations exist for speech [5] and text modalities [23]. During the workshop we started working from the SAMU-XLS-R encoder that has been trained to merge those two representation spaces into a single multi-model / multi-lingual space. Part of our work consisted in analysing the space produced by this encoder to better understand its

behaviour

- **Align modalities with different temporalities**
  An important issue when aligning speech and text sequences is the temporality and granularity of the sequences. More precisely, speech is a long, continuous and redundant audio sequence while the text is shorter and made of discrete tokens. Thus, the following research questions arose: How to design the space? Initial idea consists in combining some language-specific sub-spaces and a shared common space as was done for image translation [24]. We've tried different approaches to align spaces from different modalities across languages.
- **Transfer knowledge to under-resourced languages**
  It is common to develop language technologies for under-resourced languages by transferring knowledge from a model pre-trained on a large quantity of data. Exploiting the multi-modal/multi-lingual paradigm, we investigated the adaptation for low-resourced languages (Tamasheq).

3 Context at the beginning of the workshop

A large part of the work described in this report has been built on top of recent works from some team members [32]. More precisely, our initial baseline to project both speech and text in a cross-lingual and cross-modal space was based on the SAMU-XLS-R encoder, a self-supervised model.

Self-supervised representation learning (SSL) approaches such as Wav2Vec-2.0 [7], HuBERT [26], and WavLM [12] aim to provide powerful deep feature learning (speech embedding) without requiring large annotated datasets. Speech embeddings are extracted at the acoustic frame-level i.e. for short speech segments of 20 ms duration, and they can be used as input features to a model that is specific for the downstream task. These speech encoders have been successfully used in several tasks, such as automatic speech recognition [7], speaker verification [14, 13] and emotion recognition [40, 45]. Self-supervision learning for such speech encoders is designed to discover speech representations that encode pseudo-phonetic or phonotactic information rather than high-level semantic information [51]. On the other hand, high-level semantic information is particularly useful in some tasks such as Machine Translation (MT) or Spoken Language Understanding (SLU). In [32], the authors propose to address this issue using a new framework called SAMU-XLSR, which learns semantically-aligned multimodal utterance-level cross-lingual speech representations.

SAMU-XLSR is based on the pre-trained multilingual XLS-R [1] [6] on top of which all the embeddings generated by processing an audio file are connected to an attentive pooling module.

Thanks to this pooling mechanism (which is followed by linear projection layer and the *tanh* function), the frame-level contextual representations are transformed into a single utterance-level embedding vector. Figure 2 summarizes the training process of the SAMU-XLSR model. Notice than the weights from the pre-trained XLS-R model continue being updated during the process.

The utterance-level embedding vector of SAMU-XLSR is trained via knowledge distillation from

[1] https://huggingface.co/facebook/wav2vec2-xls-r-300m

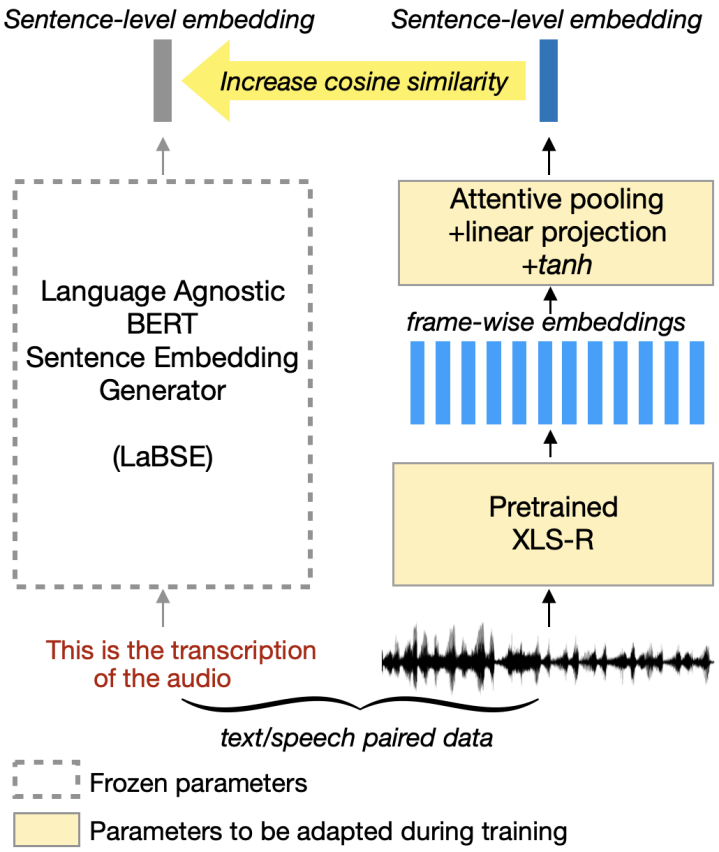


Figure 2: Training process of SAMU-XLSR.

the pre-trained language agnostic LaBSE model [22]. The LaBSE model[2] has been trained on 109 languages and its text embedding space is semantically aligned across these 109 languages. LaBSE attains state-of-the-art performance on various bi-text retrieval/mining tasks, while yielding promising zero-shot performance for languages not included in the training set (probably thanks to language similarities). Thus, given a spoken utterance, the parameters of SAMU-XLSR are trained to accurately predict a text embedding provided by the LaBSE text encoder of its corresponding transcript. Because LaBSE embedding space is semantically-aligned across various languages, the text transcript would be clustered together with its text translations.

By pulling the speech embedding towards the anchor embedding, cross-lingual speech-text alignments are automatically learned without ever seeing cross-lingual associations during training. This property is particularly interesting in the SLU context in order to port an existing model built on a well-resourced language to another language with zero or low resources for training.

4 Analysing the existing model

[2]https://huggingface.co/sentence-transformers/LaBSE

4.1 Study on the ability of SAMU-XLSR to extract semantic information in cross-lingual and cross-model scenarios in the context of a Spoken Language Understanding task

In this part of the work, we aim to examine the use of semantically-aligned speech representations for end-to-end spoken language understanding (SLU). Concretely, we employed the SAMU-XLSR model designed to generate a single embedding that captures the semantics at the utterance level, semantically aligned across different languages. We saw that the use of the SAMU-XLSR model instead of the initial XLS-R model improves significantly the performance in the framework of end-to-end SLU. We also proved the benefits of using this model towards language portability in SLU.

4.1.1 Motivations

As defined in [55] , *spoken language understanding is the interpretation of signs conveyed by a speech signal*. This interpretation refers to a semantic representation manageable by computers. Usually, this semantic representation is dedicated to an application domain that restricts the semantic field. With the massive deployment of voice assistants like Apple's Siri, Amazon Alexa, Google Assistant, etc. a lot of recent papers aim to process speech intent detection as an SLU task [27, 19, 39, 41, 1]. In such a task, only one speech intent is generally expected by sentence: the speech intent detection task could be considered as a classification task at the sentence-level and, in addition, the SLU model has to fill some expected slots corresponding to the detected intent.

SLU benchmarks related to task-driven human-machine spoken dialogue can be more or less complex, depending on the richness of the semantic representation. In this study, we focus on a hotel booking scenario through a telephone conversation, where the semantic representation is not related to speech intent detection, but based on a more complex ontology that derives from frames [8].

Our experimental work is carried out on the MEDIA SLU benchmark, described in section 4.1.2. We first expect to evaluate the performance of SAMU-XLSR used as a frame-wise feature extractor in comparison to the use of the initial XLS-R. Then we analyse the quality of the semantic encoding for each layer of the SAMU-XLSR and XLS-R model, to better understand the impact of the SAMU-XLSR training on the XLS-R model. We continue this investigation by fine-tuning the SAMU-XLSR and the XLS-R models on the downstream task. We also investigate the capability of SAMU-XLSR to transfer the semantic knowledge captured on French data to Italian data related to the same SLU task, thanks to the PortMEDIA corpus described in section 4.1.2. Last, we focus on the sentence-level embedding produced by the SAMU-XLSR model in order to measure the relevance of its semantic content to the target task, including in a language portability scenario and in a cross-modal setting.

4.1.2 Data

The French MEDIA benchmark [9] was created in 2002 as a part of a French governmental project named Technolangue. The *MEDIA Evaluation Package*[3] [4] is distributed by ELRA and freely accessible for academic research. Apart from the data itself, it defines a protocol

[3] http://catalog.elra.info/en-us/repository/browse/ELRA-E0024/

[4] International Standard Language Resource Number: 699-856-029-354-6

for evaluating SLU modules, with a task of semantic extraction from speech in a context of human-machine dialogues. 1258 official recorded dialogues were generated from around 250 speakers. Only the user's turns are semantically annotated with both semantic annotation and transcription. Table 2 presents the data distribution, in hours of speech and number of words, into the official training, development and test corpora.

The PortMEDIA corpus [5] was used in order to conduct experiments on language portability from French to Italian for SLU [36]. It has been produced on the same task as MEDIA, and follows the same specifications. It is made of 604 dialogues from more than 150 Italian speakers.

Table 2 presents the data distribution, in hours of speech and number of words, into the official training, development and test corpora. The PortMEDIA training corpus is more than four times smaller than the MEDIA one in terms of words, which makes it low resource. If speech duration seems not so low in comparison, this is due to a less precise speech segmentation that includes large portions of silence.

		Train	Dev	Test
Hours	**MEDIA**	10h52m	01h13m	03h01m
	PortMEDIA	07h18m	02h32m	04h51m
Words	**MEDIA**	94.5k	10.8k	26.6k
	PortMEDIA	21.7k	7.7k	14.7k

Table 2: Data distribution of the MEDIA and PortMEDIA corpus.

The "full" version of MEDIA and PortMEDIA has been used for all experiments. Around 150 different semantic concepts are used in this version. The following translated sentence is an example of utterance: "I would like to book one double room in Paris". We used annotations containing the transcript, the concepts and the location information of their values: "I (would like to book, *reservation*), (one, *room-number*), (double room, *room-type*) in (Paris, *city*)".

Historically, on the MEDIA corpus, two metrics are jointly used: the Concept Error Rate (CER) and the Concept-Value Error Rate (CVER). The CER is computed similarly to Word Error Rate (WER), by only taking into account the concepts occurrences in both the reference and the hypothesis files. The CVER metrics is an extension of the CER. It considers the correctness of the complete concept/value pair. Since our models generate transcript with semantic concepts, we also evaluate our systems in terms of Character Error Rate (ChER) and WER.

Both datasets are distributed by ELRA and freely accessible for academic research.

4.1.3 Layer-wise analysis of frame-level embeddings

Figure 3 presents the general architecture of the end-to-end model used. To make a layer-wise analysis, we removed the upper layers of each encoder, one by one, and extracted our speech embeddings. The encoder kept layers are frozen with their initial weights for the "Frozen" architecture, or fine-tuned by supervision to solve the MEDIA task, leading to the "Fine-Tuned" results.

[5] http://www.elra.info/en/projects/archived-projects/port-media/

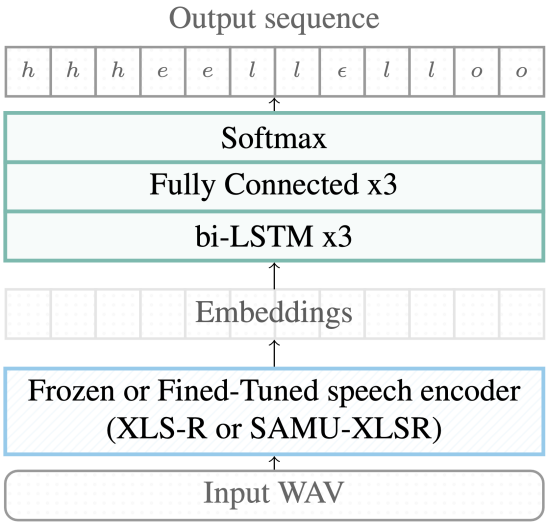


Figure 3: Neural Architecture for an SLU layer-wise analysis of speech encoders with the MEDIA dataset.

Figure 4 illustrates how the linguistic information is encoded through each layer of both encoders. First, we observed that in terms of WER, SAMU-XLSR gets better results than XLS-R. We also could see that the minimum WER is achieved with higher layers for SAMU-XLSR than it is for XLS-R, both frozen and fine-tuned. We assume this is due to the fine-tuning made on SAMU-XLSR by forcing its highest representations to be aggregated to LaBSE's text embeddings.

Figure 5 presents results measured with the Concept Error Rate metric, relevant to the specific semantics of the downstream task. We observed that the original frozen XLS-R model lost almost 7 points of CER between its best generated embeddings for semantic extraction task, layer 15, and its final generated embeddings, layer 24. On the other hand, since learning SAMU-XLSR consists on projecting its sentence-level embedding into the semantic multilingual LaBSE's encoding space, the highest layers of its encoder tend to capture and encode the semantics until the top layer. Both speech encoders give best CER results in middle layers.

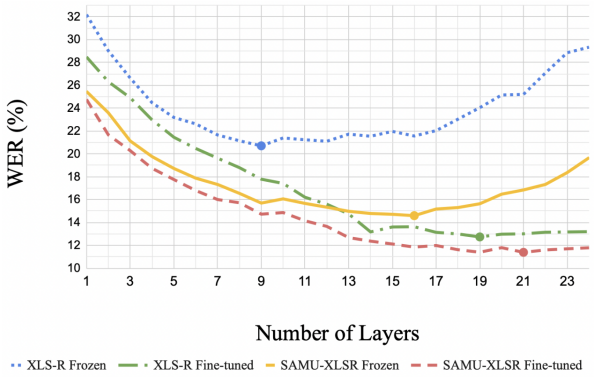


Figure 4: Layer-wise analysis of WER on the test data.

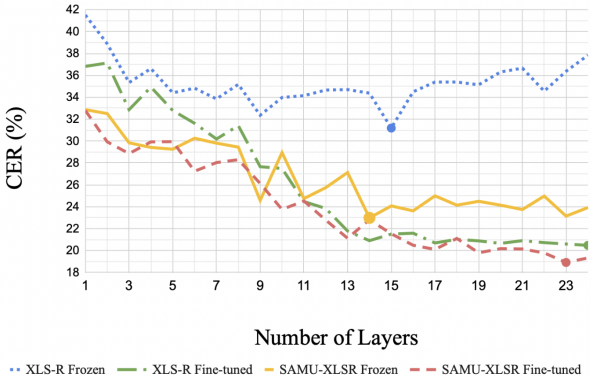


Figure 5: Layer-wise analysis of CER on the test data.

As expected, fine-tuning the speech encoders allows the models to extract as much semantic information as possible from the audio signal. Even if the semantics extracted from the frozen SAMU-XLSR middle layers were already mostly kept through upper layers, performances were

enhanced by fine-tuning the encoder.

4.1.4 Language portability

Zero-shot

To evaluate the multilingual portability of the SAMU-XLSR encoder compared to the original XLS-R, we applied zero-shot learning on our French (MEDIA) and Italian (PortMEDIA) data. We first trained each end-to-end SLU model on the French data, by freezing or fine-tuning the speech encoder, and then made a simple inference on the Italian data. We also aimed to measure how fine-tuning the speech encoder on the French data impacts the language portability capabilities.

		ChER	WER	CER	CVER
XLS-R	**Frozen**	68.77	129.08	88.24	100.44
	Fine-tuned	63.22	123.94	85.36	101.54
SAMU-XLSR	**Frozen**	49.35	100.13	54.62	99.83
	Fine-tuned	59.10	124.49	83.45	101.63

Table 3: Zero-shot results (%) from French MEDIA training to Italian PortMEDIA inference.

Results in Table 3 show that the use of a frozen SAMU-XLSR speech encoder gives strongly better performance than other setups for concept recognition in these conditions: a CER of 54.62% is attained while with the other configurations performs at more than 83% error rate. We noticed that, as expected, the performance related to the transcription itself is very bad: SAMU-XLSR is able to extract general semantics, but is not designed to provide language-dependent information useful to transcribe speech.

It also appears that fine-tuning SAMU-XLSR on French degrades the capability of the module to generate good semantic embeddings on Italian. It was not the case with the XLS-R speech encoder: a fine-tuning on same-family language enhanced the CER and other metrics of our Italian inference. We can deduct that the SAMU-XLSR forced its embeddings from different languages to be represented in the same space clusters, thanks to LaBSE. It allows the module to be a lot more efficient when dealing with multilingualism and language portability.

Low resource

Tables 4 and 5 illustrate the potential of portability of both encoders from French to Italian, with or without fine-tuning. We processed by training our model with frozen and fine-tuned XLS-R and SAMU-XLSR with PortMEDIA to have a baseline. Then, we made the same experiments on MEDIA for 100 epochs, before continuing the training on PortMEDIA for 100 epochs.

In both tables, IT means the SLU model has been trained from scratch on the Italian data. FR→IT means the SLU model weights have been initialized with the French model before being trained on the Italian data.

We can observe that using SAMU-XLSR as a speech encoder still outperforms XLS-R, with a CER of 33.01% (resp. 42.66% for XLS-R) without fine-tuning and without the use of French data. With both fine-tuning and use of French data, SAMU-XLSR is able to reach 26.18% of CER, but the gap with XLS-R – that reaches 26.92% – is less significant.

	Train Data	ChER	WER	CER	CVER
Frozen	IT	14.91	36.90	42.66	54.31
	FR→IT	12.78	32.41	35.39	49.60
Fine-tuned	IT	13.36	37.02	42.72	57.47
	FR→IT	7.55	20.01	26.92	40.11

Table 4: XLS-R PortMEDIA results (%) of PortMEDIA (IT) training and MEDIA training followed by PortMEDIA fine-tuning (FR→IT).

	Train Data	ChER	WER	CER	CVER
Frozen	IT	12.62	27.92	33.01	46.99
	FR→IT	11.01	25.09	26.90	42.70
Fine-tuned	IT	6.47	16.59	30.66	42.09
	FR→IT	7.04	17.81	26.18	39.28

Table 5: SAMU-XLSR PortMEDIA results (%) with PortMEDIA (IT) training and MEDIA training followed by PortMEDIA fine-tuning (FR→IT).

4.1.5 Semantic analysis of sentence-level embeddings

We then examined if the pooled sentence embeddings produced by SAMU-XLSR contain enough semantic information according to the MEDIA and PortMEDIA tasks, and we analyze their cross-modal and cross-lingual abilities.

We simplify the MEDIA and PortMEDIA benchmark tasks to a bag-of-concepts classification task. For each segment, the system has to predict all the concepts that are present in the speech segment. We use a multi-hot representation for the output. This simplification is necessary to be able to compare the model outputs with LaBSE's, which doesn't have frame-level embeddings. It also allows us to have a less complex model that does not require the order of concepts and, therefore, can do a more efficient knowledge transfer to new languages. Reminder that it is important for this SLU task to be able to estimate the values of the concepts and know the placement and number of times a concept appear in the transcription.

Figure 6 illustrates the architecture we implemented for this sentence-level analysis.

We applied L_2-normalisation on the embeddings. Fixing the norm (both in training and evaluation) is critical; unnormalized embeddings with large norm generate many false positives, while unnormalized embeddings with small norm generate many false negatives. Note also that the norm of the SAMU-XLSR embeddings may not be that informative, since the network is trained using cosine similarity between SAMU-XLSR and LaBSE embeddings, which is a norm-invariant objective function.

In order to test both the cross-modal and cross-lingual properties of the embeddings, we trained our classification models only on the French dataset. The Italian dataset is only used for testing, to obtain cross-lingual results. For the cross-modal properties, we trained a model on SAMU-XLSR (speech) embeddings, and tested it on both SAMU-XLSR and LaBSE (text) embeddings. We also trained a second model on LaBSE embeddings in order to observe the difference when testing it with SAMU-XLSR speech embeddings. The results, in terms of micro F_1-score are given in Table 6.

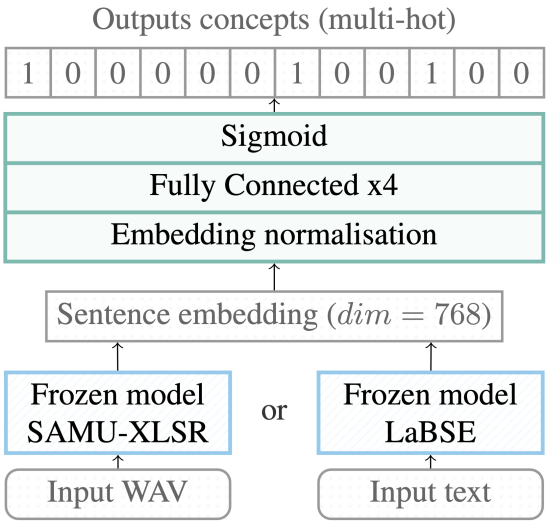


Figure 6: Neural Architecture for an SLU language portability analysis of speech encoders with the MEDIA and PortMEDIA datasets.

