Speech Translation with Foundation Models and Optimal Transport: UPC at IWSLT23

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa
Universitat Politècnica de Catalunya, Barcelona
{ioannis.tsiamas,gerard.ion.gallego,jose.fonollosa}@upc.edu

Marta R. Costa-jussà
Meta AI, Paris
costajussa@meta.com

Abstract

This paper describes the submission of the UPC Machine Translation group to the IWSLT 2023 Offline Speech Translation task. Our Speech Translation systems utilize foundation models for speech (wav2vec 2.0) and text (mBART50). We incorporate a Siamese pretraining step of the speech and text encoders with CTC and Optimal Transport, to adapt the speech representations to the space of the text model, thus maximizing transfer learning from MT. After this pretraining, we fine-tune our system end-to-end on ST, with Cross Entropy and Knowledge Distillation. Apart from the available ST corpora, we create synthetic data with SegAugment to better adapt our models to the custom segmentations of the IWSLT test sets. Our best single model obtains 31.2 BLEU points on MuST-C tst-COMMON, 29.8 points on IWLST.tst2020 and 33.4 points on the newly released IWSLT.ACLdev2023.

1 Introduction

In the past decade, the field of Speech Translation (ST) has seen significant advancements, mainly due to end-to-end models that directly translate speech, offering a more efficient method compared to traditional cascade systems (Sperber and Paulik, 2020). Despite data availability challenges, recent progress has diminished the performance disparity between these approaches (Bentivogli et al., 2021; Potapczyk and Przybysz, 2020; Inaguma et al., 2021; Ansari et al., 2020). Critical to the advancements in end-to-end models is the exploitation of ASR and MT data through pretraining strategies (Berard et al., 2018; Pino et al., 2019; Di Gangi et al., 2019; Gangi et al., 2019; Wang et al., 2020a; Zhang et al., 2020; Bansal et al., 2019).

Recently, Le et al. (2023) proposed a method to effectively utilize both ASR and MT pretraining to enhance ST. This approach involves pretraining an encoder-decoder MT system with available text data, followed by pretraining a speech encoder to generate representations similar to the MT system's encoder (*Siamese pretraining*) using Connectionist Temporal Classification (CTC) supervision (Graves et al., 2006) and Optimal Transport (Peyré and Cuturi, 2019). The resulting speech encoder and text decoder can be fine-tuned with ST data.

Another way of incorporating ASR and MT is to leverage large pretrained speech and text models as a foundation for end-to-end ST systems (Li et al., 2021; Gállego et al., 2021; Han et al., 2021; Zhang and Ao, 2022; Pham et al., 2022; Tsiamas et al., 2022b). However, these systems encounter representation discrepancy issues, which can hinder the full exploitation of pretrained foundation models. Gállego et al. (2021); Zhao et al. (2022) aimed to address this by adding *coupling modules* after the pretrained encoder, while other focus on solving the length discrepancies (Zhang et al., 2020; Xu et al., 2021a; Gaido et al., 2021). Han et al. (2021) tackled the issue by projecting speech and text features to a common semantic space using attention mechanisms and semantic memories.

In our work, we tackle the issue of misaligned speech and text encoder representations by adopting the approach proposed by Le et al. (2023). Our system uses a speech foundation model fine-tuned on English ASR, wav2vec 2.0 (Baevski et al., 2020), and an MT foundation model fine-tuned on multilingual MT (En-Xx), mBART50 (Tang et al., 2020), as described in Section 2.1. Building on prior research (Xu et al., 2021a; Han et al., 2021), we employ two encoders: an acoustic encoder from wav2vec 2.0 and a semantic encoder from mBART50. Coupling modules link these encoders to address length discrepancy. We extend Le et al. (2023) by applying CTC and OT losses to the outputs of the acoustic and semantic encoders, respectively, add a second auxiliary OT loss for the inputs of the semantic encoder, and keep the text encoder frozen to keep the MT space intact. This method aligns the speech encoder's represen-