We also reported the frame-wise baseline results obtained in the previous experiments, by converting the sequence outputs of the models to a bag-of-concepts output.

Test Data	**Test Encoder**	**Train Encoder**	
		SAMU-XLSR	LaBSE
FR	SAMU-XLSR	77.52%	71.77%
	LaBSE	78.04%	82.15%
	*frame-wise**	*84.69%*	-
IT	SAMU-XLSR	68.55%	65.14%
	LaBSE	62.05%	69.58%
	*frame-wise**	*59.76%*	-

*Table 6: Micro F_1-scores for sentence-level semantic analysis with classification model trained on French and tested on French and Italian data. *Baseline results are obtained with the language portability models on frame-wise embeddings, and converted in bag-of-concepts outputs for evaluation.*

We observed that both LaBSE and SAMU-XLSR models obtained comparable results with their corresponding test embeddings when tested on the MEDIA dataset (77.52% for SAMU-XLSR, 82.15% for LaBSE). The capacity of SAMU-XLSR to reproduce sentence-level embeddings close to the LaBSE ones is noticeable, and validates the strategy used to train SAMU-XLSR.

By comparing the results from the previous experiments with the frame-wise embeddings and this sentence-level embedding analysis, we also observed that the sentence-level embeddings are better in extracting cross-lingual representations than the frame-level ones.

4.1.6 Conclusion

In this work, we investigated the capacity of the recently introduced SAMU-XLSR in addressing a challenging SLU task. SAMU-XLSR is a speech encoder is a fine-tuned version of the

XLS-R model, using LaBSE embeddings as targets. In addition to its promising performance, we demonstrated how this speech encoder differs from the XLS-R model in the way it encodes the semantic information in its intermediate hidden representations. We also showed the real potential of the SAMU-XLSR for language portability. We also showed its capacity to build a sentence-level embedding able to highlight the semantic information of the task and its promising cross-lingual and cross-modal properties.

It is important to notice that the capacities given to the SAMU-XSLR model come from out-of-domain data: no data related to the final semantic task were needed to train the model. This is really important since SLU tasks often suffer of being low resourced, especially in a multilingual scenario.

4.2 What audio information goes through SAMU-XLSR?

We propose to study and quantify audio information contained in the different layers composing a speech encoder. And compare layers by layers the information contained in SAMU-XLSR and XLS-R. In order to realise this study, a set of tasks has been identified, making it possible to probe the presence (or absence) of specific information in the speech encoder using several targeted classification tasks carried out with parameters extracted from several hidden layer. Our goal is to reveal the link between the features given by the speech encoder hidden layers and the tasks. The classification tasks are carried out, each time with parameters extracted from specific hidden layers of the speech encoder. High performance should then reveal important task-related characteristics contained in these layers, and vice versa. Figure 7 summarizes the protocol of our approach based on an speech encoder model architecture.

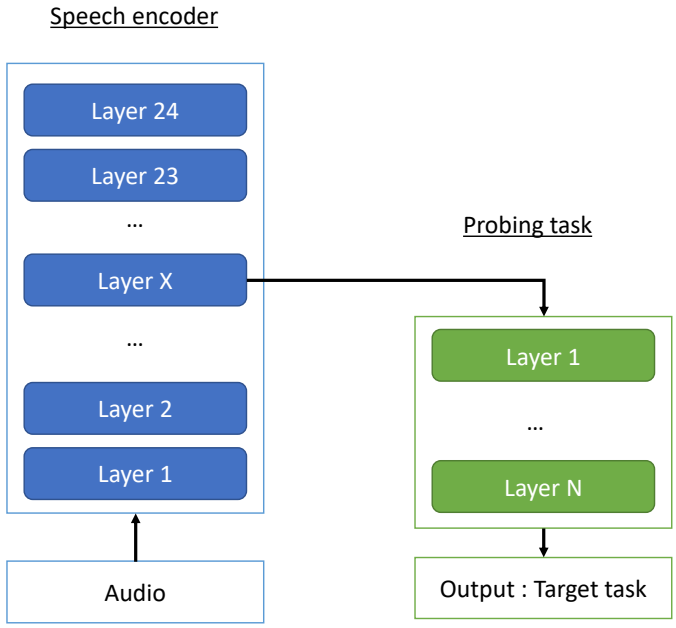


Figure 7: Proposed protocol for speech encoder information probing.

4.2.1 Probing tasks

We propose to probe the presence (or absence) of specific information in the speech encoder using several targeted classification tasks carried out with parameters extracted from several

hidden layers. Thus, if a pattern of these tasks is present in hidden layers of speech encoder, we can train a classifier to recognize it and the performance of the classifier should depend on how well the pattern is embedded in speech encoder.

For every classification tasks, we use an ECAPA-TDNN classifier [20]. The ECAPA-TDNN architecture uses cutting-edge techniques: Multilayer Feature Aggregation (MFA), Squeeze-Excitation (SE) and residual blocks. This model has recently shown impressive performance in the speaker verification [54, 53] and speaker diarization [18] domains. Except for the Automtic speech recognition task, we use an ASR system based on Kaldi.

The 3 different tasks studied for probing information contained in speech encoder are: speaker identification, automatic speech recognition and speech emotion.

- **Speaker identification** : Speaker identification aims to determine which registered speaker provides a given utterance from amongst a set of known speakers. This task is used to answer the question : “Who is speaking?”. The system has been trained on the VoxCeleb1 dataset [42], only on the development partition. Systems are evaluated on Voxceleb1 test dataset. We report the classification in terms of accuracy.
- **Automatic speech recognition** : Automatic speech recognition (ASR) aim to to transcribe spoken words into written text. The system has been train on Librispeech train dataset corpus [43] and we evaluate on . We report the results in terms of Word Error Rate (WER).
- **Emotion recognition** : Speech Emotion Recognition (SER) aim to classify speech records in seven-classes (anger, disgust, fear, joy, neutral, sadness and surprise). The system has been train on Multimodal EmotionLines Dataset (MELD) corpus [47]. This corpus is composed of 13,000 utterances from Friends TV (sitcom). We report the classification in terms of F1-score.

4.2.2 Results

Speaker identification

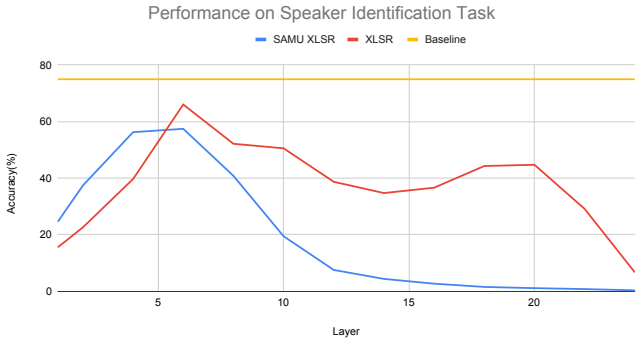


Figure 8: Performance obtained for speaker identification task at different speech encoder layer levels.

Figure 8 summarizes the performance given by the speaker identification task. Scores are expressed in terms of accuracy. We compare the performance obtained by different speech encoder (SAMU-XLSR and XLSR) and baseline system (Filter-bank) at different layers.

Globally, speech encoders (SAMU-XLSR or XLSR) obtain lower performance than baseline

system (Filter-bank). These results tends to say that speech encoders remove speaker information (speaker-specific traits).

We can observe that the best performance of SAMU-XLS and XLSR are obtained by using the sixth hidden layer (they obtained respectively 66% and 55% of accuracy). The baseline system obtained 75% of accuracy. We can observe that by using hidden layers higher than 6, the performance deteriorates.

Automatic speech recognition

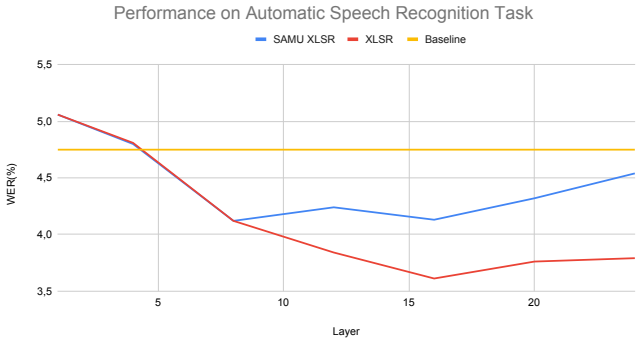


Figure 9: Performance obtained for automatic speech recognition task at different speech encoder layer levels.

Figure 9 summarizes the performance given by the automatic speech recognition task. Scores are expressed in termes of WER. We compare the performance obtained by different speech encoder (SAMU-XLSR and XLSR) and baseline system (Filter-bank) at different layers.

Globally, speech encoders (SAMU-XLSR or XLSR) obtained better results than baseline system. The best performance are obtained for SAMU-XLSR and XLSR by using the sixteenth hidden layer (they obtained respectively 4.13% and 3.61% of WER). The baseline system obtained 4.75% of WER. Speech encoder can better convey phonemes information.

Emotion recognition

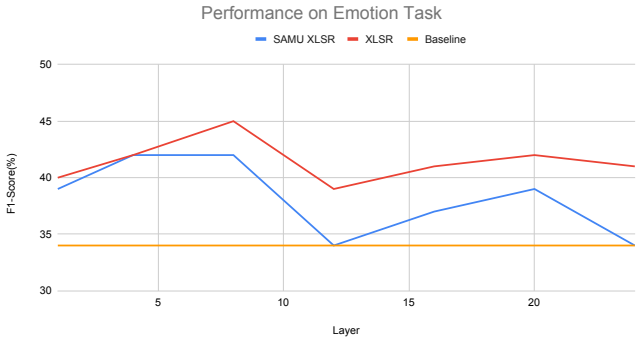


Figure 10: Performance obtained for emotion task at different speech encoder layer levels.

Figure 10 summarizes the performance given by emotion recognition task. Scores are expressed in terms of F1-Score.

Globally, speech encoders (SAMU-XLSR or XLSR) obtained better results than baseline system (Filter-bank). The best performance are obtained for SAMU-XLSR and XLSR by using the eighth hidden layer (they obtained respectively 42% and 45% of F1-Score).

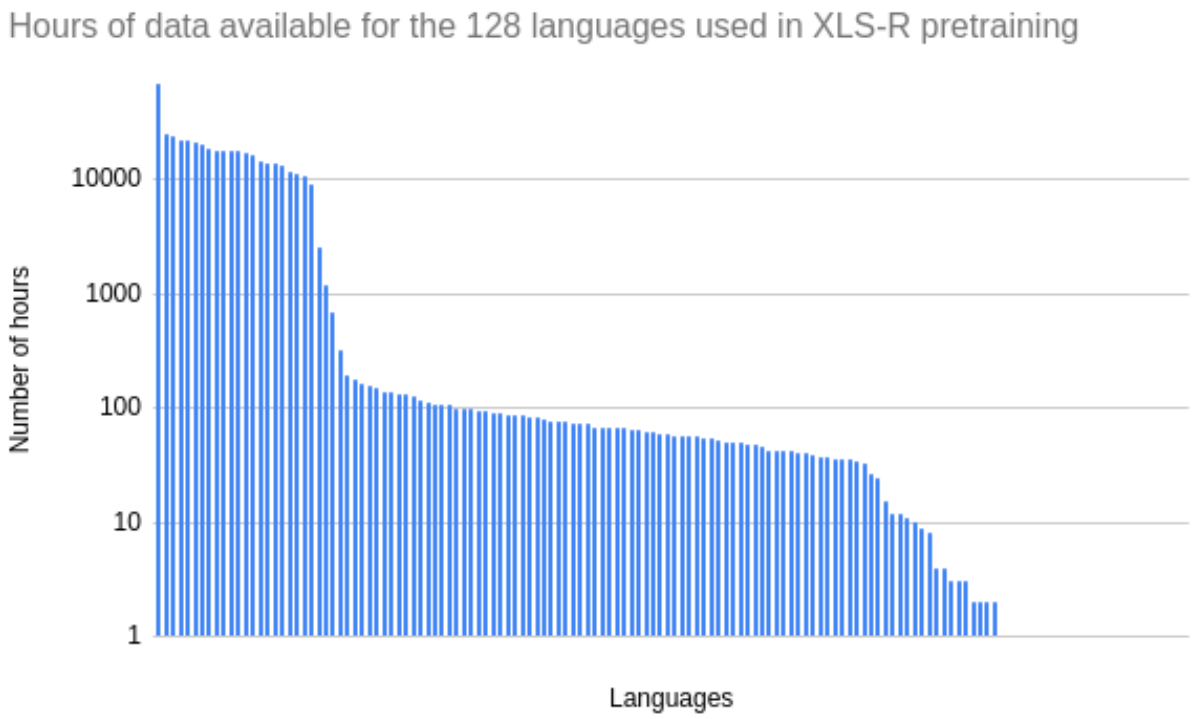


Figure 11: Bar plot of the number of hours available for the pre-training of XLS-R.

4.3 Conclusions

We proposed a protocol that aims at highlighting audio information contained in speech encoder. By analyzing the performance obtained by each specific task in the different hidden layers of speech encoder, we were able to realize the information contained at various levels of the speech encoder, whether at the speaker, at the phonetic, or at the emotion information. We observe that speech encoder provide phoneme and emotion information. The information provided by speech encoders are much better than Filter-banks or MFCC. But unfortunately speaker information are suppressed in the speech encoder. And if we want to use speaker information in a machine translation system we will have to find an another way to convey this information.

4.4 Does SAMU-XLSR perform well for all language families?

4.4.1 A language-family wise approach on SAMU-XLSR performances

SAMU-XLSR [32] is a cross-lingual speech encoder, fine-tuned from a base XLS-R [6] model to maximise the cosine similarity of its embeddings to the ones produced by the cross lingual text encoder LaBSE [22]. Those 3 models were respectively trained on sets of 51, 128 and 101 languages, with respectively 8 327 and 435 482 hours of speech and 23 014 061 000 sentences. However, as shown in the graph 11, the amount of data available for each language is far from balanced for each of those sets.

Cross lingual models trained on massive amount of data usually get better results on lower resources languages than those only trained on one language [31]. Those models use knowledge transfer across the different languages to leverage knowledge from higher resource to

improve performances on lower resource languages. The structures of languages are not uniform across the world, so we looked at a way to sort languages. Ethnologue [10] proposed a classification of languages in Families, Sub-families and other sub-divisions. We used that classification on the languages used for the pre-training of the cross lingual models. The amount of data available by Family and Sub-Family is presented in the table 7.

As shown in the table 7, there is high variation in the quantity of training data available across all models. However, a higher amount of data does not mean the system has been trained more on that language, as the re-balancing samplers used in pre-training use a parameterised upsampling algorithm to balance the datasets. This can lead to repetition of samples for low-resource languages, and so a higher amount of data means that the systems will have seen more data variation from one language than from another.

To improve the performances for different sub-families, we propose to train different version of SAMU-XLSR only on the data from a given family, to improve the performances on downstream tasks applied on this family.

4.4.2 Training sub-families versions of SAMU-XLSR

We decided to train different versions of SAMU-XLSR [32] for different subfamilies included in its base training set. The 7 subsets chosen are presented in the table 8.

For each of those subsets, we started from a SAMU-XLSR [32] model pretrained from CommonVoice v8.0 [4] dataset, and trained it further using the exact same protocol, only with smaller set of data. As a comparison, the total amount of speech used for the initial pretraining is 8327 hours. Once our seven specialized models trained, we evaluated them on multiple downstream tasks. Each model was trained on 4 V100 GPU cards for 20hours. Given the variable amount of data used, them number of epochs varies from one model to another, the exact numbers being provided in table 9

4.4.3 Application on downstream tasks

To evaluate our model, we choose three downstream tasks on which we fine-tuned it, to measure the improvement of each specialized model from the base SAMU-XLSR.

SLU task We performed the same french SLU task presented in the subsection 4.1 on each of the specialized model. This task consist of fine-tuning a given pre-trained model on the train subset of the french MEDIA dataset [9] to predict concepts. The results in Concept Error Rate (**CoER**) are presented in the table 10 for every specialized model, as well as for the original SAMU-XLSR model, measured on the eval subset of the french MEDIA dataset. The train/eval split used here is described in [9], where they are respectively described as the "*Adaptation set*" and the "*Test set*".

As we can see in the 10 table, we got no significant improvement on the different families, except for the Indo-Iranian model. All the improvements are in the confidence interval, so we can not conclude on any progress here.

Translation task We decided to evaluated the different models performances on a Speech to Text translation task. As explained in the section 5.5.3, the Tamasheq to French translation

Table 7: Table of the amount of data available for XLS-R, SAMU-XLSR and LaBSE by languages Family and Sub-Family. LaBSE amounts are in thousands of text sentences, XLS-R and SAMU-XLSR amounts are in hours of speech.

Family	Sub-Family	XLS-R	SAMU-XLSR	LaBSE
Abkhaz-Adyge	Abkhaz-Abaza	1	0	0
Afro-Asiatic	Berber	0	50	0
	Chadic	75	3	8458
	Cushitic	82	0	11805
	Semitic	9357	93	303167
Artificial Language	Esperanto	97	1407	74630
Atlantic-Congo	North-Central Atlantic	0	0	1782
Atlantic-Congo	Volta-Congo	15918	287	236985
Austroasiatic	Khmeric	0	0	29773
	Vietic	131	4	220946
Austronesian	Malayo-Polynesian	529	25	462141
Basque	Basque	113	98	52689
Classical Indo-European	Paleo-Balkan	56	0	89268
	Armenic	55	1	73501
	Balto-Slavic	96451	1366	3610606
	Celtic	206	120	100604
	Germanic	145247	3440	9650074
	Graeco-Phrygian	17761	15	204252
	Indo-Iranian	1199	413	824158
	Italic	104445	2593	2585127
Dravidian	South Dravidian	254	217	334132
Hmong-Mien	Hmongic	0	0	4507
Japonic	Japanesic	49	40	1410416
Kartvelian	Georgian-Zan	127	7	68016
Koreanic	Koreanic	61	0	211037
Mongolic-Khitan	Mongolic	68	12	45520
Sino-Tibetan	Bodic	81	0	449
	Burmo-Qiangic	33	0	42850
	Sinitic	325	229	1274737
Tai-Kadai	Kam-Tai	150	142	125215
Tupian	Maweti-Guarani	2	0	0
Turkic	Bolgar	4	0	0
	Common Turkic	550	278	482992
Uralic	Finnic	24634	40	285576
	Finno-Ugric	17421	19	188648

Family	Subset	Languages	Hours	Files
Indo European	Italic	Catalan, French, Spanish, Italian, Portuguese, Romanian, Galician	2593	648 450
	Germanic	English, German, Dutch, Swedish, Danish	3440	534 649
	Balto-Slavic	Belarusian, Russian, Polish, Ukrainian, Czech, Latvian, Slovakian, Bulgarian	1366	1 487 068
	Indo-Iranian	Persian, Hindi	413	318 517
Sino-Tibetan	Chinese	Chinese (HK & CN)	325	465 493
Atlantic-Congo	Volta-Congo	Kinyarwanda, Swahili	287	110 843
Afro-Asiatic	Berber	Kabyle	93	158 157

Table 8: Table of languages used in the different subsets of training, and the total number of hours of speech available for each subset and the number of files.

Model	Italic	Germanic	Balto-Slavic	Volta-Congo	Chinese	Berber	Indo-Iranian
Epochs	22	20	12	14	10	36	3

Table 9: Number of epochs of training for each model.

Model	CoER
Base	20.2%
Balto-Slavic	20.4%
Berber	20.8%
Chinese	20.5%
Germanic	20.4%
Indo-Iranian	**19.7**%
Italic*	20.1%
Volta-Congo	*19.9*%

Table 10: CoER for concept extraction on different models, all fine-tuned then evaluated respectively on french MEDIA dataset train and eval subsets. The CoER were computed with a confidence interval of $\pm$ 0.8.

was one of the targets of our team during the workshop. We focused at first on this task, fine-tuning SAMU-XLSR and different versions of the models on this translation task with the data of IWSLT 2022 (see section 5.5.3), then we computed the BLEU score [44] over the translations generated. The results for tamasheq to french are presented in the table 11.

From the table 11, we can see that no model outperformed the base model on the translation task, but from the different specialized versions, the Indo-Iranian is still the best.

Following the end of the workshop, we plan to measure the performances of those models

Model	BLEU
Base	**11.56**%
Balto-Slavic	10.47%
Berber	10.29%
Chinese	9.58%
Germanic	9.86%
Indo-Iranian	*10.9*%
Italic*	10.8%

Table 11: BLEU scores over a Tamasheq to French translation task, evaluated on the IWSLT2022 dataset on various SAMU-XLSR versions.

on speech to text translation from 11 languages (French (Fr), German (De), Dutch (Nl), Russian(Ru), Spanish (Es), Italian (It), Turkish (Tr), Persian (Fa), Swedish (Sv), Mongolian (Mn) and Chinese (Zh)) to English (En), using the CoVoST [56] dataset.

ASR task Another task is planned : Automatic Speech Recognition over 61 languages of CommonVoice v8.0 [4]. We have chosen the languages that contained more than 1000 utterances in their evaluation set, and computed their embeddings for the seven specialized version of SAMU-XLSR plus the base version. The complete list of those languages using the ISO 639-1 code is : ar, be, bg, ca, cs, cy, da, de, el, en, eo, es, et, eu, fa, fi, fr, fy-NL, ga-IE, gl, hi, hu, id, it, ja, ka, km, ky, lt, lv, mn, mt, nl, pl, pt, ro, ru, rw, sk, sl, sv-SE, sw, ta, th, tr, tt, ug, uk, uz, zh-CN, zh-HK, ab, ba, br, ckb, cv, dv, ia, kab, lg, rm-sursilv, rah, zh-TW.

The train set for each of those languages was used to train a different ASR system [] for each of the models . Then each model was evaluated on the first 1000 utterances of each evaluation set.

The results are not accessible yet because of the computation time, but will be at some point.

4.4.4 Comparing models using cosine similarity on embeddings

In the order to get access to a better analysis of the system for a minimal computation time, we propose a method of analysis directly on the embeddings produced. This method is applied on our test test : the speech utterances and associated transcriptions contained in the CommonVoice v8.0 [4] evaluation subset described in the previous section : 1000 utterance of speech for 61 languages.

The models XLS-R [6] is a cross-lingual model taking *speech* utterances and generating *sequences* of embeddings of variable length. LaBSE [22] is a cross-lingual model taking *text* utterances and generating *sentence* embeddings. SAMU-XLSR [32] is a version of XLS-R that has been trained further, and has been added an attentive pooling layer that allows it to produce *sentence* embeddings as well as *sequences* of embeddings from speech utterances. Our base SAMU-XLSR version (*base*) and family flavoured versions (*balto*, *berber*, *chinese*, *germanic*, *indo*, *italic*, *volta*) will be characterized by the v variable.

Computing the embeddings Given S_n^L the $n^{th} \in [0, 999]$ utterance of speech from a language L in the test set, and T_n^L its associated transcription, we use :

- a pretrain version of XLS-R to compute the sequence of embeddings of length l_n^L :

  $XLSR_n^L = (x_{n,i}^L \in \mathbb{R}^{1024})_{i \in [1, l_n^L]}$ from S_n^L.

- a pretrain version of LaBSE to compute the associated sentence embedding :

  $labse_n^L \in \mathbb{R}^{768}$ from T_n^L.

- our base version of SAMU-XLSR and different family versions to compute the associated sequence of embeddings of length l_n^L :

  ${}_vSAMU_n^L = ({}_vy_{n,i}^L \in \mathbb{R}^{1024})_{i \in [1, l_n^L]}$

  and the pooled embedding:

  ${}_vsamu_n^L \in \mathbb{R}^{768}$ both from S_n^L.

Computing the cosine similarities Then for a given utterance (S_n^L, T_n^L), we compute two cosine similarities (using the cos function defined as $cos : x, y \mapsto \frac{x \cdot y}{||x|| \cdot ||y||}$):

- The cosine similarity between sentence embeddings, for a version v of SAMU-XLSR :

  ${}_vd_n^L = cos(labse_n^L, {}_vsamu_n^L)$.

- The average of the element-wise cosine similarities computed between elements of the sequence of embeddings of length l_n^L from XLS-R and from a given version v of SAMU-XLSR :

  ${}_vD_n^L = \frac{1}{l_n^L} \sum_{i=1}^{l_n^L} cos(x_{n,i}^L, {}_vy_{n,i}^L) \mid (x_{n,i}^L, {}_vy_{n,i}^L) \in XLSR_n^L \times {}_vSAMU_n^L$

For a given language and a given version of SAMU-XLSR, we use the average cosine similarities for all the utterances.

- For a comparison with LaBSE : ${}_vd^L = \frac{1}{1000} \sum_{n=0}^{999} {}_vd_n^L$.
- For a comparison with XLS-R : ${}_vD^L = \frac{1}{1000} \sum_{n=0}^{999} {}_vD_n^L$.

The first metric gives us the mean similarity between LaBSE and trained SAMU-XLSR, so we can measure how well the latest fitted its embedding to LaBSE for each language. The second gives us the same for XLS-R and SAMU-XLSR, so we can measure how the embeddings were moved during training. However, for the second, further training on the model can change the order of the embeddings in the sequence, giving poor frame-wise similarities.

The results The cosine similarities scores measured for each version and each language are presented in the table 14 and 15 for the sequences comparison and 12 and 13 for the sentence embeddings.

Those tables are difficult to read, so we decided to compute the mean and standard deviation of the similarity for each model, for the languages in and out of their respective training sets. Those in-domain and out-domain similarities are available in the graph 12.

The left graph in the figure 12 shows three main points about the LaBSE embeddings :

1. All the dots are under the $x = y$ line, so SAMU-XLSR embeddings are always closer to LaBSE for the seen languages.

	base	*indo*	*germanic*	*italic*	*balto*	*volta*	*berber*	*chinese*
ab	0.236	0.219	0.220	0.199	0.239	0.213	0.408	0.204
ar	0.859	0.808	0.532	0.595	0.660	0.610	0.172	0.352
ba	0.491	0.429	0.279	0.267	0.372	0.287	0.356	0.242
be	0.914	0.816	0.634	0.604	0.915	0.494	0.158	0.398
br	0.395	0.380	0.378	0.379	0.384	0.349	0.391	0.303
ca	0.917	0.822	0.758	0.915	0.765	0.611	0.254	0.460
ckb	0.316	0.349	0.245	0.280	0.290	0.310	0.339	0.249
cs	0.766	0.617	0.513	0.517	0.884	0.429	0.168	0.329
cv	0.222	0.236	0.169	0.168	0.240	0.201	0.314	0.182
cy	0.772	0.606	0.467	0.428	0.472	0.394	0.185	0.265
da	0.487	0.458	0.656	0.419	0.449	0.381	0.137	0.294
de	0.920	0.854	0.916	0.776	0.774	0.650	0.216	0.474
dv	0.056	0.060	0.035	0.034	0.046	0.093	0.122	0.061
el	0.431	0.384	0.374	0.389	0.391	0.320	0.120	0.277
en	0.904	0.857	0.872	0.822	0.804	0.772	0.303	0.655
eo	0.941	0.881	0.789	0.827	0.867	0.734	0.257	0.434
es	0.934	0.879	0.815	0.930	0.840	0.720	0.286	0.527
et	0.717	0.511	0.488	0.449	0.532	0.346	0.197	0.302
eu	0.758	0.603	0.479	0.466	0.496	0.331	0.201	0.319
fa	0.929	0.938	0.694	0.699	0.801	0.580	0.133	0.478
fi	0.396	0.347	0.383	0.346	0.374	0.279	0.150	0.220
fr	0.928	0.894	0.841	0.917	0.841	0.751	0.286	0.607
fy-NL	0.805	0.685	0.803	0.574	0.611	0.458	0.232	0.359
gl	0.910	0.846	0.791	0.928	0.808	0.665	0.245	0.504
hi	0.553	0.763	0.453	0.462	0.488	0.416	0.210	0.296
hu	0.451	0.390	0.414	0.403	0.439	0.319	0.178	0.267
ia	0.793	0.673	0.616	0.815	0.641	0.497	0.321	0.375
id	0.777	0.639	0.488	0.486	0.525	0.394	0.177	0.247
it	0.917	0.831	0.771	0.916	0.803	0.657	0.276	0.485
ja	0.423	0.394	0.336	0.295	0.328	0.258	0.106	0.248
ka	0.491	0.441	0.479	0.450	0.491	0.380	0.207	0.347
kab	0.242	0.226	0.222	0.230	0.246	0.233	0.881	0.210
kmr	0.642	0.701	0.378	0.391	0.436	0.340	0.165	0.289
ky	0.834	0.668	0.469	0.424	0.569	0.386	0.142	0.301
lg	0.182	0.160	0.158	0.154	0.168	0.285	0.248	0.099

Table 12: Cosine similarities averages by language between LaBSE and SAMU-XLSR sentence embeddings, Part. 1.

	base	*indo*	*germanic*	*italic*	*balto*	*volta*	*berber*	*chinese*
lt	0.559	0.492	0.516	0.492	0.777	0.394	0.208	0.381
lv	0.482	0.473	0.440	0.446	0.617	0.378	0.196	0.331
mn	0.512	0.367	0.304	0.291	0.314	0.249	0.137	0.216
mt	0.584	0.536	0.524	0.547	0.545	0.448	0.185	0.346
nl	0.878	0.776	0.947	0.661	0.704	0.539	0.167	0.393
pl	0.891	0.750	0.620	0.602	0.892	0.498	0.146	0.378
pt	0.901	0.791	0.683	0.930	0.742	0.575	0.199	0.403
rm-sursilv	0.491	0.446	0.426	0.495	0.423	0.379	0.290	0.332
ro	0.796	0.688	0.627	0.936	0.684	0.511	0.141	0.406
ru	0.933	0.845	0.646	0.638	0.946	0.548	0.158	0.426
rw	0.459	0.353	0.292	0.321	0.316	0.750	0.081	0.202
sah	0.246	0.283	0.119	0.131	0.156	0.240	0.238	0.144
sk	0.798	0.726	0.554	0.588	0.940	0.576	0.209	0.434
sl	0.661	0.588	0.509	0.504	0.914	0.368	0.195	0.325
sv-SE	0.664	0.569	0.926	0.521	0.549	0.433	0.189	0.344
sw	0.498	0.456	0.372	0.363	0.381	0.945	0.143	0.250
ta	0.675	0.513	0.376	0.342	0.354	0.235	0.151	0.216
th	0.915	0.799	0.561	0.519	0.628	0.394	0.129	0.401
tr	0.818	0.720	0.519	0.498	0.581	0.457	0.182	0.384
tt	0.755	0.602	0.365	0.351	0.480	0.342	0.277	0.282
ug	0.906	0.836	0.455	0.425	0.601	0.443	0.186	0.355
uk	0.832	0.717	0.548	0.542	0.878	0.473	0.181	0.377
uz	0.823	0.766	0.403	0.395	0.553	0.422	0.154	0.333
zh-CN	0.781	0.644	0.397	0.379	0.481	0.303	0.145	0.880
zh-HK	0.954	0.899	0.531	0.534	0.687	0.464	0.126	0.973
zh-TW	0.558	0.438	0.227	0.252	0.290	0.230	0.061	0.727

Table 13: Cosine similarities averages by language between LaBSE and SAMU-XLSR sentence embeddings, Part. 2.

2. All the specialized versions have their points to the right of the base one, so more training on a subset means you get closer to the languages of this family.
3. The specialized versions have different distances to the out-domain families, meaning you can get a better or worst generalisation depending on the subset on which it was fine-tuned.

   It can be noted that the best is the Indo-Iranian here (which got the best performances in the previous sections) and the worst is the Berber (which was one of the worst in the previous sections).

The right graph in the figure 12 shows three main points about the XLSR embeddings :

	base	*indo*	*germanic*	*italic*	*balto*	*volta*	*berber*	*chinese*
ab	0.018	0.037	0.011	0.014	0.015	0.008	0.004	0.019
ar	0.017	0.031	0.020	0.018	0.021	0.018	0.005	0.025
ba	0.026	0.042	0.032	0.031	0.028	0.027	0.012	0.035
be	0.022	0.037	0.026	0.027	0.020	0.022	0.002	0.026
br	0.042	0.063	0.038	0.037	0.044	0.045	0.031	0.042
ca	0.020	0.033	0.023	0.021	0.022	0.022	0.009	0.023
ckb	0.033	0.047	0.027	0.028	0.034	0.030	0.015	0.034
cs	0.029	0.047	0.029	0.033	0.025	0.031	0.006	0.027
cv	0.015	0.030	0.015	0.016	0.012	0.008	-0.004	0.018
cy	0.020	0.039	0.018	0.019	0.023	0.021	0.004	0.014
da	0.022	0.047	0.022	0.026	0.029	0.027	0.008	0.023
de	0.019	0.032	0.017	0.023	0.020	0.021	0.006	0.015
dv	0.026	0.041	0.021	0.023	0.025	0.017	-0.001	0.018
el	0.032	0.051	0.029	0.032	0.031	0.028	0.013	0.033
en	0.012	0.026	0.012	0.015	0.016	0.020	0.011	0.014
eo	0.017	0.027	0.017	0.019	0.017	0.017	0.005	0.012
es	0.019	0.032	0.021	0.021	0.021	0.021	0.005	0.019
et	0.019	0.032	0.021	0.019	0.018	0.010	-0.002	0.012
eu	0.019	0.035	0.018	0.020	0.018	0.010	-0.006	0.020
fa	0.026	0.037	0.030	0.030	0.028	0.028	0.013	0.032
fi	0.033	0.048	0.027	0.027	0.030	0.023	-0.000	0.017
fr	0.018	0.030	0.020	0.019	0.020	0.023	0.009	0.021
fy-NL	0.021	0.040	0.022	0.024	0.024	0.028	0.014	0.021
gl	0.016	0.028	0.020	0.018	0.019	0.018	0.000	0.019
hi	0.027	0.033	0.021	0.025	0.025	0.021	0.007	0.018
hu	0.031	0.045	0.026	0.026	0.028	0.021	0.006	0.020
ia	0.026	0.043	0.024	0.027	0.025	0.024	0.018	0.027
id	0.030	0.042	0.027	0.027	0.030	0.025	0.001	0.020
it	0.018	0.031	0.021	0.020	0.019	0.019	0.004	0.018
ja	0.037	0.048	0.026	0.028	0.031	0.023	0.000	0.023
ka	0.018	0.034	0.018	0.019	0.020	0.013	-0.003	0.016
kab	0.052	0.069	0.043	0.046	0.053	0.048	0.020	0.050
kmr	0.033	0.046	0.032	0.035	0.035	0.031	0.016	0.038
ky	0.022	0.035	0.029	0.033	0.025	0.023	-0.003	0.026
lg	0.023	0.037	0.014	0.020	0.022	0.012	-0.004	0.016

Table 14: Cosine similarities averages by language between XLS-R and SAMU-XLSR sequences of embeddings, Part. 1.

	base	*indo*	*germanic*	*italic*	*balto*	*volta*	*berber*	*chinese*
lt	0.021	0.038	0.021	0.024	0.019	0.016	0.002	0.020
lv	0.045	0.063	0.038	0.040	0.041	0.045	0.023	0.041
mn	0.023	0.034	0.024	0.025	0.027	0.020	-0.004	0.019
mt	0.026	0.044	0.025	0.026	0.027	0.027	0.006	0.024
nl	0.018	0.034	0.018	0.022	0.022	0.025	0.006	0.018
pl	0.023	0.037	0.027	0.027	0.020	0.026	0.008	0.029
pt	0.025	0.042	0.028	0.025	0.028	0.029	0.013	0.030
rm-sursilv	0.017	0.035	0.018	0.021	0.021	0.018	0.007	0.016
ro	0.025	0.041	0.026	0.022	0.028	0.026	-0.000	0.022
ru	0.019	0.031	0.023	0.024	0.016	0.020	-0.001	0.022
rw	0.031	0.044	0.025	0.027	0.027	0.024	0.002	0.026
sah	0.018	0.032	0.022	0.015	0.015	0.007	-0.010	0.015
sk	0.035	0.056	0.035	0.035	0.033	0.038	0.033	0.040
sl	0.032	0.048	0.029	0.032	0.026	0.032	0.013	0.031
sv-SE	0.031	0.053	0.025	0.033	0.035	0.037	0.014	0.031
sw	0.013	0.029	0.015	0.017	0.016	0.018	-0.010	0.016
ta	0.015	0.019	0.015	0.015	0.017	0.005	-0.014	0.010
th	0.023	0.039	0.028	0.029	0.032	0.031	0.017	0.017
tr	0.038	0.053	0.046	0.049	0.043	0.043	0.020	0.047
tt	0.024	0.037	0.032	0.031	0.027	0.029	0.005	0.035
ug	0.013	0.024	0.021	0.022	0.015	0.017	0.000	0.020
uk	0.021	0.036	0.024	0.026	0.018	0.023	0.007	0.028
uz	0.018	0.031	0.025	0.026	0.021	0.020	0.003	0.026
zh-CN	0.017	0.037	0.015	0.022	0.024	0.019	0.009	0.010
zh-HK	0.017	0.036	0.020	0.021	0.021	0.026	0.018	0.014
zh-TW	0.037	0.061	0.036	0.042	0.051	0.046	0.022	0.025

Table 15: Cosine similarities averages by language between XLS-R and SAMU-XLSR sequences of embeddings, Part. 2.

1. Almost all the dots are over the $x = y$ line, so SAMU-XLSR embeddings are always further to XLSR for the seen languages.

   With the exception of the Berber one, that got further for out-domain than for in-domain.

2. Almost all the specialized versions have lower similarities than the base one, so more training means you get further from the XLS-R model.

   With the exception of the Indo-Iranian one, that managed to get closer to XLS-R during its pretraining.

The results obtained on this cosine analysis were mostly expected, but we can underline that they are coherent with the results obtained on downstream tasks earlier, and that it might be

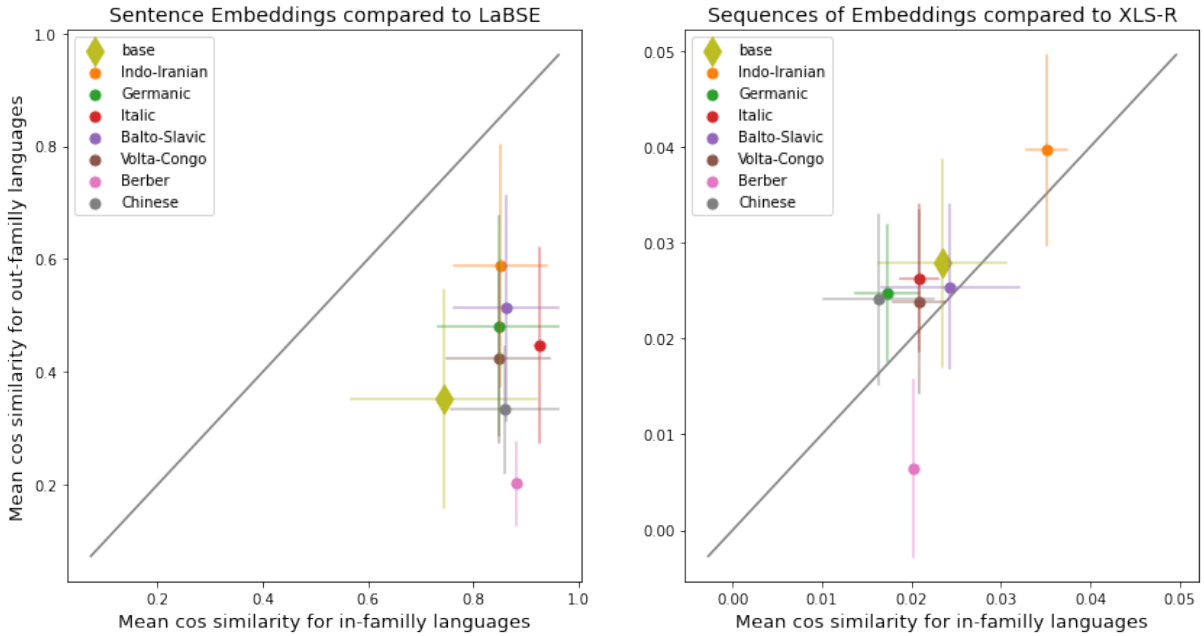


Figure 12: Average cosine similarity between embeddings, for In-domain languages and Out-domain languages. The base model is represented by a diamond, and the deviation bars are represented by vertical and horizontal bars. The line $x = y$ is plotted, to better visualize the models more similar for in-domain languages than out-domain languages. The left plot is the sentence embeddings similarities, the right plot is the sequences of embeddings similarities.

interesting to force a certain similarity of the embeddings with the root model XLS-R when training, to get eventually the best results, as shown with the examples of the Indo-Iranian and Berber models.

4.5 What speech can we generate from existing representations?

TODO: "resp: Thibault"

One of our goals is to generate speech corresponding to an input, whether this input is speech or text. We used some Speech Generation systems to do some experiments about speech synthesis from different pretrained embedding spaces, in order to get an idea of how well we can expect to generate speech from our common space. To do so, we trained some TTS systems and vocoders to generate speech from these embedding spaces.