Proceedings of the 20th International Conference on Spoken Language Translation (IWSLT 2023), pages 397–410
July 13-14, 2023 ©2023 Association for Computational Linguistics

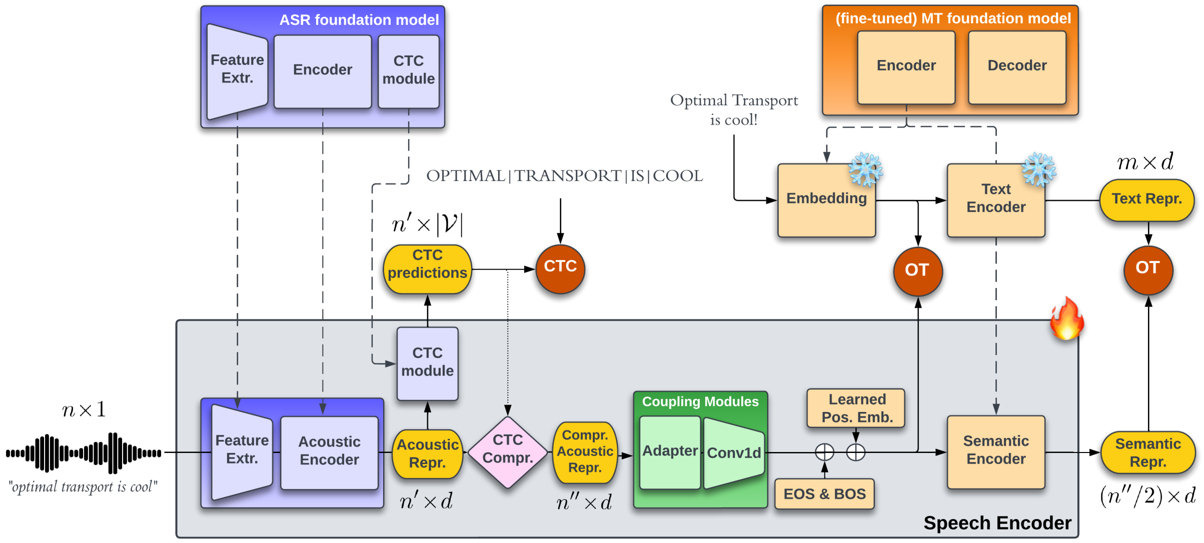

Figure 1: Extended Siamese pretraining

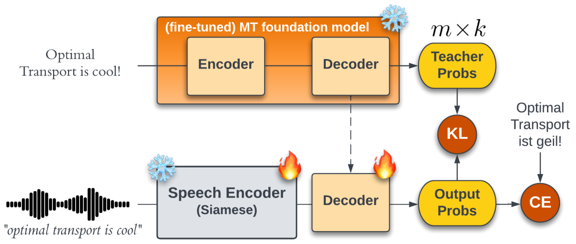
(fine-tuned) MT foundation model
Optimal Transport is cool!
Encoder
Decoder
$m \times k$
Teacher Probs
KL
Optimal Transport ist geil!
Speech Encoder (Siamese)
Decoder
Output Probs
CE
"optimal transport is cool"

Figure 2: Speech Translation fine-tuning

tations with the MT foundation model, effectively improving the final ST system's performance by mitigating representation mismatch.

In summary, we participate in the IWSLT 2023 Offline Speech Translation task, focusing on translating spoken English to written German, by employing an end-to-end system. We leverage ASR and MT foundation models with the Siamese pretraining approach, to effectively bring their encoder's representations closer. We furthermore decouple acoustic and semantic modeling in our speech encoder, adjust for the length miss-match between speech and text with several coupling modules, and apply *knowledge distillation* (Hinton et al., 2015) from MT (Liu et al., 2019; Gaido et al., 2020), using mBART50.

2 Methodology

Our system, an encoder-decoder transformer, leverages ASR and MT foundation models (§2.1). We initially train the speech encoder with an Extended Siamese pretraining (§2.2), and then fine-tune it with the MT decoder for end-to-end ST (§2.3).