4.5.1 Datasets for Speech Synthesis

To perform speech generation, we used different datasets to train our systems.

1. LJSpeech : Monospeaker read english audiobooks
2. SynPaFlex : Monospeaker read french audiobooks
3. VCTK : Multispeaker read english newspaper
4. CommonVoice : Multispeaker read multilingual sentences

5. CVSS : Multispeaker sentences in different languages from CommonVoice synthetized using TTS systems

4.5.2 Description of the systems used

Tacotron2 [52] is a common Text-to-Speech system that usually produces mel-spectrogram from a sequence of characters. It is used with a vocoder, that generates raw audio according to the generated mel-spectrogram. For these experiments, we trained Tacotron2 to generate mel-spectrograms from the different sets of embeddings. Then, we used Waveglow [48] as a vocoder, which had been previously trained on LJSpeech dataset, to produce raw audio from those mel-spectrograms.
HifiGAN [33] is a vocoder, which generates a waveform from a spectrogram using Generative Adversarial Networks, but we used it to generate raw audio directly from quantized representations extracted from embedding spaces.

4.5.3 Embedding spaces used for synthesis

We performed speech generation from different pretrained embeddings, which are coming either from text representations or from raw audio. We used as representations coming from text:

- Raw text (Usual Text-to-Speech)
- LabSE embeddings (Multilingual, one sentence-level embedding or token-level sequence of embeddings)
- SpeechT5 textual embeddings (Multimodal system, english only)
- XLM-R embeddings (Cross-lingual representation of text)

Coming from speech, we used the following representations:

- WavLM embeddings (Audio english only, frame-level sequence of embeddings)
- SAMU-XLSR embeddings (Sentence-level embedding and frame-level sequence)
- SpeechT5 speech embeddings (Multimodal system, english only, is supposed to match those coming from text
- XLS-R embeddings (Cross-lingual representation of speech)

We mostly focused on English language at first, since some of these systems have only been trained with English data. We used LJSpeech dataset to generate speech with a single speaker.

5 Improving over the initial model

In this section we describe the work done in order to build a multi-modal and multi-lingual encoder-decoder. Several approaches have been investigated either building on top of the SAMU-XLSR encoder or trying another approach to produce the common representation space.

5.1 Pre-trained LMs for low-resource machine translation

We conducted an empirical study to understand and quantify the effect of pre-trained LMs and multilingual ASR in low-resource MT and ST respectively. Regarding the experimental study, we first simulate low-resource scenarios using data from high-resource category, and apply our knowledge on the real low-resource scenario. We looked into three different axes in the simulation experiments:

1. Resources / data for pre-training.
2. Architectures / models for pre-training.
3. Amount of fine-tuning data.

The Table 16 presents more details on the kind of resources available in our setup. In our scenario, we always assume that the source languages come from low-resource category, whereas the target languages come from high-resource category.

	Resource type				
Resource category	Monolingual text	Parallel text	Bilingual dictionary	Paired speech-text	Paired parallel speech-text
High	✓	✓	✓	✓	✓
Low	✗	Low	✓	✗	Low

Table 16: Caption

Table 17 presents the data splits for low-resource simulations. Both the HOW2 and MUST-C are speech translation datasets from English to Portuguese and German respectively. These datasets were used in low-resource simulations for both MT and ST experiments.

Dataset name	Language pair	Training	Valid	Test
HOW2 [50]	en → pt	300 hr (183979 utts)	3 hr (2018 utts)	3.7 hr (2305 utts)
HOW2-153h	en → pt	153 hr (94397 utts)		
HOW2-51h	en → pt	51 hr (31408 utts)		
HOW2-17h	en → pt	17 hr (10511 utts)		
MUSTC-v1 [11]	en → de	400 hr (229703 utts)	2.5 hr (1423 utts)	4 hr (2641 utts)
MUSTC-v1-153h	en → de	153 hr (87613 utts)		
MUSTC-v1-51h	en → de	51 hr (29187 utts)		
MUSTC-v1-17h	en → de	17 hr (9778 utts)		

Table 17: Datasets and splits used for low-resource simulation.

Given, the above scenario, we studied how various pre-training objectives can affect the downstream MT / ST performance. More precisely, we experimented with two architectures for pre-training LMs, apart from using already trained models available on the internet. The following sub-section briefly explains the two pre-training LM architectures.

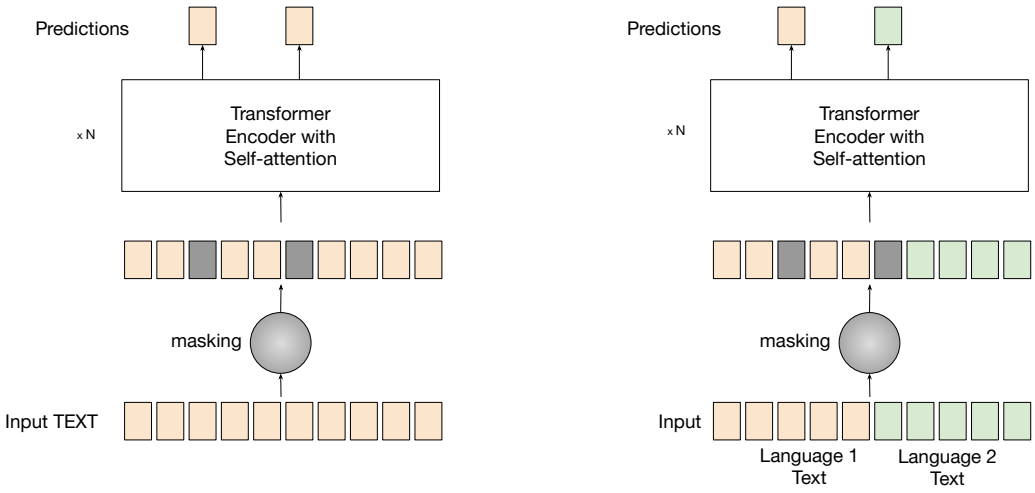


Figure 13: Transformer encoder for pre-training language models. The input is text tokens. Masking function will randomly mask 15% of the input tokens. The model is trained to maximize the likelihood of true tokens at the masked positions. (Left) Maksed language model (MLM) trained on text from one language at a time. (Right) Translation language model (TLM) trained on parallel text from pairs of languages.

5.1.1 LM architectures for pre-training

The first one is called XLM [15], which is based on transformer encoder architecture training to maximize the log-likelihood of true tokens at masked positions from the input tokens. Figure 13 illustrates two variants of XLM. The one on the left is trained with text from one language at a time, whereas the one on the right, called TLM is trained with parallel text from two languages as input. This allows the model to learn semantic relations across the input languages. However the TLM can only be trained in the presence of parallel text, which we assume is not available for languages from low-resource category.

Fig. 14 compares the BLEU scores on MT trained from scratch on various amounts of data, with the ones relying a pre-trained model. The model was pre-trained only on the in-domain data, i.e., the entire training corpus with MLM objective. Additionally, we also used an existing pre-trained model, namely XLM-17 [6] that was pre-trained on Wikipedia text from 17 languages. From Fig. 14 we can observe that as the amount of training data decreases, the performance drops. However, the degradation is much lower when relying on a pre-trained model. Finally, we can see that pre-training only on in-domain data (368k sentences) yields decent results as compared to the one pre-trained on a much larger model pre-trained on 17 languages from Wikipedia.

5.1.2 Pseudo parallel data for pre-training LMs

While many low-resource languages lack monolingual data for pre-training LMs, it is relatively easier to obtain a bilingual dictionary. Under such scenario, one can create pseudo (noisy) parallel data, i.e., by simply replacing words in the monolingual text from a high-resource language, using the bilingual dictionary. Such a pseudo (noisy) parallel data could be used for pre-training [21, 57]. We conducted low-resource simulation experiments under this category.

[6] https://github.com/facebookresearch/XLM#pretrained-cross-lingual-language-models

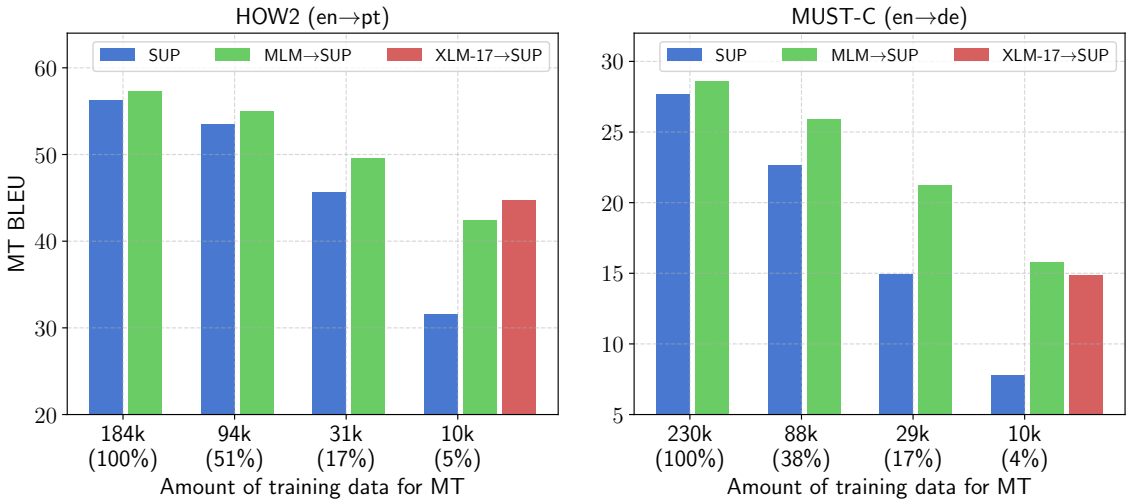


Figure 14: Supervised MT vs pre-trained LM followed by fine-tuning on various amounts of data. SUP indicates supervised training. MLM→SUP indicates in-domain MLM pre-training followed by fine-tuning. XLM-17→SUP indicates that model was pre-trained on 17 languages from Wikipedia followed by fine-tuning.

We used high-quality dictionaries for English-Portuguese and English-German [7].

Each sentence in Portuguese (high-resource) was tokenized to split words and other tokens from each other, but case was preserved. First average overall ratio was counted to know ratio upper bound. For each line in data the ratio 'how many of tokens are replaceable' was counted, if this ratio was lower than desired ratio, whole line was deleted, otherwise words was randomly replaced by words from dictionary until the ratio was equal or higher than desired ratio. The size of pt-en vocabulary is 108686, and the Table 18 presents the overlap in terms on word types (unique) and tokens (overall).

	train		valid		test	
Language	unique [%]	overall [%]	unique [%]	overall [%]	unique [%]	overall [%]
pt	42.69	65.32	70.14	65.38	68.11	65.08
en	53.69	76.67	81.63	76.67	82.15	76.86

Table 18: Portuguese-English dictionary overlap with HOW2 dataset.

Given the overlap, we can create various amounts of noising (word replacement). Table 19 presents the percentage of noising (word replacement) and resulting amount of retained sentences. Higher replacement, results in lower amount of sentences. For the experiments, we considered only the in-domain data, i.e., the entire training set of 184k sentences per language (see Table 17).

Noise [%]	Sentences retained [%]
10	99.07
30	98.06
60	69.24

Table 19: Percentage of data retained after noising (word replacement).

The pseudo parallel data with different amounts of noising was used to different pre-train the transformer based encoder using MLM. Then each of the model was fine-tuned on the 10k parallel sentences (Table 17) as prepared for the low-resource simulations. The Fig. 15 compares these translation results in BLEU relying on pre-trained models from noisy data. The figure has lower and upper bounds, which indicate two different pre-trained models. The lower bound is the scenario where only monolingual Portuguese data was seen during pre-training, and the upper bound represents the scenario where both English and Portuguese was seen during pre-training. From the left part of Figure 15 we can see that TLM pre-training on pseudo parallel data with 50% replacement, we could achieve the same BLEU score as the upper bound. However the similar trend was not seen on MUST-C dataset (right part of the Figure). It requires further investigation to ascertain if the behaviour is general across several datasets and language pairs.

[7] https://github.com/facebookresearch/MUSE

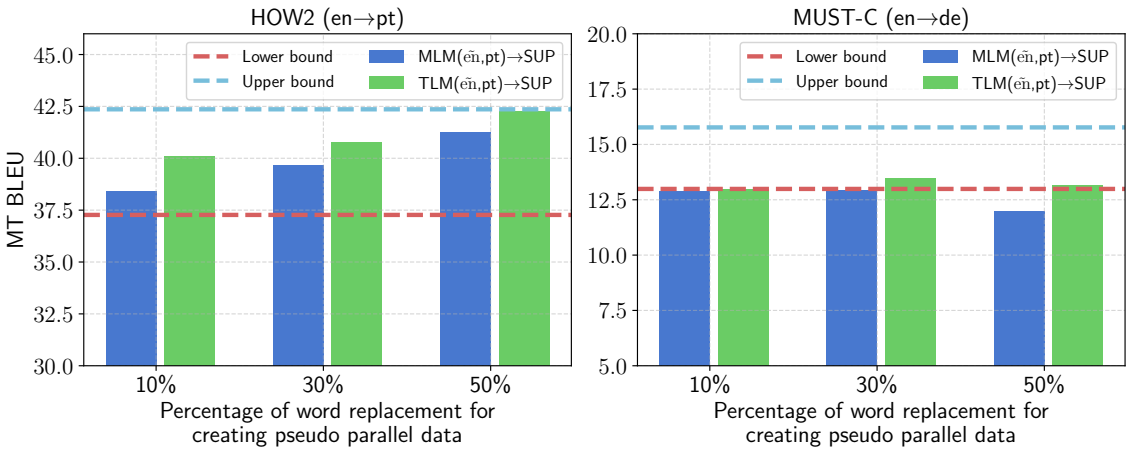


Figure 15: Comparison of BLEU scores after fine-tuning on 10k parallel sentences, while the model was pre-trained on various amounts of (noised) pseudo-parallel data. The lower-bound represents the model that was pre-trained only on target language, where as upper-bound represents the model that was pre-trained on monolingual data from both the source and target languages.

5.2 Pre-training a multilingual ASR for low-resource speech translation

In order to quantify the affect of amount of labelled data on speech translation, we emulated low-resource scenarios using HOW2 dataset. We used the same data splits as presented in Table 16 and conducted experiments. One of the ways of training a encoder-decoder speech translation system is by training independent encoder-decoder based ASR and MT systems, and then combining speech encoder from ASR with text decoder from MT and fine-tune on the labelled speech translation data. This scheme is illustrated in Fig. 16

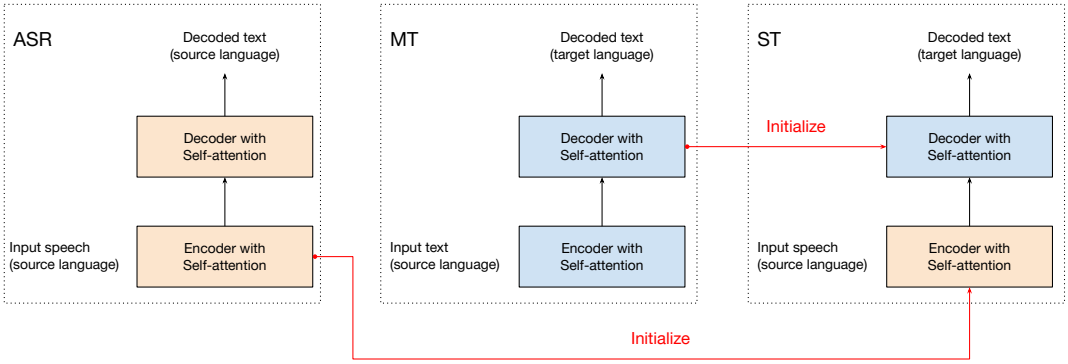


Figure 16: Block diagram showing the pipeline fro training a typical speech translation system using an encoder-decoder framework.

Fig. 17 shows the evaluation of speech translation in terms of BLEU scores, when using various amounts of training data. In all the cases we train an ASR, and MT followed by fine-tuning for ST. We can see that the BLEU score drops from 44 to 9 as the amount of training data decreases from 300 hr to 17 hr. The current speech translation pipeline is difficult to fine-tune given low-amounts of data (17hr).

A sequence-to-sequence encoder-decoder framework for automatic speech recognition models both acoustic and language information into a single end-to-end trainable model. The decoder

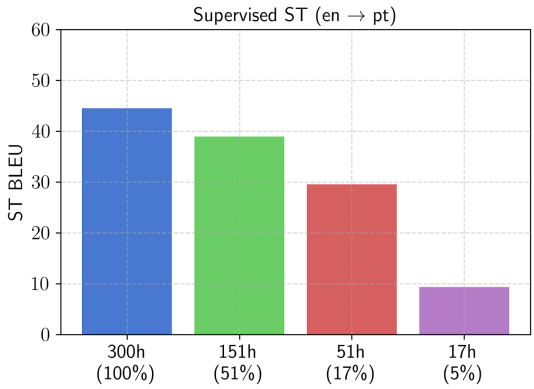


Figure 17: Low-resource simulation of speech translation on English (en) → Portuguese (pt) from HOW2 dataset. For all the splits, we train both ASR and MT models independently. Then initialize an ST model with encoder from ASR and decoder from MT, followed by fine-tuning.

also acts as an internal language model. This motivated us to explore a multilingual ASR for speech translation in low-resource settings. We hypothesize that the internal language model of the multilingual ASR can be beneficial, when translating from speech in low-resource language to text in high-resource language. For initial experiments, we selected 6 languages (es, pl, fr, de, it, nl) and pooled 50 hours of speech, together with corresponding transcriptions for each language from Mozilla common voice v8 corpus. All the languages are written in Latin script, which allowed us to build a shared sub-word vocabulary using uni-gram algorithm from `sentencepiece` toolkit. The multilingual ASR was trained on 300 hr (50 hr $\times 6$ languages $=$ 300 hr) in total and then fine-tuned on 17 hr split from HOW2 corpus (en $\rightarrow$ pt). Note that the initial ASR model has not seen any Portuguese text, but it could still generate text sequences as all the prior languages are written in Latin script. To compare with different initialization schemes, we also trained ASR on source language (en) which is then used to initialize for speech translation system. Also, we trained a speech translation system with random initialization, which shows the benefit of pre-trained ASR for initialization. The results of the speech translation experiments are given in Table 20. We can see that random initialization (1) yields worse results, which is followed by system (2) that is initialized from ASR and MT trained on 17 hr and 10k (low-amounts) respectively. The next systems (3), (4) and (5) represent the scenario, where we have varying amounts of training data for ASR in source language (en). Systems (6) and (7) assume that training ASR in source language is infeasible and instead relies on multilingual ASR. From these experiments, we could see that initializing ST with source language ASR or an multilingual ASR is beneficial in low-resource settings.

We used the same multilingual ASR and trained speech-translation system from Tamasheq (taq) speech to French (fr) text using training data from IWSLT 2022. We also compare with other baselines provided by the organizers. The results are shown in right sub-plot of Fig. 18. We see that our system is slightly better than other systems based on self-supervised training, nevertheless all the systems are have relatively low BLEU scores. We plan to investigate further into this direction.

5.3 Can we disentangle modalities from the common space?

Assuming that our encoders can produce multi-lingual speech and text representations that all in the same common joint embedding space, it is now necessary to develop a decoder that

System	Speech Encoder Init.	Text Decoder Init.	ASR Aux. Loss	BLEU
(1)	Random	Random	YES	1.9
(2)	ASR-17h-src	MT-10k	YES	9.5
(3)	ASR-51h-src	ASR-Decoder	YES	16.5
(4)	ASR-153h-src	ASR-Decoder	YES	17.9
(5)	ASR-300h-src	ASR-Decoder	YES	19.6
(6)	Multi-ASR-300h	Multi-ASR-Decoder	YES	15.5
(7)	Multi-ASR-300h	Multi-ASR-Decoder	NO	14.6

Table 20: ASR-XXh-src is trained on source (en) speech, where XX represents the amount of training data in hours (h). MT-10k is source (en) → target (pt). Multi-ASR-300h did not see neither source speech nor target language text.

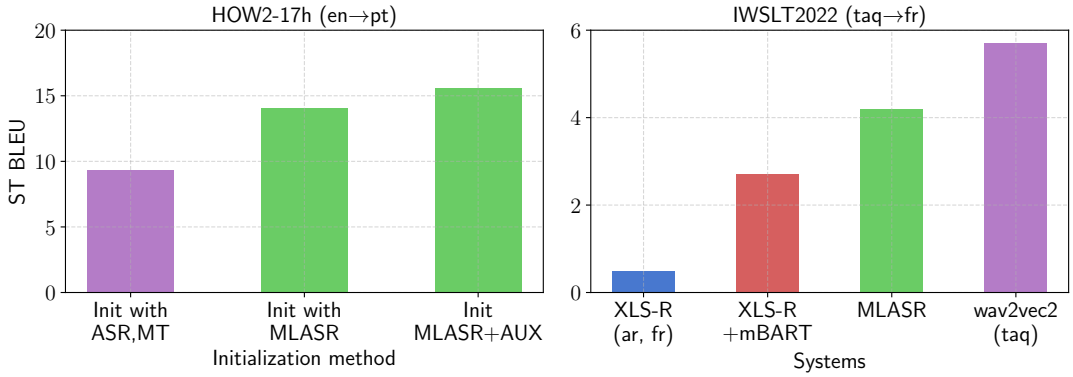


Figure 18: BLEU scores for speech translation. (Left) Low-resource simulation in English → Portuguese from HOW2 dataset. (Right) Tamasheq → French from IWSLT 2022 evaluation dataset.

can disentangle the information to project it in a space that is more suitable to generate text or speech in the target language. Figure 19 depicts a framework in which each couple of target modality and language is generated from this common space by using a specific decoder. This approach is highly sub-optimal as it requires to train two decoder (speech and text) for each target language and doesn't allow any mutualisation during the training step.

We propose in this part to develop a unique decoder that could take as input a sequence of embeddings from the multi-modal, multi-lingual joint space and that produces either a sequence of embeddings that is suitable to generate speech or text in the target language. This decoder would address the difficulty of the length of the sequence of embedding that is usually much shorter for text than for speech. Ideally, such a decoder would then allow the use of a unique multi-lingual vocoder that would generate speech from the pseudo-speech embeddings and a unique ASR system that would do the equivalent job for text generation. The architecture of such a framework is depicted in Figure 20

In this framework, a first decoder is responsible of disentangling the modality and translate the sequence at the same time.

5.3.1 Proposed architecture

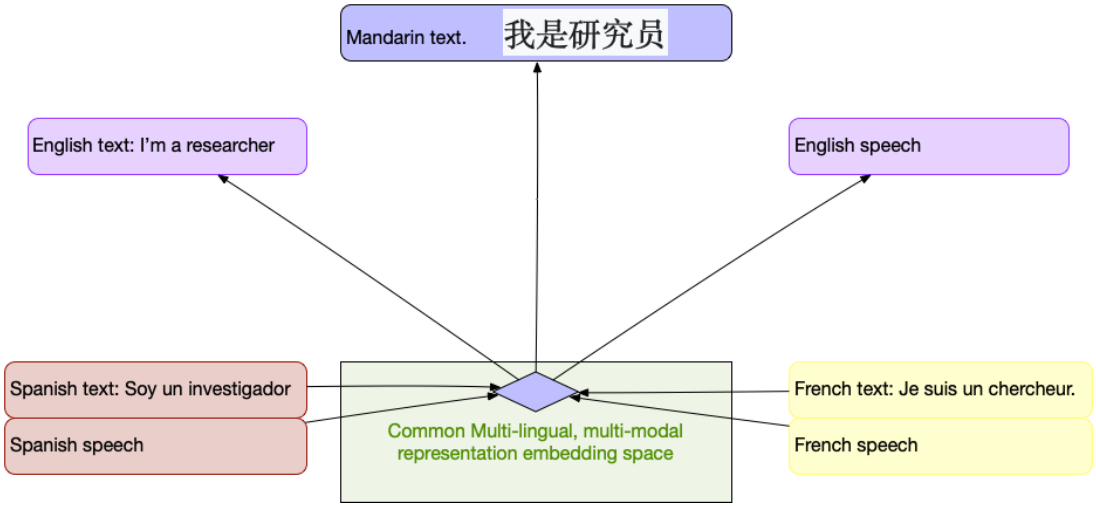


Figure 19: Framework of a multi-decoder architecture.

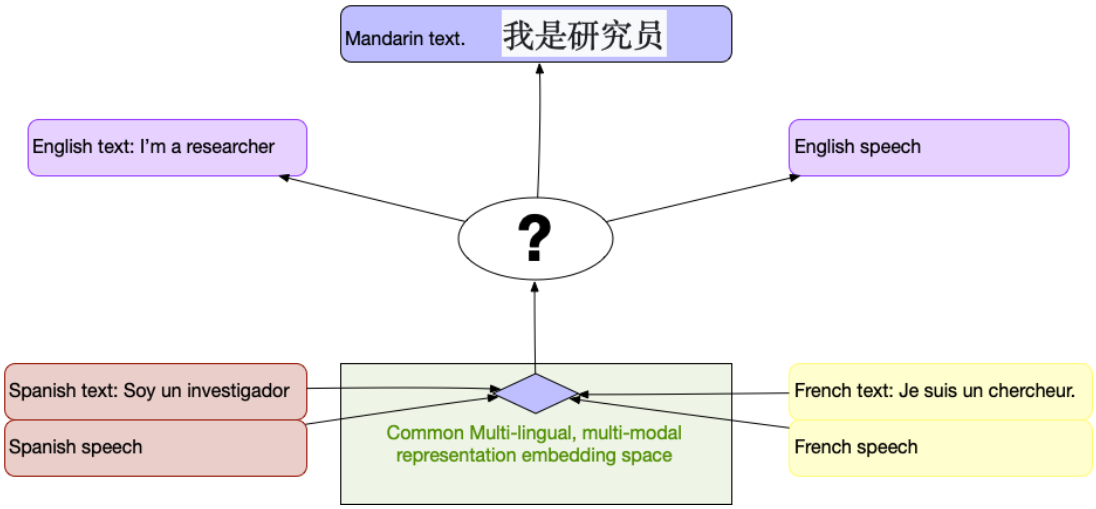


Figure 20: Decoding framework based on a single decoder disentangling both language and modality.

5.3.2 Training data and process

5.3.3 Duration estimation

TODO: "loan, veux tu ajouter quelque chose ici?"

5.3.4 Preliminary results on text generation

TODO: "resp text generation: loan"

5.4 Can we build a common space by aligning sequences instead of single embeddings?

TODO: "resp: Peter"

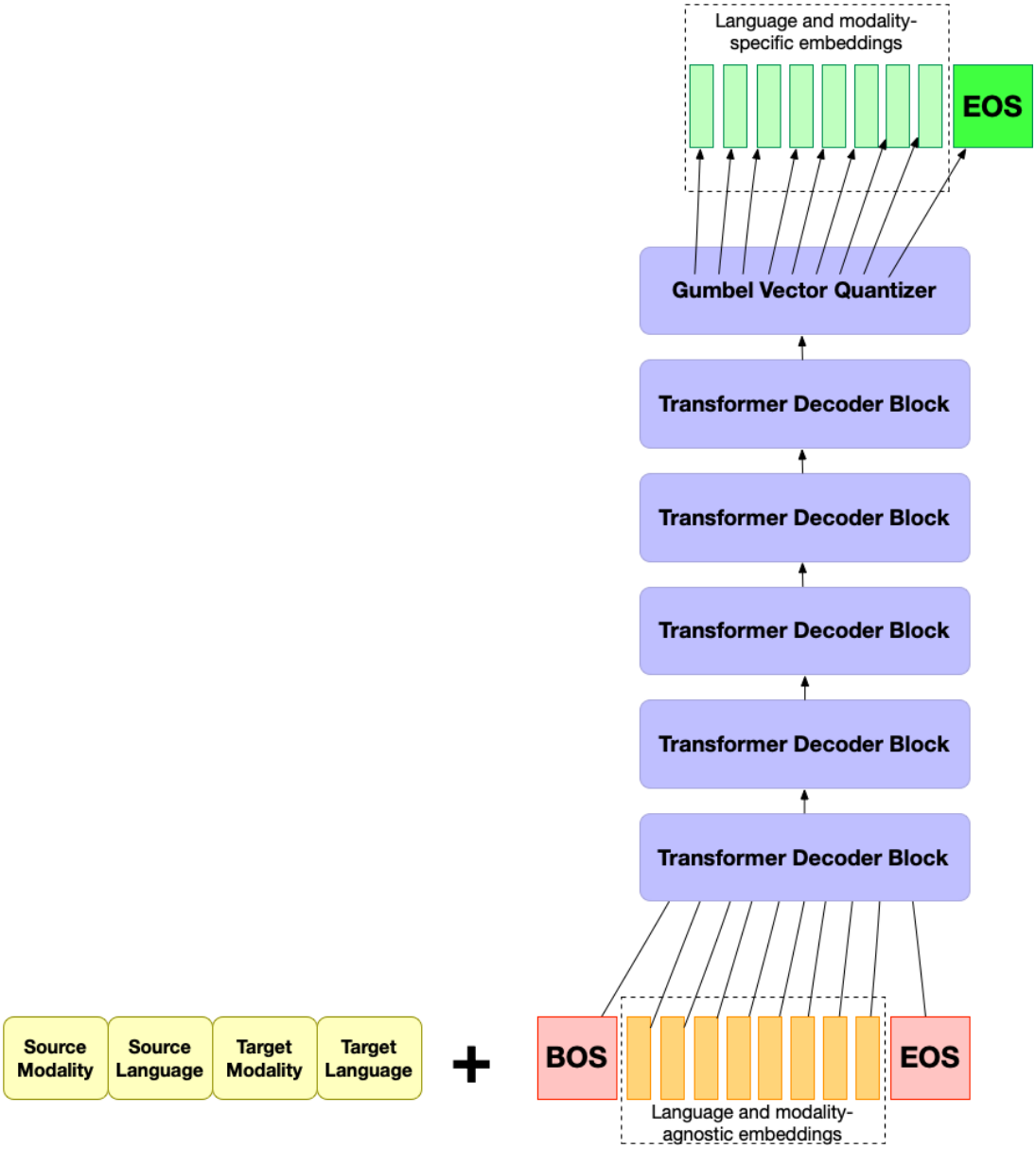


Figure 21: Architecture of the proposed decoder.

As discussed in the Global Scope of the Workshop topic, the alignment of modalities with differing temporalities is complex. Our primary system considers alignment through self-attentive pooling cosine minimisation. We also considered sequence alignment following both empirical results in a mono-lingual setting [3] and to consider the relative advantages of a joint multimodal spaces [25] over the coordinated ones produced bu SAMU-XLS-R [32].

The proposed architecture is based on SpeechT5 [3], a shared space speech and text encoder-decoder model that is trained to align sequences of embeddings from English utterances. The main contributions of this system are :

- To take speech and text inputs by using two different *pre-nets*, one for each modality, producing sequential representations with equal hidden feature dimensions.
- To force a convergence of speech and text embeddings within a joint space by using a restricted size common codebook and randomly mapping a fixed proportion of utterance

frames into the codebook.

- To reconstruct speech and text by using two different *post-nets*.

Unlike prior translation work, we enforce a joint space across speech and text *sequences*, with 20ms speech frame features and BPE word token hidden states both passed to the same encoder-decoder transformer architecture. In order to generate informative cross-lingual representations for our system we use the cross-lingual XLM-R text encoder [16] (100 languages) and the XLS-R speech encoder [6] (128 languages) as 'pre-nets'. Furthermore, we use the codebook objective to close representations across modalities **and** languages, with non-aligned multilingual speech and text data.

We proposed a set of experiments to progress from an uni-lingual multi-modal system [3] to a cross-lingual multi-modal system, by increasing step by step the amount of languages used for training. We speculate that our model can learn powerful cross-lingual representations, even if the time allowed during the workshop was not enough to get comprehensive results.

5.4.1 Model Architecture

The architecture we use is informed by both SAMU-XLS-R, and by SpeechT5.

Text Pre-Encoder We replace the base SpeechT5 text pre-net with a multilingual different encoder.

We tokenize raw input text with Sentence Piece [34] in a unigram setting. We follow the XLM-R [16] authors in setting a vocabulary size of 250K. As an initial text pre-encoder, we take the pretrained XLM-R Base model with 270M parameters. XLM-R is trained on 100 languages with a multinomial sampling distribution parameterised by α [15]. This model generates sequences of hidden representations of hidden dimension 768 at the character level. During training, we freeze XLM-R for the first 10 epochs, before allowing a scaled loss $lr_s = 0.1 \cdot lr$ update to the final 4 layers for the rest of ESPERANTO pretraining.

Speech Pre-Encoder We changed the base SpeechT5 speech pre-net using a different encoder, described here.

We accept Mel-filterbanks as an input for the speech prenet. We then use XLS-R [6], an architecture inheriting from wav2vec 2.0 [7]. We initialise the speech encoder with this XLS-R model, pretrained using speech data from 128 languages with a total of duration of 436K hours.

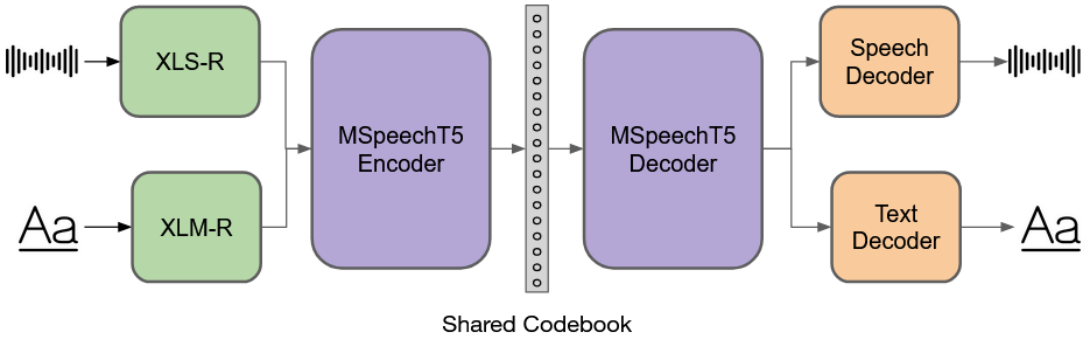


Figure 22: MSpeechT5 Architecture

MSpeechT5 We pass the hidden speech and text representations to an encoder-decoder model with 12 encoders layers and 6 decoder layers. Both encoder and decoders have 12 attention heads. To encourage learning crossmodal and crosslingual representations the model is pretrained with a larger set of objectives than the base SpeechT5:

1. Speech: Speech unit cluster classification (HuBERT)
2. Speech: Log Mel-filterbank reconstruction
3. Speech: Sequence length prediction
4. Text: Masked language prediction
5. Speech + Text: Codebook mapping with diversity loss
6. Speech + Text: Sequence Reconstruction Objective
7. Speech + Text: Sequence Mixing Objective

Unimodal Unilinugal objectives Unimodal objectives are inherited from the SpeechT5 architecture.

Speech unit cluster classification (1) specifies a loss between the models confidence of the HuBERT Speech Cluster unit taken from the HuBERT model trained on shared speech data. We use units from after 1 epoch of training and fit k-means with clusters=500.

Log Mel-filterbank reconstruction(2) is a sequence reconstruction objective seeking to minimise the sum of the L1 distances between source and target log Mel-filterbank features with a dimension of 80.

Sequence length prediction (3) is a binary cross entropy loss on the stop token of the log Mel-filterbank sequence reconstruction

Masked language prediction (4) is an auto-regressive language reconstruction loss over randomly masked input text tokens.

Multimodal multiligual objectives The **shared codebook objective** (5) is inherited from SpeechT5. Frames from either speech or text are mapped to a fixed size codebook with a prodiversity loss. SpeechT5 [3] proposes 1K codebook entries. We scale the size of the codebook C by number of languages $L \in \mathbb{N}^*$ with the following :

$$C = 1000 + 250 \log_2(L) \tag{1}$$

We intuit that this objective encourages the alignment of semantic representations between cross-lingual and cross-modal frames with similar content.

Shared sequence reconstruction objective (6) requires aligned sentences and is designed to encourage the model to share information across languages and modalities. The objective is heavily masked sequence reconstruction (Text or Speech) with a guiding semantically aligned statement X^a from a different language or modality, where $\hat{X}$ is a masked sequence:

$$L_{SR} = \sum_{n=1}^{t} logp(\mathbf{y}_n^t | \mathbf{y}_{<n}^y, \hat{\mathbf{X}}^t, \mathbf{X}^a)$$

Sequence mixing objective (7) is based on the intuition that across languages and modalities, utterances with the same semantic content should have the same quantized states in the

codebook. It is enforced by randomly sampling $\alpha = 10\%$ of the codes C_m^a in an aligned target X^a. The objective is to minimize L2 distance between the$\alpha = 10\%$ of the closest source frames H^s to target code centroids C_m^a, using each source frame at most once. This objective may be approximated by:

$$L = \sum_{\alpha n} \min_{n,m}(||H_n^s - C_m^a||)$$

with the provision that each frame H_n^s is only used once.

5.4.2 Experimental Setup

Data The model is trained on a cleaned custom assembled Common-Crawl [17] with 100 languages. We also the CommonVoice Dataset [4] to leverage unaligned data. For the Aligned Loss objectives - We use aligned data from the CovostV2 dataset [56]. We use task-specific corpora for finetuning and evaluating on the downstream tasks which are described in Subsection 3.

Models trained We train and evaluate our model progressively across a 3 Setups that are presented in the table 21.

Table 21: Versions of SpeechT5 trained.

Version	Languages	Comment
Mono-Lingual	EN	Baseline SpeechT5 [3]
Bi-Lingual	EN, FR	To compare with EN to FR translation perfs.
Multi-Lingual	EN, FR, DE, ES, IT, AR, JP, ZH-CH, TA, TU	Toward multi-linguality.

Expected Results During the workshop we implemented the model as described in the preceding section. We are currently working on moving from SpeechT5 baseline results to a multilingual setting with the cross-lingual encoders. Unfortunately, we did not produce usable multilingual results within the timeframe of the workshop.

5.5 From SAMU-XLSR to Translation

5.5.1 Semantically-Aligned Multimodal Cross-lingual Speech Representations

In [32] we proposed the SAMU-XLSR: Semantically-Aligned Multimodal Utterance-level Cross-Lingual Speech Representation learning framework. Unlike previous works on speech representation learning, which learns multilingual contextual speech embedding at the resolution of an acoustic frame (10-20ms), this work focuses on learning multimodal (speech-text) multilingual speech embedding at the resolution of a sentence (5-10s) such that the embedding vector space is semantically aligned across different languages. We combine state-of-the-art multilingual acoustic frame-level speech representation learning model XLS-R with the Language Agnostic BERT Sentence Embedding (LaBSE) model to create an utterance-level multimodal multilingual speech encoder SAMU-XLSR. Although we train SAMU-XLSR with only multilingual

transcribed speech data, cross-lingual speech-text and speech-speech associations emerge in its learned representation space. To substantiate our claims, we use SAMU-XLSR speech encoder in combination with a pre-trained LaBSE text sentence encoder for cross-lingual speech-to-text translation retrieval, and SAMU-XLSR alone for cross-lingual speech-to-speech translation retrieval. We highlight these applications by performing several cross-lingual text and speech translation retrieval tasks across several datasets. A more detailed description of the model can be found in the original paper [32].

5.5.2 Speech to Text Translation

Task Overview **X→EN Text Translation**: We use the CoVoST-2 [56] X-EN speech-translation dataset for this task. The task consists on the translation from input language X∈{RU, IT, FR, ES, TR, DE, ET, CY, NL, ID, CA, FA, AR, ZH, SV, MN, SL, JA, TA, LV} into English. We propose to use SAMU-XLSR and XLS-R to perform speech to text translation.

Models The model uses the classical encoder decoder framework. The encoder is either initialized with SAMU-XLSR or XLS-R and fully-fine tuned or fine-tuned using adapters layers.

The decoder is either randomly initialized or initialized with mBart [38] and partially fine-tuned using different strategies (encoder attention and layer norm fine-tuning [37]).

Data The previous SAMU-XLSR was trained on 21 languages. We trained a new version of the model using 51 languages, as the intersection between the languages supported by LaBSE and the languages included in the common voice 8.0 corpus.

The languages are reported in table 22.

We evaluated our models on the 21 CoVoST2 languages. See the details in table 23.

In the rest of the section, we consider as high resource languages the one with more than 100h, and the low resource languages the one with less than 10h of data for training.

Evaluation Metric We evaluated our models using the well known BLEU score.

Results In this section we report the results that we got using different initialization of the encoder and decoder, we also investigate different fine tuning strategies and zero-shot experiments.

Figure 23 shows results using randomly initialized decoder. We can see that SAMU-XLSR is performing better than XLSR for medium and low resource languages in every configurations. SAMU-XLSR with adapter is slightly less good than XLSR for High resource languages. We believe this is because of the important amount of training data that can not be handled by the adapter layers.