2.1 System architecture

As depicted in Figures 1 and 2, the encoder of our system is composed of several interconnected modules, while the decoder is adopted directly from the MT foundation model. The speech encoder is designed to generate representations closely resembling those of the MT foundation model, ensuring better compatibility between them. The following paragraphs provide a detailed overview of its key components and their functions.

Acoustic Modeling The speech waveform $x \in \mathbb{R}^n$ is first processed by a feature extractor, which consists of several strided convolutional layers, downsampling the input to a length of n'. Following, a Transformer encoder with dimensionality d is responsible for the acoustic modeling. Both these modules are initialized from an ASR foundation model.

CTC Compression The obtained acoustic representation $h \in \mathbb{R}^{n' \times d}$ is passed through a linear layer (initialized from the ASR model) and a softmax to generate the ASR vocabulary predictions $p^{(ctc)} \in \mathbb{R}^{n' \times |\mathcal{V}|}$, where $\mathcal{V}$ is the size of the vocabulary. We apply CTC compression (Gaido et al., 2021) to the acoustic representation, averaging the representations corresponding to repeating predictions on $p^{(ctc)}$ and removing those associated with the blank token. This process results in a new compressed representation $h^{(compr)} \in \mathbb{R}^{n'' \times d}$, where n'' denotes the compressed length of the sequence. This compression helps to reduce the length discrepancy between speech and text representations, which, in turn, facilitates the alignment process during Siamese pretraining (§2.2).

Coupling Modules Next, we apply an *adapter* (Houlsby et al., 2019), consisting of a linear projection to $8d$, a non-linear activation, a linear projection back to d. This module serves to (1) process the collapsed representations resulting from the compression and (2) provide sufficient parameters between the CTC and first OT loss to decouple their influence (§2.2). After the adapter we apply a strided 1D Convolution that subsamples the sequence by a factor of 2, which can help transform it closer to a sub-word level representation, rather than a character-level one, and subsequently aid in the Optimal Transport training with the sub-word level representation from the text encoder (§2.2).

Semantic Modeling At this point, we modify the representation to better match the input expected by the MT encoder. This is achieved by prepending and appending special tokens that correspond to the BOS and EOS tokens used in MT. We also re-introduce positional information to the representation with learned positional embeddings. Both the special tokens $t^{bos}, t^{eos} \in \mathbb{R}^d$ and the positional embeddings $E^{pos} \in \mathbb{R}^{(M+2) \times d}$ (with M representing the maximum sequence length) are learnable parameters initialized from the MT foundation model. The motivation is to bring the representation closer to the text embedding from the MT model, facilitating OT loss convergence (§2.2). Finally, the representation is processed by several more transformer encoder layers, which are initialized from the MT model and are responsible for semantic modeling.

2.2 Siamese pretraining

Our approach builds upon the Siamese pretraining proposed by Le et al. (2023), which exploits both ASR and MT pretraining to improve ST performance. This approach involves pretraining the encoder of an ST system jointly with Connectionist Temporal Classification (CTC) and Optimal Transport (OT), bringing its representations close to those of an MT encoder. This pretraining strategy has demonstrated superior results compared to traditional ASR pretraining with encoder-decoder and Cross-Entropy (Le et al., 2023). In this work, we build upon the method of Le et al. (2023) in several ways. First, we decouple the CTC and OT losses to correspond to the acoustic and semantic representations. Second, we add an extra auxiliary OT loss to better adapt the input to the semantic encoder. Next, we also employ CTC-based compression and coupling modules to better align the length of speech features with corresponding sub-word text representations. Finally, we opt to freeze the text encoder to not modify the MT decoder's representation space. The extended Siamese pretraining scheme is illustrated in Figure 1. For brevity, we refer to it simply as "Siamese" throughout the rest of the paper.