Figure 24 shows the results using mBart decoder initialization. Only the layer norm and the encoder attention layers were fine-tuned. In those experiments we also fine-tuned only adapters when using the SAMU-XLSR model as initialization of the encoder, when using XLSR the encoder is fully fine tuned. We can observe that using mbart to initialize the decoder lead to improvements using XLSR and SAMU-XLSR as encoder. SAMU-XLSR is outperforming XLSR by a big margin in every configurations.

Language	Hours	Language	Hours	Language	Hours
English	2886	Basque	144	Romanian	36
Kinyarwanda	2383	Arabic	139	Tatar	29
Esperanto	1856	Portuguese	130	Greek	25
German	1133	Western Frisian	125	Hungarian	24
Catalan	1036	Chinese HK	120	Lithuanian	20
Belarusian	987	Dutch	105	Mongolian	18
French	902	Chinese CN	95	Slovakian	18
Spanish	739	Ukrainian	76	Hindi	16
Swahili	655	Turkish	68	Maltese	16
Persian	365	Czech	67	Galician	15
Tamil	341	Uighur	63	Finnish	14
Thai	340	Indonesian	53	Bulgarian	11
Italian	335	Northern Kurdish	53	Slovenian	10
Uzbek	227	Swedish	48	Irish	8
Russian	193	Estonian	44	Georgian	8
Polish	162	Kyrgyz	44	Latvian	8
Welsh	145	Japanese	43	Danish	6

Table 22: Training of SAMU-XLSR: Languages and hours used from Common Voice 8

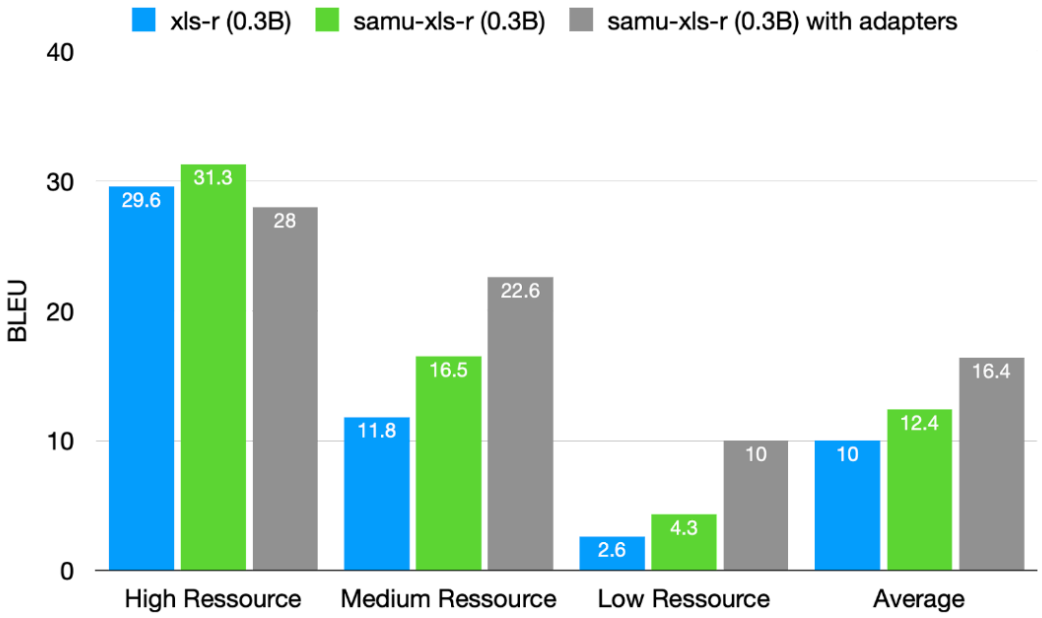


Figure 23: BLEU results on CoVost2 using randomly initialized decoder

As a comparison, we also report the results of mBart initialized and randomly initialized decoder using SAMU-XLSR as encoder initialization and using adaptors for the fine-tuning in figure 25.

Source language (X)	# utts	CoVoST 2 (hours, X)	CVSS-C (hours, English)	CVSS-T (hours, English)
French	207364	264.3	174.0	192.7
German	127822	184.3	112.4	124.2
Catalan	95852	135.6	88.1	95.0
Spanish	79012	113.1	69.5	73.7
Persian	53901	49.2	25.3	29.3
Italian	31698	44.2	29.4	30.5
Russian	12112	18.2	13.3	13.2
Chinese	7085	10.4	8.7	9.3
Portuguese	9158	10.3	5.7	6.5
Dutch	7108	7.3	4.9	5.1
Turkish	3966	4.1	3.0	3.1
Estonian	1782	3.4	2.8	2.7
Mongolian	2067	3.0	1.9	2.1
Latvian	2337	2.1	1.2	1.4
Arabic	2283	2.1	1.1	1.2
Slovenian	1843	2.0	1.1	1.3
Swedish	2160	1.7	1.0	1.2
Welsh	1241	1.7	0.9	1.0
Tamil	1358	1.6	0.9	1.1
Japanese	1119	1.3	0.8	0.8
Indonesian	1243	1.2	0.7	0.7

Table 23: Training data in CoVoST 2 and CVSS databases

We also observed that our SAMU-XLSR / mBart model was not performing equally depending on languages, as reported in figure 26.

We performed some experiment in which we built bilingual models (instead of having one 21 to X languages model, we trained 21 models language specific to english). The results, presented in figure 27, are far better for some low resources languages (Latvian and Indonesian for example). For some languages (Arabic and Dutch), performances are degraded. We believe that those languages are close to others in the training set and that they were able to benefit from them in the multilingual training.

We were also interested about zero-shot translation. We performed the training on high resource languages (> 100h) and use the other languages during inference. Once again, results presented in figure 28 shows that SAMU-XLSR performs better than XLSR due to the fact that he is learning semantically aligned representations of speech.

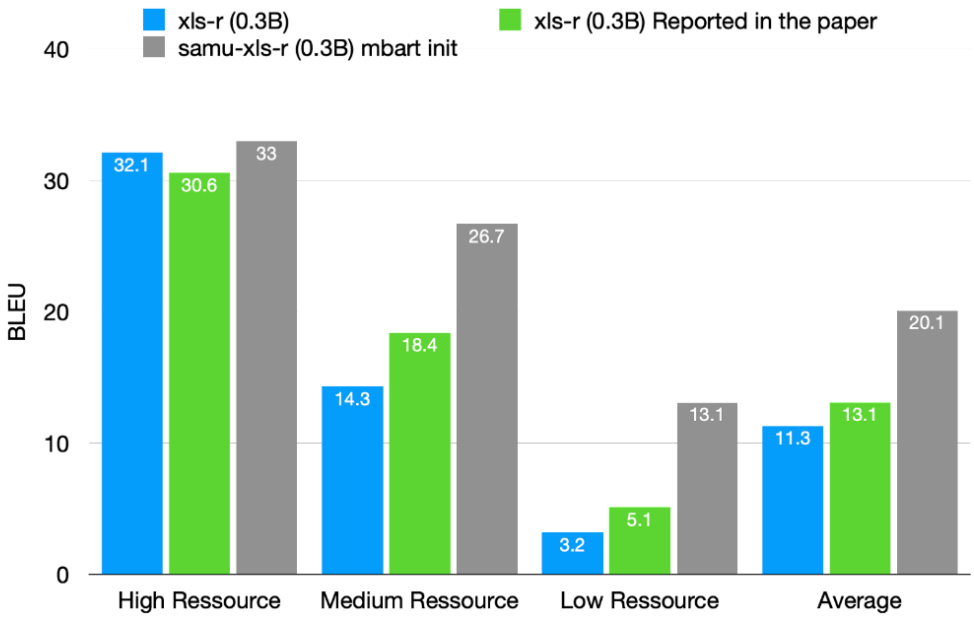


Figure 24: BLEU results on CoVost2 using mBart initiliazed decoder

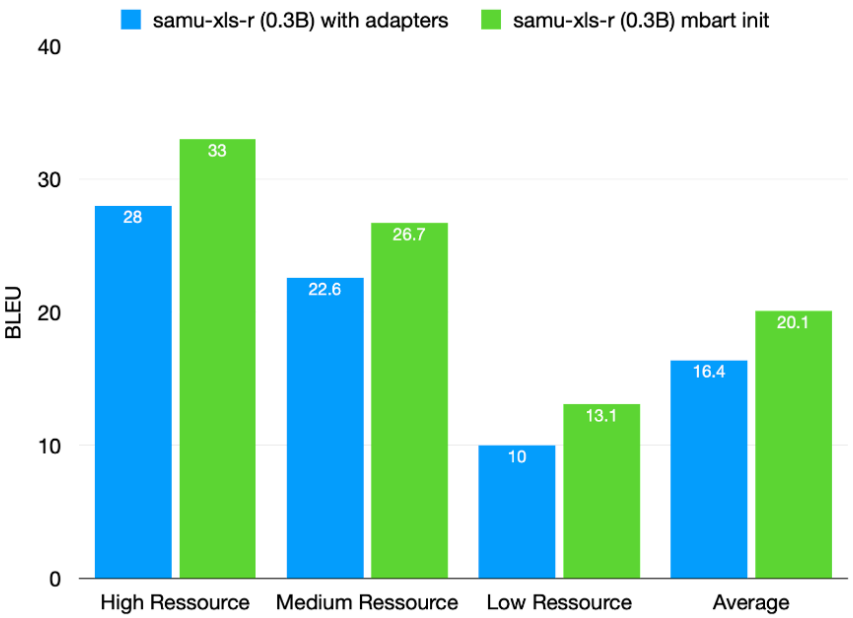
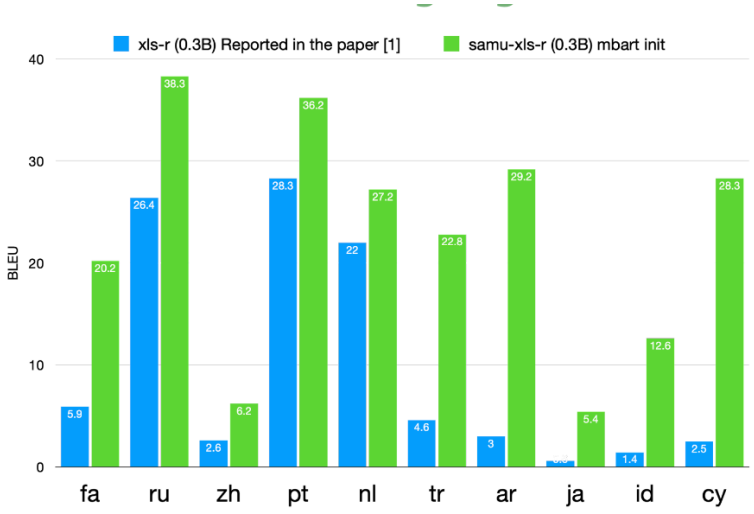


Figure 25: BLEU results on CoVost2 comparison random / mBart decoder

5.5.3 IWSLT 2022: Low-resource Speech to Text Translation

Task Overview One of the main goals of this project is to deal with low resource scenarios in the speech translation framework. In order to introduce under resourced setups in our study, we performed different experiments under the conditions of the IWSLT 2022 challenge low resource speech translation task. From this task, we will be focusing on resource-scarce settings for translating input speech in Tamasheq into French text. Tamasheq language is a variety of Tuareg, widely spoken in North Africa, namely Algeria, Mali, Niger and Burkina Faso. Data from 2014 estimate that there are around 500.000 Tamasheq speakers around the world, most



[1] Babu, A., Wang, C., Tjandra, A et al. (2021). XLS-R: Self-supervised cross-lingual speech representation learning at scale. *arXiv preprint arXiv:2111.09296*.

Figure 26: BLEU results on CoVost2 for different languages

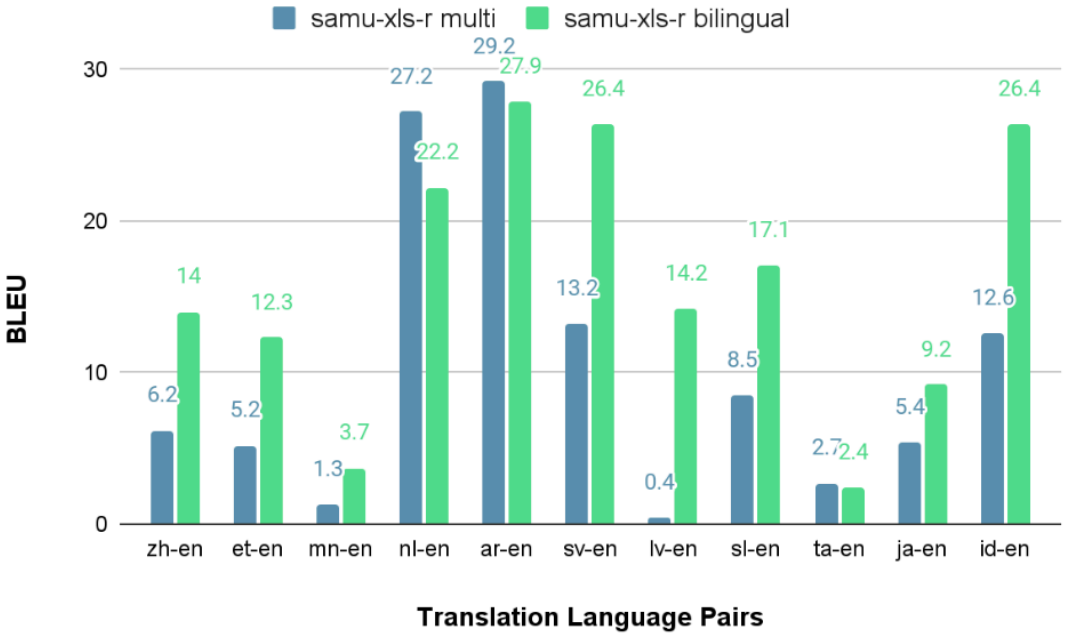


Figure 27: BLEU results on CoVost2 using bilingual models

of them being Malian.

Models During the workshop, a new version of SAMU-XLSR was developed with even more languages. New SAMU-XLSR model trained on 60 languages including data from BABEL.

Data In order to support the challenge, an annotated corpus with parallel Tamasheq speech and French text was released. Data consists of 17 hours of manually labelled speech from radio recordings translated to French. Additionally, a bigger collection of unlabeled raw audio data was also made available, providing the participants with speech data in Tamasheq language

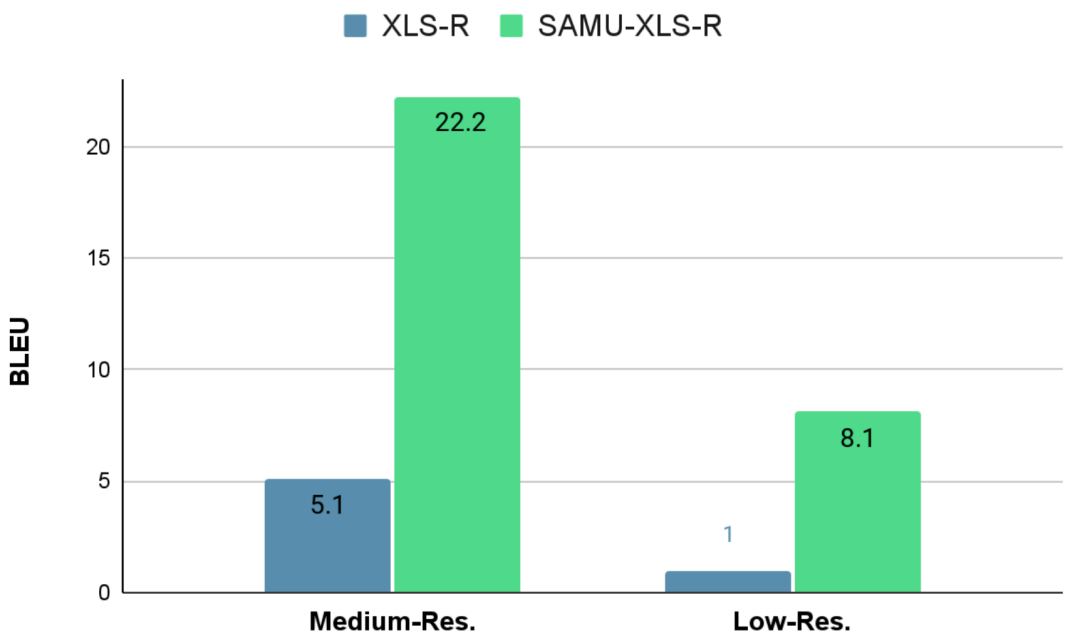


Figure 28: BLEU results on zero-shot translation

Language	Hours	Language	Hours	Language	Hours
Georgian	46	Mongolian	42	Turkish	70
Kazakh	36	Swahili	30	Vietnamese	79
Lao	59	Tamil	63	Zulu	56
Lithuanian	38	Tagalog	76		

Table 24: Training of SAMU-XLSR: Languages and hours used from BABEL

(234 hours), and other 34 languages spoken in Niger: French (116 hours), Fulfulde (114 hours), Hausa (105 hours) and Zarma (100 hours). The described dataset can be obtained freely from the IWSLT 2022 Github repository. [8]

Evaluation Metrics As usually done in translation tasks, the original evaluation considered the BLEU metric as its scoring system. We used the same metric in our experiments.

Results Figure 29 shows BLEU scores on the Tamasheq data test partitions for our proposed system using SAMU-XLSR as speech encoder (both 51 languages and 60 languages versions) compared to previous results reported in the original IWSLT'22 challenge leaderboard. First three results (in orange tones) have been extracted directly from the original post evaluation analysis paper [2]. These results feature different approaches used in the evaluation to deal with the task. They mainly rely on the use of large pretrained models, finetuning them on a set of in domain or domain related data. Best result achieved a BLEU score of 5.7 by finetuning a wav2vec 2.0 model using in domain data from the released unlabelled Tamasheq data. Last two results (in green tones) show the performance of our proposed SAMU-XLSR speech encoders in the Tamasheq to French speech to text translation. It can be seen that both the previous

[8] https://github.com/mzboito/iwslt2022_tamasheq_data

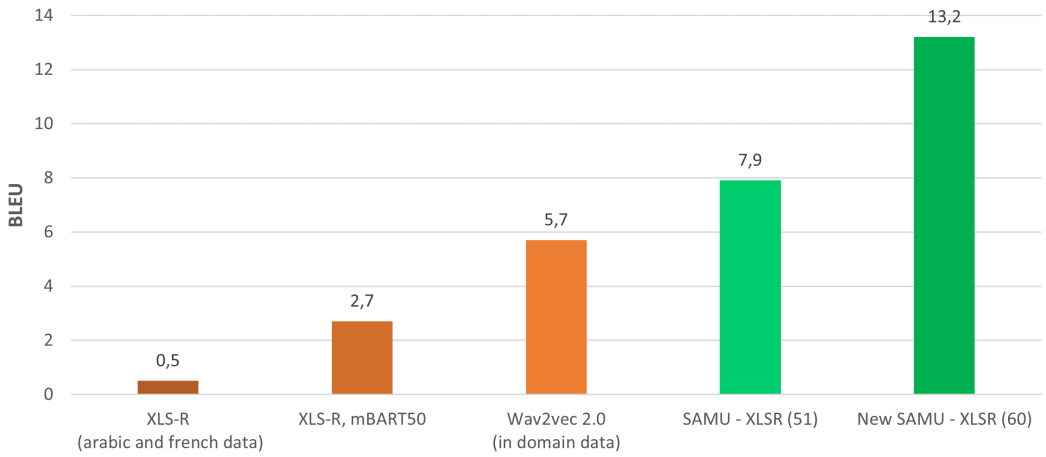


Figure 29: BLEU results on Tamasheq data test partition.

version of SAMU-XLSR with 51 languages, and the new one with 60 languages, outperform the best result so far in this task. The previous version of the SAMU-XLSR model, achieves a 7.9 BLUE score without seeing any Tamasheq data in its pretraining phase, highlighting again the importance of aligning the semantical information between languages. The new version of SAMU-XLSR trained using 60 languages benefits significantly from including Tamasheq in its pretraining step, marking a new state of the art on the Tamasheq to French speech to text translation task with a BLEU score of 13.2, showing a 67.2% relative improvement compared to the 51 languages version of SAMU-XLSR.

5.5.4 Direct (Textless) Speech to Speech Translation

Task Overview The speech to speech translation task seeks to convert speech generated in one language into speech in a different language. Traditional speech to speech systems rely on a cascaded approach that concatenates different systems, namely automatic speech recognition, machine translation and speech synthesis, or a speech to text system concatenated with a speech synthesis model. The idea of direct speech to speech translation has already been explored in the literature [30] [29], showing greats benefits in terms of lower computational costs and inference latency when compared to the cascaded approach. Yet, a performance gap between the direct and the cascaded approaches can still be observed due to the challenges of simultaneously learning the alignment between two languages and the acoustic and linguistic characteristics.

In this work, we adopt the framework originally proposed in [35] that performs textless speech to speech translation by extracting a set of discrete acoustic units on the target speech and then training a speech to unit translation model that predicts those discrete representations.

Models An schematic overview of the direct speech to speech translation system is presented in Figure 30.

Data For the direct speech to speech translation task, the English audios from the CVSS-C (canonical voice) dataset were used as target speech. Several Acoustic Unit Discovery Systems were tested for extracting the discrete units from the English audio: XLS-R, SAMU-XLSR, m-HuBERT and SpeechT5. The corresponding source speech audios were obtained

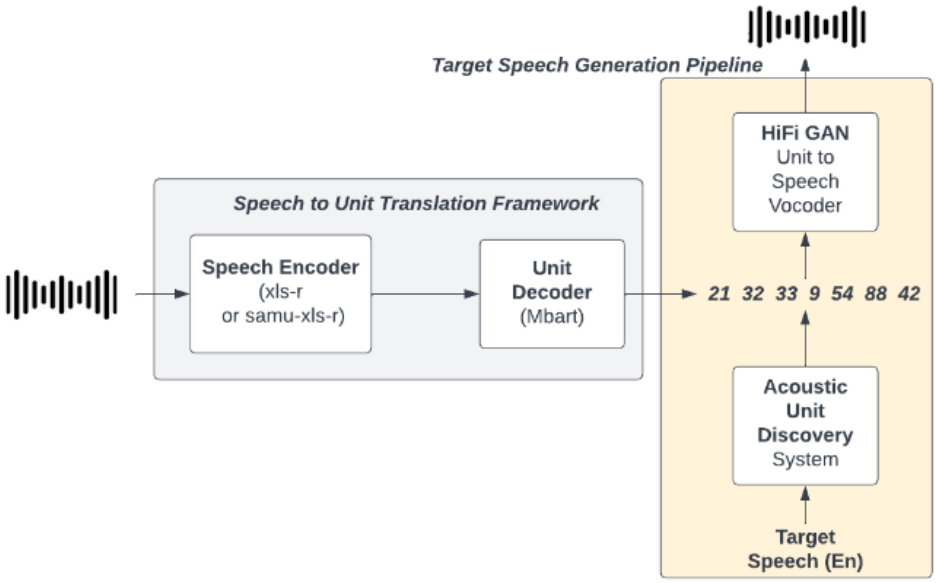


Figure 30: Textless speech to speech translation system.

from the Common Voice 4.0 corpus for the 21 languages whose translations are available in CVSS. See [28] and table 23 for more details on the CVSS dataset.

Evaluation Metric In order to evaluate the speech to speech translation, as it is not feasible to directly compare two audio signals, we use open-source ASR models [9] to obtain a transcription for the output audios. Then, the score used to evaluate our system is the BLEU metric between the obtained transcriptions and the normalised reference text.

Results To evaluate our speech to speech translation system, the BLEU scores in the test partition of the CVSS dataset are obtained and introduced in Figure 31. In this figure, a comparison between the four different systems which have been used to obtain the discrete units as a reference to train the speech to unit translation framework is presented. Moreover, two different speech encoders have been applied in the speech to unit framework to train the different systems with the reference acoustic units from each of the Acoustic Unit Discovery System. According to these results, it is worth mentioning that regardless of the Acoustic Unit Discovery System employed to extract the units, the best speech encoder is the SAMU-XLSR.

On the other hand, in Figure 32, the effect of using a different number of acoustic units has been analysed. As this figure shows, the results improved when a high number of acoustic units are extracted with m-HuBERT as Acoustic Unit Discovery System. While in the case of using SpeechT5, the results achieved are also better than the ones presented in Figure 31 with this kind of system when more acoustic units are extracted.

5.5.5 Text to speech translation

Task Overview Considering that one of the initials goals of the project was to develop a fully multimodal and multilingual translation system, and once the performance of the SAMU-XLSR

[9] https://huggingface.co/facebook/wav2vec2-large-960h-lv60-self

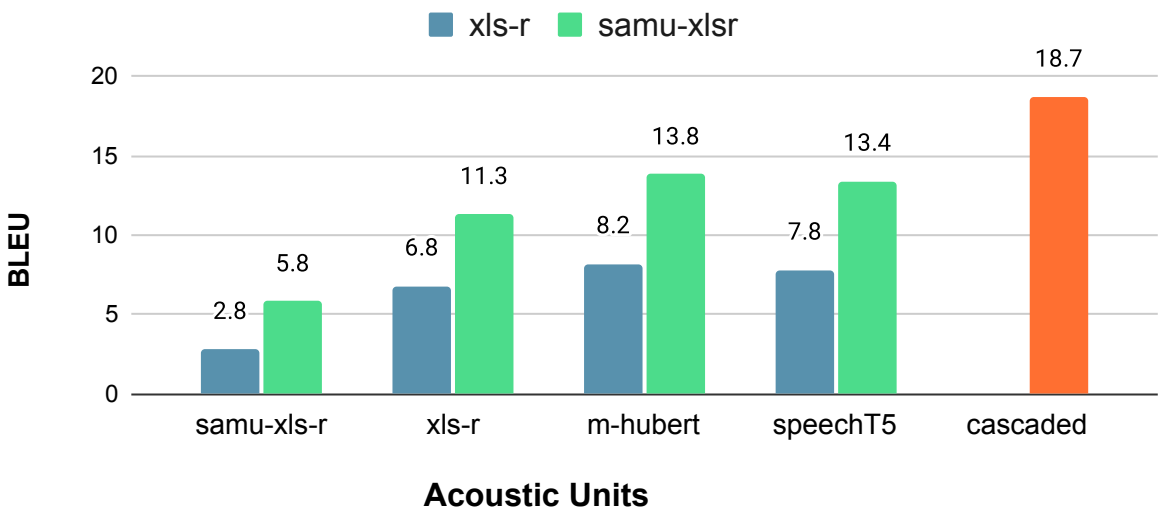


Figure 31: BLEU results on CVSS 21 X → EN translation task.

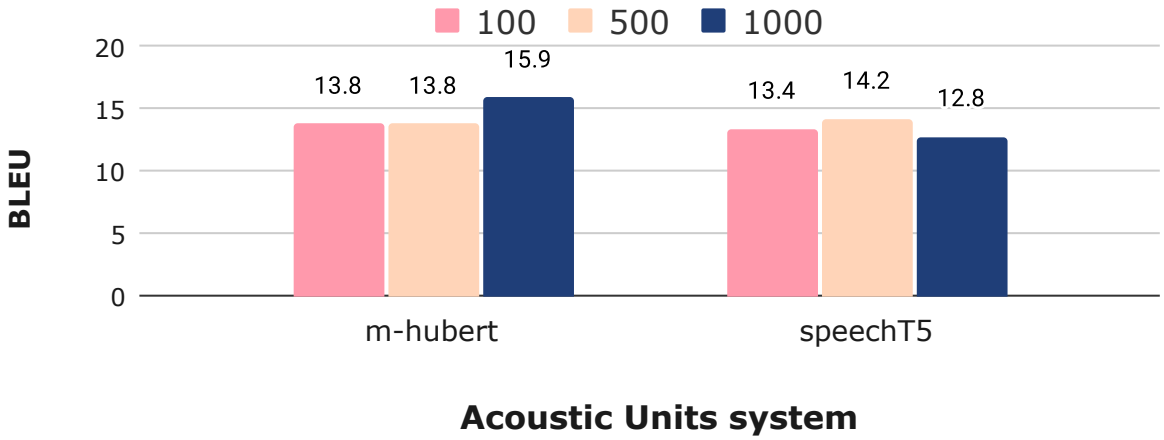


Figure 32: BLEU results on CVSS 21 X → EN translation task depending on the number of units.

model has been experimentally validated on the speech to speech translation task, we proposed to extend the speech to speech translation framework so that it can also accept text as input. The task can be then formally defined as a text to speech translation task. Seeking to be able to use the same speech vocoder as the one used in the mentioned speech to speech translation task, we perform the translation in two steps: first a text to acoustic unit conversion and then speech generation through the same HiFi-GAN as described in the previous experiments.

Models In a similar way, as done in the speech to speech translation tasks, we use an encoder-decoder architecture to perform text to speech translation as Figure 33 depicts. Considering that converting text inputs to acoustic units can be considered as a machine translation task, we use a pre-trained text model as initialization for our encoder-decoder architecture. Namely, we consider MBart model in its two variations, MBart25 and MBart50 [38, 37]. After initialization, the full architecture is finetuned on the text to acoustic unit translation task. Finally, the same HiFi GAN unit to speech vocoder as the one described in section 5.5.4 is applied to

generate speech utterances.

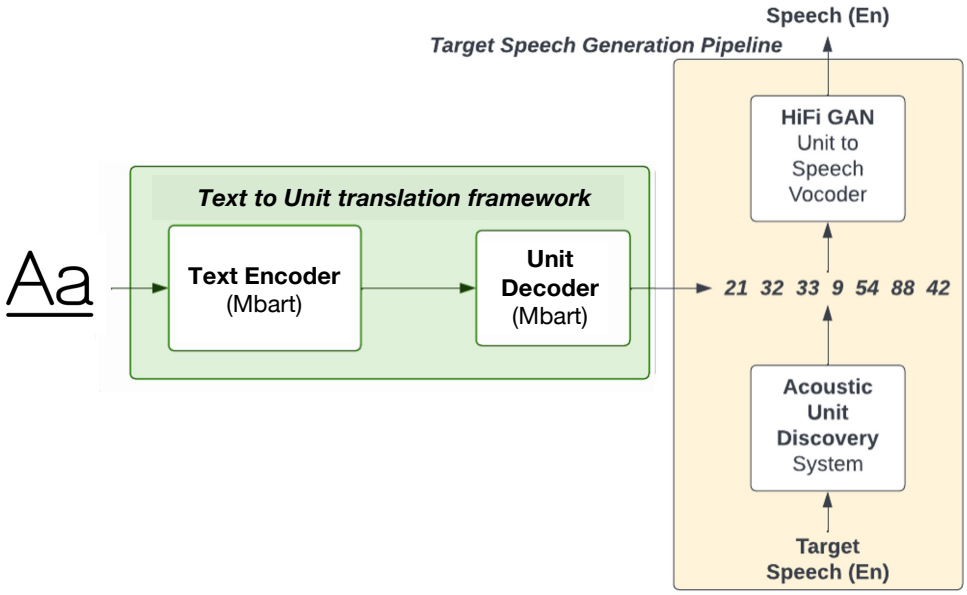


Figure 33: Text to speech translation system.

Data Considering that the CVSS dataset [28], already used in the direct speech to speech translation task, also provides the text transcription for the input audios, we use this dataset to perform a 21 to EN text to speech translation task. Target audios are forwarded through the acoustic unit discovery system to obtain its unit representations. These acoustic units are then used as targets to train the text to unit translation system.

Evaluation Metrics Same evaluation protocol as described in subsection 5.5.4 applies for the text to speech translation system. An ASR system is run to obtain transcriptions for the output audio signals and BLEU score is computed between those transcriptions and the normalised text references.

Results Figure 34 presents the BLEU scores in the test partition of the CVSS dataset for our proposed text-to-speech system. In this figure, the performance is shown separately for high, medium and low resource languages. Moreover, the average of the results is also presented. These results show a large performance improvement in all splits when the MBart50 model is used as a pre-training model to initialize our encoder-decoder pipeline for the text-to-speech system.

For a more in-depth analysis of the differences found between the two types of models employed, we can see the results for each language of the 21 languages available in the CVSS dataset in Figure 35. This figure shows that the performance of each language improves using MBart50, but the improvement achieved in the languages which are not included in MBart25, marked with 1 in the figure, is particularly remarkable. Note that even languages that are not included in either MBart25 or MBart50, marked with 2, benefit from the influence of having more languages in the second model and improve their results.

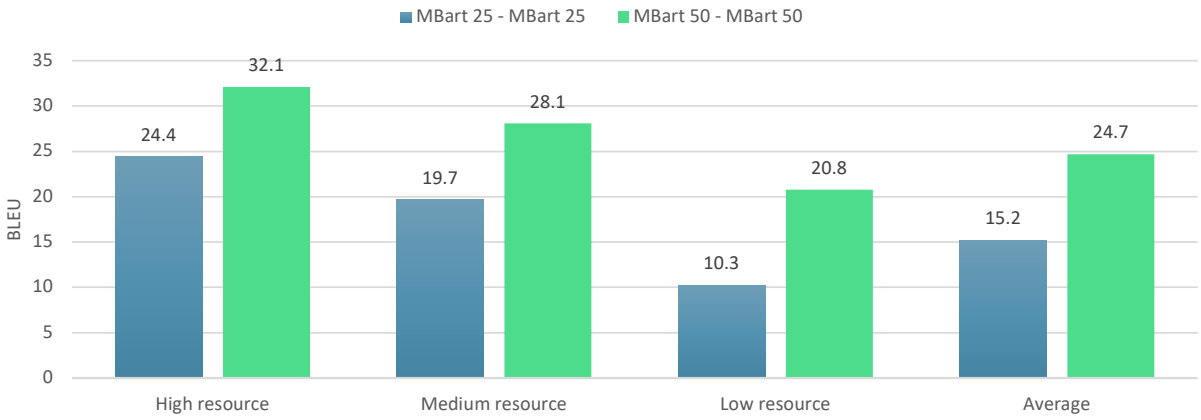


Figure 34: BLEU results on CVSS data test partition.

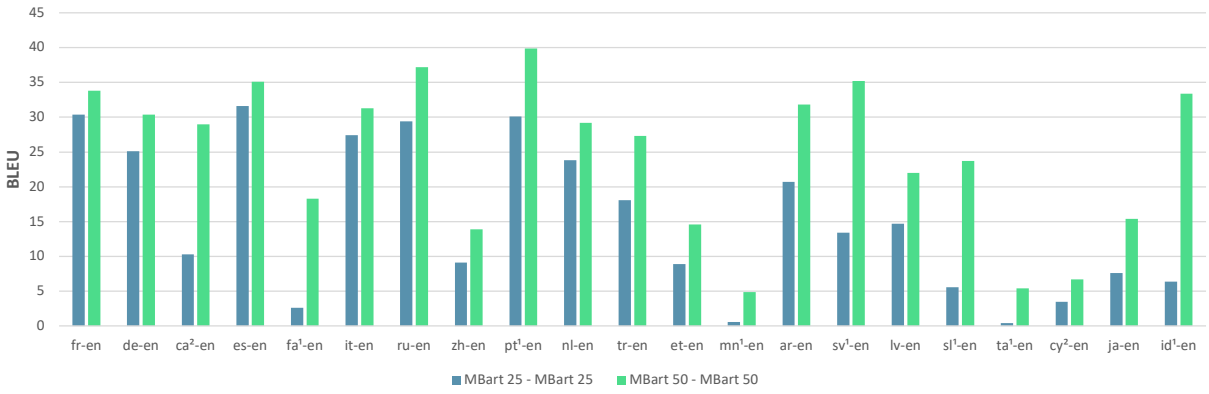


Figure 35: BLEU results on CVSS data test partition for each language available.
[1] *Languages not present in mBART-25, but present in mBART-50.*
[2] *Languages not present in mBART-25 or mBART-50.*

6 Evaluating Speech-to-speech translation without text

Our goal is to develop a metric in order to compare a speech hypothesis (*H*) with a speech reference (*R*) along several axes. Main axis is meaning, i.e similarity score should be high if both utterances convey same message. But other axes are interesting: eg. high similarity if *H* and *R* voices are similar (similar speaker, gender, etc.). Our textless metric should have a strong correlation with a conventional similarity metric applied to the transcripts of *H* and *R*.

Such textless metric could be interesting for: (a) evaluating a S2S translation system w/o falling back on a transcription of *H* and *R*; (b) some languages (eg. Tamasheq) for which we cannot fall back to a transcription (>50% of languages have no written form); and (c) defining a training objective for S2S model learning.

6.1 A BLEU for speech

One approach we tried is a baseline approach that tries to leverage text-based translation metrics into speech-based metrics. The idea is to generate audio symbols based on the clustering

of audio features, and use those symbols as pseudo-words in standard MT metrics such as BLEU, WER or TER. If the metric based on speech correlates with the same metric on the transcribed text then the approach works.

The training setup is thus:

- Generate features from audio
- Build *n* k-means cluster centers from those features

The application setup is then:

- Generate features from the two audio sentences to compare
- Map each feature to a k-means cluster, creating audio symbols
- Reduce consecutive repetitions of the same symbol into one instance (de-duplication)
- Compare the two symbol streams using the MT metric

And the validation is done by comparing the metric results on audio and on text on sentence pairs.

The experimental corpus used is commonvoice 4.0 english. It is originally an ASR/TTS corpus, with sentences both as text and speech. We normalized and tokenized the transcriptions and pairwise compared them to find pairs with at least one quadrigram in common. That condition ensures that the BLEU score is non-zero, and thus the sentences are not too far away to the point of the comparison being meaningless. 31M pairs were left, giving a little over 850K kept sentences. The standard metrics (WER, TER, BLEU, chrF) were then computed on those pairs.

For creating the clusters the experience plan included:

- Features: cepstrum or wav2vec2
- Distance: l2 or cosine
- Count: 10 to 10000 (10, 20, 50, etc)

Not every combination was done within the duration of JSALT but we believe the initial results are representative.

To get a quick "eyeball" estimation of the correlations obtained we created scatterplots with the text BLEU on X and the audio BLEU on Y. The Figure 36 includes example of the results.

A good correlation would put all the points near the diagonal. We can see that the result is essentially random. The lower audio BLEU scores with higher number of clusters also confirm a quasi-randomness of the audio symbol streams.

The conclusion is that this approach, while seductive, fails are producing useful results. Something more elaborate is needed, such as a learned metric as presented in the next section.

6.2 A learnt metric (extending the COMET approach)

As we have seen that text-based metrics applied to discretized speech units coming from HuBERT or XLSR models are not reliable, we believe that learning a metric that will capture semantic similarity (or all kinds of similarities) between audio hypothesis and reference is an

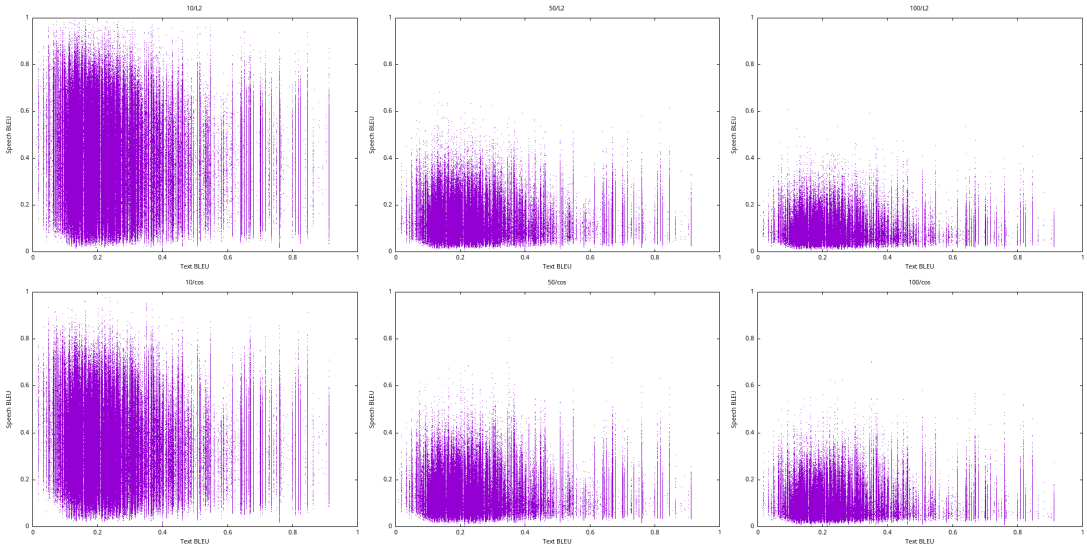


Figure 36: Scatterplots with 10/50/100 clusters and l2 distance (top) or cosine distance (bottom).

interesting option. To quickly experiment with this idea, we re-used the COMET[10] framework widely used in machine translation evaluation. COMET [49] can be used for evaluating machine translation systems with currently available high-performing metrics (trained to correlate with human judgements) but also to train and develop new metrics.

We adapt the COMET framework to our textless metric as illustrated in figure 37: both audio *H* and *R* are transformed in a sequence of speech units (we will use HuBERT [26] in our preliminary experiments). Both sequences of discrete units are then mapped into a sequence of characters and encoded with XLM-Roberta [16], and then pooled in a single vector. A regression layer will then predict the true (text-based) metric we want to approximate. In our experiments we will use ChrF metric [46] as a target and a mean square error loss will be used to train the model parameters. It is important to note that, as done in initial COMET approach, not only the regression layer parameters will be learnt during training but also some parameters of the XLM-Roberta encoder (after 30% of the first epoch and for the rest of the training).