The Siamese pretraining is supervised by a combination of loss functions, each serving a distinct purpose. The CTC loss ensures the performance of the acoustic modeling by applying to the predictions of the CTC module. Meanwhile, the two OT losses target the input and output of the semantic encoder, and aim to align them with the text encoder representations. We calculate the OT loss as the Wasserstein distance (Frogner et al., 2015) between the text and speech representations, using an upper bound approximation, which is efficiently evaluated by the Sinkhorn algorithm (Knopp and Sinkhorn, 1967). Since the Wasserstein distance is position invariant, we follow (Le et al., 2023), and apply positional encodings, to make it applicable to sequences. The combined loss function for the Siamese pretraining stage is given by:

$$\mathcal{L}^{siamese} = \alpha\,\mathcal{L}^{CTC} + \beta\,\mathcal{L}^{OT_1} + \gamma\,\mathcal{L}^{OT_2} \quad (1)$$

Where α, β, and γ are hyperparameters that control the relative importance of each loss component in the combined pretraining loss.

2.3 Speech Translation fine-tuning

Upon obtaining the encoder from §2.2, we utilize it to initialize our ST system's encoder, while using the MT foundation model to initialize the decoder (Fig. 2). In addition to the Cross Entropy loss, we optionally provide guidance for the ST training through Knowledge Distillation (KD) (Tan et al., 2019), using the MT foundation model as a teacher. Specifically, we only use the top-k predictions rather than the entire distribution, and soften them using a temperature T (Gaido et al., 2020).

Since CTC supervision is not employed at this stage, we freeze the Feature Extractor, Acoustic Encoder, and CTC module from our encoder. During training, we optimize the parameters of the ST system's encoder and decoder with respect to the combined loss function, which is the sum of the Cross Entropy loss and the optional KD loss:

$$\mathcal{L}^{ST} = \lambda \, \mathcal{L}^{CE} + (1 - \lambda) \, \mathcal{L}^{KL} \tag{2}$$

Where $\mathcal{L}^{CE}$ is the Cross Entropy loss, $\mathcal{L}^{KL}$ is the Kullback–Leibler divergence between the MT and ST output distributions, and $0 \leq \lambda \leq 1$ is a hyperparameter that controls the relative importance of each loss component in the combined ST loss.

3 Data

3.1 Datasets

To train our ST models we used data from three speech translation datasets, MuST-C v3 (Cattoni et al., 2021), Europarl-ST (Iranzo-Sánchez et al., 2020) and CoVoST-2 (Wang et al., 2020b). MuST-C is based on TED talks, Europarl-ST on the European Parliament proceedings, and CoVoST is derived from the Common Voice dataset (Ardila et al., 2020). Their statistics are available in the first part of Table 1. We use as development data the IWSLT test sets of 2019 and 2020 (Niehues et al., 2019; Ansari et al., 2020), which are based on TED talks, and the ACL development set of 2023, which contains 5 presentations from ACL 2022. All development data are unsegmented, meaning that they are long and continuous speeches. We apply SHAS segmentation (§5) before translating them. For the Siamese pretraining, we used the English ASR data from MuST-C v3 and Europarl-ST, as well as CommonVoice v11 (Ardila et al., 2020) (Table 1).

3.2 Data Augmentation

We employ data augmentation, to create more ST data for training our models (Table 1). We use the MT foundation model, to translate the transcript of English CommonVoice v11 (Ardila et al., 2020). Since CommonVoice data contains various accents, we expect the synthetic data will be helpful for translating the ACL talks domain, which has predominantly non-native English accents. We additionally utilize SegAugment (Tsiamas et al., 2022a), which creates alternative versions of the training data by segmenting them differently with SHAS (Tsiamas et al., 2022c). We apply SegAugment to MuST-C v3, with three different length parameterizations: *medium (m)* (3 to 10 seconds), *long (l)* (10 to 20 seconds), and *extra-long (xl)* (20 to 30 seconds). We expect that SegAugment will be beneficial for translating the SHAS-segmented test sets, due to the similar segmentations of the training data it provides, as shown in Tsiamas et al. (2022a).

	Original	**Siamese**	**ST**
ST datasets			
MuST-C v3	427	417	421
↪ SegAugment	$1,364^\dagger$	–	$1,007^\dagger$
Europarl-ST	77	64	75
CoVoST 2	362	–	344
ASR datasets			
CommonVoice v11	1,503	1,361	$1,082^\dagger$
Total	–	1,842	2,929

Table 1: Filtered training data (in hours) for Siamese and ST training stages. Synthetic data is denoted with †.

3.3 Data Filtering

Siamese pretraining We remove speaker names, as well as events like "Laughter" and "Applause", we convert numbers to their spelled-out forms,[1] convert all text to lowercase, and finally remove all characters that are not included in the vocabulary of the ASR foundation model. Furthermore, we apply a step of ASR-based filtering, to filter out noisy examples stemming from wrong audio-text alignments, where we remove examples with high word-error-rate (WER). We adjust the threshold for each dataset dynamically, ensuring that the resulting data has a WER of 0.11. Thus, the thresholds are 0.5 for MuST-C, 0.28 for Europarl-ST, and 0.4 for CommonVoice, which indicates that Europarl-ST has a significant number of misalignments, a conclusion supported by manual inspection. Removing them allowed for faster convergence during Siamese pretraining.

ST fine-tuning We apply text normalization to the original ST data, remove speaker names and event-related tags from the MuST-C dataset, discard examples with extreme source-to-target text length ratios (Gaido et al., 2022), and finally remove audio-transcription misaligned examples with ASR-based filtering, using a fixed WER threshold of 0.5. For the synthetic CommonVoice data, we remove the ones already present in CoVoST. We also filter the synthetic examples of SegAugment, as the SHAS segmentation frequently resembles the original segmentation, thus resulting in highly similar examples. We retain only the ones that are sufficiently dissimilar from

[1] https://github.com/savoirfairelinux/num2words

the original ones, based on text similarity measures, using TF-IDF features from the translations. More concretely, for each talk id, we compute the similarity matrix of its original translations and the new candidates from SegAugment, find the most similar original example for each new candidate, and add it to the filtered data only if its similarity score is below 0.8. We apply this approach also between the different SegAugment versions (*m, l, xl*).

4 Experiments

Here we describe the experiments we carried out in this work. The implementation details are available in §A.1.

IWSLT '22 System For the IWSLT 2022 offline task, our submission employed a HuBERT encoder (Hsu et al., 2021a) and an mBART50 (En-Xx) decoder, which were efficiently fine-tuned to ST with the LNA strategy (Li et al., 2021) and parallel adapters (He et al., 2022), using datasets such as MuST-C v2, Europarl-ST and CoVoST. The architecture included three 1D convolutional layers between the encoder and decoder, resulting in a subsampling of the encoder representation by a factor of 8. The final ensemble also comprised models utilizing Knowledge Distillation and a wav2vec 2.0 encoder (Tsiamas et al., 2022b).

Baseline Our baseline has four main differences compared our last year's best system. We did an initial exploratory analysis of various encoders (§A.3), including different versions of wav2vec 2.0, and HuBERT. Upon observing no significant differences, we opted to utilize wav2vec 2.0 fine-tuned with pseudo-labels (Xu et al., 2021b), a more prevalent choice within the research community. Despite the strong performance demonstrated by efficient fine-tuning with LNA and parallel adapters, we chose to switch to standard ST fine-tuning in order to optimize performance. Moreover, we employ a semantic encoder initialized from the MT model. Lastly, we also pre-train the foundation models, wav2vec 2.0 with CTC on the ASR data of MuST-C, and mBART50 on the parallel text of MuST-C. It is important to note that only MuST-C data was utilized for the baseline.

Siamese Pre-training Instead of pre-training the speech encoder with CTC only, we follow the Siamese pre-training method (§2.2), with the encoder architecture described in §2.1, to align the encoder representations with the MT model's representation space. The system, instead of using three layers of 1D convolutions, now incorporates also CTC-based compression, a large adapter, and finally a single layer of 1D convolutions. Following the Siamese pre-training on MuST-C's ASR data, we jointly fine-tune the model and the MT decoder on the MuST-C ST data. Similar to the baseline, the MT model is also fine-tuned on the parallel text of MuST-C beforehand.