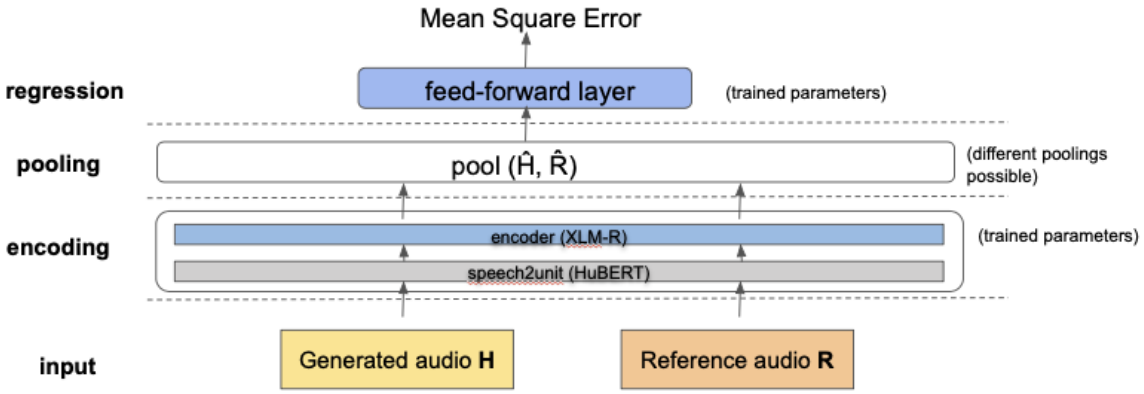


Figure 37: Adaptation of the COMET framework to our textless metric.

We first experiment on the CVSS corpus (English target part) [28]. All *H* and *R* are synthetic

[10] https://github.com/Unbabel/COMET

Input	Train Data	Encoder	Epochs	Metric	ρ (Pearson)	ρ (Spearman)
Text	None	None	-	chrF	1	1
Text	14.7k utt.	XLM-R	5	learnt chrF	0.902	0.922
Audio	None	Hubert-50	-	chrF	0.431	0.386
Audio	14.7k utt.	Hubert-50 +XLM-R	5	learnt chrF	0.542	0.480
Audio	14.7k utt.	Hubert-200 +XLM-R	5	learnt chrF	0.595	0.567
Audio	207.4k utt.	Hubert-200 +XLM-R	5	learnt chrF	0.755	0.700
Audio	207.4k utt.	Hubert-200 +XLM-R	10	learnt chrF	**0.779**	**0.733**

Table 25: Correlations between true ChrF from text and learnt ChrF from audio, for different experimental setups.

speech from different speaker voices. They correspond to similar or (slightly) dissimilar transcripts as illustrated by the ChrF distribution displayed in figure 38 (left). To obtain dissimilar audios with different voices, we applied the following process to our original CVSS speech utterances: (a) ASR transcription; (b) BART encoding and decoding;[11] and (c) TTS from the noisy transcript (with a different speaker voice).

Our first experiments are summarized in table 25 and show correlations between true ChrF from text and learnt ChrF from audio for different experimental setups. The first two lines display our target (text-based ChrF) and topline (learnt ChrF from text) respectively. The remaining columns use audio *H* and *R* inputs: third column is our baseline (ChrF is computed from sequences of HuBERT units) while lines 4-7 display results obtained with our learnt metric. Overall we observe that more speech units (200 instead of 50) improves correlation and that adding training data (207k utterances instead of 14.7k utterances) improves learnt metric as well. Finally our best result on this synthetic dataset is obtained while training longer (10 epochs instead of 5).

Figure 38 (right) displays the distribution of our learnt ChrF scores (with the best configuration; last line of table 25 obtained from audio). We can observe that both distributions are very similar and that COMET has learnt, from training data, to reproduce the scores distribution using audio input only!

As a final experiment, we also evaluated on natural speech: we use common voice initial corpus (English part) where *H* and *R* are natural speech utterances and most of the time from different speakers. They correspond to similar or dissimilar transcripts pairs and those pairs were selected based on minimum n-gram similarities. Our target is again the ChrF metric obtained from text. We train our learnt textless ChrF on 2M pairs of audio utterances (using 200 HuBERT units) and evaluate on 100k pairs of utterances for which ChrF is known. The training loss is displayed on figure 39 and we obtain very good correlations coefficients on the dev set: $\rho(Pearson) = 0.970$ and $\rho(Spearman) = 0.866$.[12] From the loss curve, we clearly see

[11] To further add noise.

[12] Training on more data (18M utterances instead of 2M) lead to small improvements of correlation coefficients: $\rho(Pearson) = 0.99$ and $\rho(Spearman) = 0.88$.

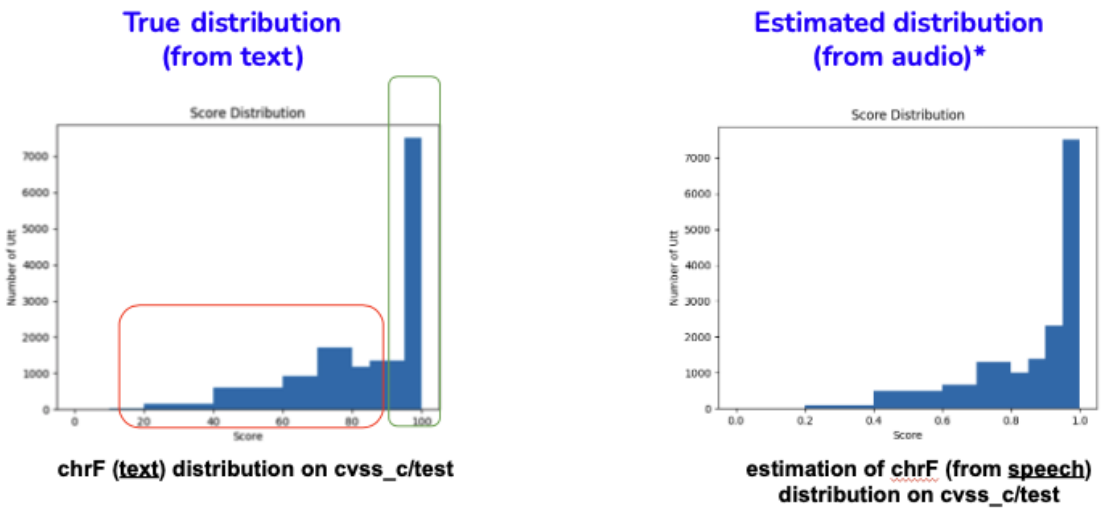


Figure 38: Distributions of ChrF scores on the test set of our prepared CVSS corpus: (left) true ChrF from text and (right) learnt ChrF from audio.

the moment where parameters of the XLM-Roberta encoder (after 30% of the first epoch) start to be adapted in addition to the regression layer parameters: at this moment XLM-R specializes itself at encoding HuBERT units (instead of characters).

Overall, this demonstrates that our approach also works to compare meaning of natural speech utterances.

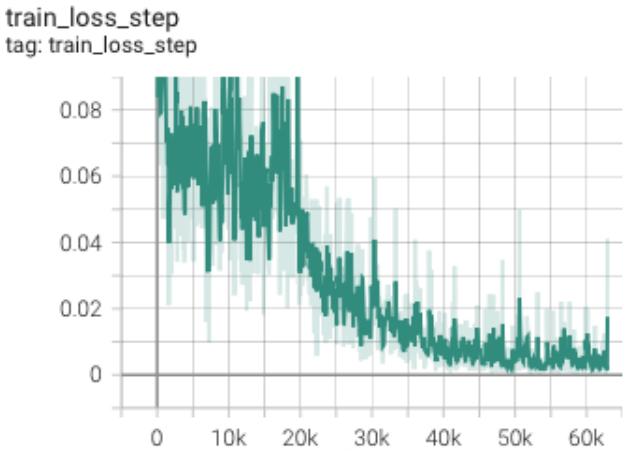


Figure 39: Training loss during 1st epoch of training textless ChrF on 2M natural audio utterances .

7 Acknowledgment

This work was granted access to the HPC resources of IDRIS under the allocation 2022-A0131012565 made by GENCI.

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101007666.

8 Dissemination

https://esperanto.univ-lemans.fr
https://www.clsp.jhu.edu/speech-translation-for-under-resourced-languages/

Bibliography

[1] Bhuvan Agrawal et al. “Tie your embeddings down: Cross-modal latent spaces for end-to-end spoken language understanding”. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 7157–7161.

[2] Antonios Anastasopoulos et al. “Findings of the IWSLT 2022 Evaluation Campaign”. In: *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*. Dublin, Ireland: ACL, May 2022, pp. 98–157. DOI: 10.18653/v1/2022.iwslt-1.10. URL: https://aclanthology.org/2022.iwslt-1.10.

[3] Junyi Ao et al. “Speecht5: Unified-modal encoder-decoder pre-training for spoken language processing”. In: *arXiv preprint arXiv:2110.07205* (2021).

[4] Rosana Ardila et al. “Common voice: A massively-multilingual speech corpus”. In: *arXiv preprint arXiv:1912.06670* (2019).

[5] Arun Babu et al. “XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale”. In: *arXiv preprint arXiv:2111.09296* (2021).

[6] Arun Babu et al. “XLS-R: Self-supervised cross-lingual speech representation learning at scale”. In: *arXiv preprint arXiv:2111.09296* (2021).

[7] Alexei Baevski et al. “wav2vec 2.0: A framework for self-supervised learning of speech representations”. In: *arXiv preprint arXiv:2006.11477* (2020).

[8] Collin F Baker, Charles J Fillmore, and John B Lowe. “The berkeley framenet project”. In: *COLING 1998 Volume 1: The 17th International Conference on Computational Linguistics*. 1998.

[9] Hélene Bonneau-Maynard, Matthieu Quignard, and Alexandre Denis. “MEDIA: a semantically annotated corpus of task oriented dialogs in French”. In: *Language Resources and Evaluation* 43.4 (2009), pp. 329–354.

[10] Lyle Campbell. *Ethnologue: Languages of the world*. 2008.

[11] Roldano Cattoni et al. “Must-c: A multilingual corpus for end-to-end speech translation”. In: *Computer Speech & Language* 66 (2021), p. 101155.

[12] Sanyuan Chen et al. “Wavlm: Large-scale self-supervised pre-training for full stack speech processing”. In: *IEEE Journal of Selected Topics in Signal Processing* (2022).

[13] Sanyuan Chen et al. “Why does Self-Supervised Learning for Speech Recognition Benefit Speaker Recognition?” In: *arXiv preprint arXiv:2204.12765* (2022).

[14] Zhengyang Chen et al. “Large-scale self-supervised speech representation learning for automatic speaker verification”. In: *ICASSP 2022-2022 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2022, pp. 6147–6151.

[15] Alexis Conneau and Guillaume Lample. “Cross-lingual Language Model Pretraining”. In: *Advances in Neural Information Processing Systems*. Ed. by H. Wallach et al. Vol. 32. Curran Associates, Inc., 2019. URL: https://proceedings.neurips.cc/paper/2019/file/c04c19c2c2474dbf5f7ac4372c5b9af1-Paper.pdf.

[16] Alexis Conneau et al. “Unsupervised Cross-lingual Representation Learning at Scale”. In: *CoRR* abs/1911.02116 (2019). arXiv: 1911.02116. URL: http://arxiv.org/abs/1911.02116.

[17] Alexis Conneau et al. “Unsupervised Cross-lingual Representation Learning at Scale”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 8440–8451. DOI: 10.18653/v1/2020.acl-main.747. URL: https://aclanthology.org/2020.acl-main.747.

[18] Nauman Dawalatabad et al. “ECAPA-TDNN embeddings for speaker diarization”. In: *Interspeech*. 2021.

[19] Thierry Desot, François Portet, and Michel Vacher. “Corpus generation for voice command in smart home and the effect of speech synthesis on End-to-End SLU”. In: *12th Conference on Language Resources and Evaluation (LREC 2020)*. 2020, pp. 6395–6404.

[20] Brecht Desplanques, Jenthe Thienpondt, and Kris Demuynck. “ECAPA-TDNN: Emphasized Channel Attention, Propagation and Aggregation in TDNN based speaker verification”. In: *Interspeech*. 2020.

[21] Xiangyu Duan et al. “Bilingual Dictionary Based Neural Machine Translation without Using Parallel Sentences”. In: *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, July 2020, pp. 1570–1579. DOI: 10.18653/v1/2020.acl-main.143. URL: https://aclanthology.org/2020.acl-main.143.

[22] Fangxiaoyu Feng et al. “Language-agnostic BERT Sentence Embedding”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022, pp. 878–891.

[23] Fangxiaoyu Feng et al. “Language-agnostic bert sentence embedding”. In: *arXiv preprint arXiv:2007.01852* (2020).

[24] Abel Gonzalez-Garcia, Joost Van De Weijer, and Yoshua Bengio. “Image-to-image translation for cross-domain disentanglement”. In: *NeurIPS* (2018).

[25] Wenzhong Guo, Jianwen Wang, and Shiping Wang. “Deep Multimodal Representation Learning: A Survey”. In: *IEEE Access* 7 (2019), pp. 63373–63394. DOI: 10.1109/ACCESS.2019.2916887.

[26] Wei-Ning Hsu et al. “Hubert: Self-supervised speech representation learning by masked prediction of hidden units”. In: *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 29 (2021), pp. 3451–3460.

[27] Yinghui Huang et al. “Leveraging unpaired text data for training end-to-end speech-to-intent systems”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 7984–7988.

[28] Ye Jia et al. “CVSS Corpus and Massively Multilingual Speech-to-Speech Translation”. In: *CoRR* abs/2201.03713 (2022). arXiv: 2201.03713. URL: https://arxiv.org/abs/2201.03713.

[29] Ye Jia et al. “Direct Speech-to-Speech Translation with a Sequence-to-Sequence Model”. In: *Proc. Interspeech 2019* (2019), pp. 1123–1127.

[30] Ye Jia et al. “Translatotron 2: High-quality direct speech-to-speech translation with voice preservation”. In: *International Conference on Machine Learning*. PMLR. 2022, pp. 10120–10134.

[31] Melvin Johnson et al. “Google’s Multilingual Neural Machine Translation System: Enabling Zero-Shot Translation”. In: *Transactions of the Association for Computational Linguistics* 5 (2017), pp. 339–351. DOI: 10.1162/tacl_a_00065. URL: https://aclanthology.org/Q17-1024.

[32] Sameer Khurana, Antoine Laurent, and James Glass. “SAMU-XLSR: Semantically-Aligned Multimodal Utterance-level Cross-Lingual Speech Representation”. In: *IEEE Journal of Selected Topics in Signal Processing* (2022). DOI: 10.1109/JSTSP.2022.3192714.

[33] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. “Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis”. In: *Advances in Neural Information Processing Systems* 33 (2020), pp. 17022–17033.

[34] Taku Kudo and John Richardson. “SentencePiece: A simple and language independent subword tokenizer and detokenizer for Neural Text Processing”. In: *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*. Brussels, Belgium: Association for Computational Linguistics, Nov. 2018, pp. 66–71. DOI: 10.18653/v1/D18-2012. URL: https://aclanthology.org/D18-2012.

[35] Ann Lee et al. “Direct Speech-to-Speech Translation With Discrete Units”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2022, pp. 3327–3339.

[36] Fabrice Lefèvre et al. “Leveraging study of robustness and portability of spoken language understanding systems across languages and domains: the PORTMEDIA corpora”. In: *Proceedings of the Eighth International Conference on Language Resources and Evaluation (LREC’12)*. 2012, pp. 1436–1442.

[37] Xian Li et al. “Multilingual Speech Translation from Efficient Finetuning of Pretrained Models”. In: *ACL/IJCNLP (1)*. 2021.

[38] Yinhan Liu et al. “Multilingual denoising pre-training for neural machine translation”. In: *Transactions of the Association for Computational Linguistics* 8 (2020), pp. 726–742.

[39] Loren Lugosch et al. “Using speech synthesis to train end-to-end spoken language understanding models”. In: *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2020, pp. 8499–8503.

[40] Manon Macary et al. “On the use of self-supervised pre-trained acoustic and linguistic features for continuous speech emotion recognition”. In: *2021 IEEE Spoken Language Technology Workshop (SLT)*. IEEE. 2021, pp. 373–380.

[41] Markus Müller et al. “In Pursuit of Babel-Multilingual End-to-End Spoken Language Understanding”. In: *2021 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*. IEEE. 2021, pp. 1042–1049.

[42] Arsha Nagrani, Joon Son Chung, and Andrew Zisserman. “Voxceleb: a large-scale speaker identification dataset”. In: *Interspeech*. 2017.

[43] Vassil Panayotov et al. “Librispeech: an asr corpus based on public domain audio books”. In: *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2015, pp. 5206–5210.

[44] Kishore Papineni et al. “Bleu: a method for automatic evaluation of machine translation”. In: *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*. 2002, pp. 311–318.

[45] Leonardo Pepino, Pablo Riera, and Luciana Ferrer. “Emotion recognition from speech using wav2vec 2.0 embeddings”. In: *arXiv preprint arXiv:2104.03502* (2021).

[46] Maja Popović. “chrF: character n-gram F-score for automatic MT evaluation”. In: *Proceedings of the Tenth Workshop on Statistical Machine Translation*. Lisbon, Portugal: Association for Computational Linguistics, Sept. 2015, pp. 392–395. DOI: 10.18653/v1/W15-3049. URL: https://aclanthology.org/W15-3049.

[47] Soujanya Poria et al. “MELD: A Multimodal Multi-Party Dataset for Emotion Recognition in Conversations”. In: *Association for Computational Linguistics (ACL)*. 2019.

[48] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. “Waveglow: A flow-based generative network for speech synthesis”. In: *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE. 2019, pp. 3617–3621.

[49] Ricardo Rei et al. “COMET: A Neural Framework for MT Evaluation”. In: *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2685–2702. DOI: 10.18653/v1/2020.emnlp-main.213. URL: https://aclanthology.org/2020.emnlp-main.213.

[50] Ramon Sanabria et al. “How2: A Large-scale Dataset For Multimodal Language Understanding”. In: *Proceedings of the Workshop on Visually Grounded Interaction and Language (ViGIL)*. NeurIPS. 2018. URL: http://arxiv.org/abs/1811.00347.

[51] Ramon Sanabria et al. “Measuring the Impact of Individual Domain Factors in Self-Supervised Pre-Training”. In: *arXiv preprint arXiv:2203.00648* (2022).

[52] Jonathan Shen et al. “Natural tts synthesis by conditioning wavenet on mel spectrogram predictions”. In: *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE. 2018, pp. 4779–4783.

[53] Jenthe Thienpondt, Brecht Desplanques, and Kris Demuynck. “The IDLab Short-duration Speaker Verification Challenge 2020 System”. In: ().

[54] Jenthe Thienpondt, Brecht Desplanques, and Kris Demuynck. “The IDLab VoxSRC-20 submission: Large margin fine-tuning and quality-aware score calibration in DNN based speaker verification”. In: *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. 2021.

[55] Gokhan Tur and Renato De Mori. *Spoken language understanding: Systems for extracting semantic information from speech*. John Wiley & Sons, 2011.

[56] Changhan Wang, Anne Wu, and Juan Pino. “CoVoST 2 and Massively Multilingual Speech-to-Text Translation”. In: *arXiv e-prints* (2020), arXiv–2007.

[57] Xinyi Wang, Sebastian Ruder, and Graham Neubig. “Expanding Pretrained Models to Thousands More Languages via Lexicon-based Adaptation”. In: *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Dublin, Ireland: Association for Computational Linguistics, May 2022, pp. 863–877. DOI: 10.18653/v1/2022.acl-long.61. URL: https://aclanthology.org/2022.acl-long.61.

Esperanto
Exchanges for SPEech
ReseArch aNd TechnOlogies
Horizon 2020 project

Consortium

Disclaimer

All information provided reflects the status of the ESPERANTO project at the time of writing and may be subject to change.

Neither the ESPERANTO Consortium as a whole, nor any single party within the ESPERANTO Consortium warrant that the information contained in this document is capable of use, nor that the use of such information is free from risk. Neither the ESPERANTO Consortium as a whole, nor any single party within the ESPERANTO Consortium accepts any liability for loss or damage suffered by any person using the information.

This document does not represent the opinion of the European Community, and the European Community is not responsible for any use that might be made of its content.

Copyright Notice

This project has received funding from the European Union's Horizon 2020 research and innovation programme under the Marie Skłodowska-Curie grant agreement No 101007666