More Data We extend the previously described process by incorporating additional data. Initially, we fine-tune mBART50 using all the MT data (Table 6). Subsequently, we perform the Siamese pre-training and ST fine-tuning employing all the available speech data (Table 1). By incorporating a larger dataset, we aim to enhance the system's generalization capabilities and overall performance.

Data Augmentation We employ two data augmentation techniques to increase the performance of our system during ST fine-tuning (§3.2), while no modifications are made to the Siamese pre-training. First, we investigate the use of SegAugment (Tsiamas et al., 2022a), which we apply to MuST-C v3. Secondly, we generate synthetic data from Common Voice (Ardila et al., 2020), by leveraging the fine-tuned mBART50 (§A.2).

KD We use knowledge distillation with the fine-tuned mBART50 as the teacher (§A.2). The loss for training the ST model is the average of the standard cross entropy and the Kullback-Leibler (KL) divergence between the MT and ST output probability distributions. We utilize all available ST data in this experiment, including both real and synthetic data.

5 Audio Segmentation

To segment the audio of the IWSLT test sets, we use SHAS (Tsiamas et al., 2022c). The tst2023 test set, unlike previous years, contains another two domains apart from TED talks, which are ACL presentations and Press conferences. We tune the parameters of SHAS separately for each domain, but since no development set is available for the press conferences, we decided to treat it as the ACL domain. For fine-tuning the segmentation parameters, we used the ST model that was trained with synthetic data from CommonVoice and SegAugment and initialized from Siamese pre-training (Table 2, 2d). We evaluate the performance of the

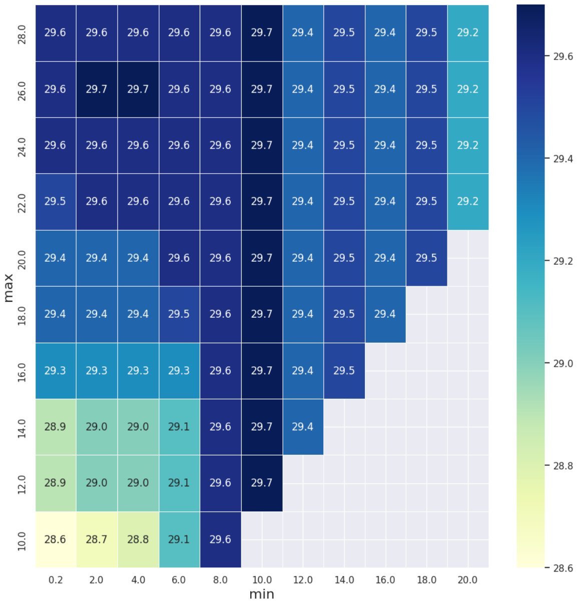

Figure 3: BLEU scores on IWSLT.tst2020 for different combinations of min and max segment length parameters of SHAS.

ST model on many different combinations of the min and max segment length parameters, between 0.2-30 seconds on IWSLT.tst2019 and 0.2-18 on ACLdev2023. In Figure 3, we observe that the minimum segment length of 10 seconds is consistently reaching the best BLEU of 29.7 points. We decided to choose the combination of 10-26 seconds, since the max of 26, seemed to be slightly better compared to other neighboring values. As depicted in Figure 4, smaller segments are better for the ACL domain, with the best BLEU score obtained for min of 0.2 and max of 12. We hypothesize that the differences in the optimal segmentation between the IWSLT and ACL sets is because the ACL data are essentially out-of-domain for our ST models. In turn, the ST models are not confident in their predictions to handle long segments, and thus it is better to translate short segments instead.

6 Results

In Table 2 we provide the BLEU scores on MuST-C tst-COMMON and the IWLST test sets of tst2019 and tst2020 (TED domain), and acl2023 (ACL domain). We are using the original segmentation for MuST-C and apply SHAS with the optimal parameters (§5) of 10-26 secs for the TED domain, and 0.2-12 secs for the ACL one. We also provide the results from our submission to IWSLT '22.

In the first part of Table 2, we observe that this year's baseline (1a) improves results from last year best single model in both MuST-C and IWSLT test sets, although it only uses data from MuST-C. The reasons behind these improvements are the proper fine-tuning of learning rate and regularization parameters, as well as the choice of the speech encoder (§A.3). For the next exepriment (1b), by using the Siamese pretraining (§2.2), instead of just using CTC for the pretraining, we obtain substantial improvements in MuST-C v2, tst2020, and acl2023, indicating the efficacy of our pretraining method when applied on top of foundation models.

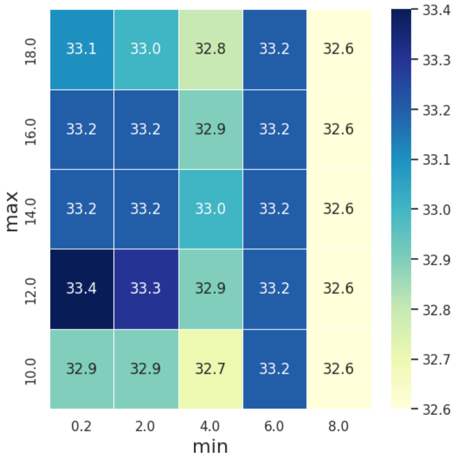

Figure 4: BLEU scores on IWSLT.ACLdev2023 for different combinations of min and max segment length parameters of SHAS.

Adding more data in all parts of the training (2a), including the MT fine-tuning, Siamese pre-training and ST fine-tuning, did not bring any meaningful improvements to MuST-C and IWSLT.tst2019/20, but it dramatically improved the results on the acl2023 development set. We hypothesize that the CommonVoice and CoVoST data play an important role due to the large representation of foreign accents, similar to those in acl2023. Following, with the inclusion of SegAugment in the ST fine-tuning (2b) we observe an increase in all test sets, with larger ones in the IWSLT test sets, since SegAugment data have the same segmentation. Then, also using synthetic data from CommonVoice (2c) has minor improvements in MuST-C and a slight decrease in IWSLT. Despite that, we included synthetic data in subsequent experiments, since they were running in parallel. Applying Knowledge Distillation with the fine-tuned mBART50 as a teacher (2d), brings moderate gains of 0.1-0.4 BLEU in the IWSLT sets, and finally an increase in the learning rate (2e) from 5e-5 to 7.5e-5 provide a model that scored the best in tst2020 and acl2023.

		Dataset	MuST-C		IWSLT		
		split	v2	v3	tst2019	tst2020	acl2023
UPC '22 (Tsiamas et al., 2022b)							
0	a	Best Single	29.4	-	24.9	26.8	-
	b	Best Ensemble	30.8	-	25.4	27.8	-
Only MuST-C							
1	a	Baseline	29.8	29.9	25.7	27.3	25.1
	b	1a + Siamese Pretraining	30.8	30.1	25.9	28.5	26.4
Extended Data Conditions							
2	a	1b + More Data	30.8	30.7	26.0	28.0	31.6
	b	2a + SegAugment	31.3	30.9	26.6	29.4	32.4
	c	2b + synthCV	**31.4**	**31.0**	26.5	29.4	32.3
	d	2c + Knowledge Distillation	30.9	30.7	**26.8**	29.5	32.7
	e	2c + higher LR	31.2	30.8	26.4	**29.8**	**33.4**
Ensembles							
3	a	Ensemble (2d, 2e)	31.4	31.1	26.9	29.7	32.8
	b	Ensemble (2c, 2d, 2e)	31.4	31.1	**27.0**	**29.9**	32.7
	c	Ensemble (2b, 2c, 2d, 2e)	**31.5**	**31.2**	**27.0**	29.8	33.1

Table 2: BLEU scores for En-De MuST-C and IWSLT sets. In **bold** are the best scores by single models, and in **underlined bold** are the best scores overall.

Ensembling multiple models provided small increases in all sets. We believe that there is very little variation in our best models (2b-2e), since they are initialized from the same Siamese pre-training (2b), thus resulting in ineffective ensembles. In general, and in terms of single models, we improve our results from last year by 1.6 BLEU in tst2019 and 2.1 BLEU in tst2020, while the difference is larger in terms of single models.

7 Conclusions

We described the submission of the UPC Machine Translation group for the IWSLT 2023 Offline ST task. Our system leverages ASR and MT foundation models and a Siamese pretraining step to maximize the transfer learning from MT. We show that Siamese pretraining can bring significant improvements to our ST models, while fine-tuning with KD can also be helpful. We furthermore show that synthetic data are crucial at improving performance in the IWSLT test sets. In future work, we plan to investigate the zero-shot capabilities of optimal transport in the context of foundation models.

8 Submission Results

In Tables 3, 4 and 5, we present the official submission results for IWSLT 2023 with our best system, which is the Ensemble 3c of Table 2. Systems are evaluated on the three test sets (TED, ACL, Sub) with three metrics; BLEU (Papineni et al., 2002), chrF (Popović, 2017), and COMET (Rei et al., 2020). The TED test set also has two available references.

Metric	**BLEU**			**chrF**		**COMET**	
Reference	1	2	both	1	2	1	2
System 3c	25.5	29.8	36.6	0.56	0.58	0.7985	0.8098

Table 3: Official Results for the TED test set 2023.

Metric	**BLEU**	**chrF**	**COMET**
System 3c	32.1	0.6	0.7473

Table 4: Official Results for the ACL test set 2023.

Metric	**BLEU**	**chrF**	**COMET**
System 3c	15.6	0.47	0.3746

Table 5: Official Results for the Sub test set 2023.

Acknowledgements

The work done by Ioannis Tsiamas and Gerard I. Gállego was supported by the ADAVOICE project, PID2019-107579RB-I00 / AEI / 10.13039/501100011033

References

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondrej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, Alexander H. Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation, IWSLT 2020, Online, July 9 - 10, 2020*, pages 1–34. Association for Computational Linguistics.

R. Ardila, M. Branson, K. Davis, M. Henretty, M. Kohler, J. Meyer, R. Morais, L. Saunders, F. M. Tyers, and G. Weber. 2020. Common voice: A massively-multilingual speech corpus. In *Proceedings of the 12th Conference on Language Resources and Evaluation (LREC 2020)*, pages 4211–4215.

Alexei Baevski, Yuhao Zhou, Abdelrahman Mohamed, and Michael Auli. 2020. wav2vec 2.0: A framework for self-supervised learning of speech representations. In *Advances in Neural Information Processing Systems*, volume 33, pages 12449–12460. Curran Associates, Inc.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2019. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 58–68, Minneapolis, Minnesota. Association for Computational Linguistics.

Luisa Bentivogli, Mauro Cettolo, Marco Gaido, Alina Karakanta, Alberto Martinelli, Matteo Negri, and Marco Turchi. 2021. Cascade versus direct speech translation: Do the differences still make a difference? In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2873–2887, Online. Association for Computational Linguistics.

Alexandre Berard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. 2018. End-to-End Automatic Speech Translation of Audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6224–6228, Calgary, AB. IEEE.

Roldano Cattoni, Mattia Antonino Di Gangi, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2021. Must-c: A multilingual corpus for end-to-end speech translation. *Computer Speech & Language*, 66:101155.

Alexis Conneau, Alexei Baevski, Ronan Collobert, Abdelrahman Mohamed, and Michael Auli. 2021. Unsupervised Cross-Lingual Representation Learning for Speech Recognition. In *Proc. Interspeech 2021*, pages 2426–2430.

Mattia A. Di Gangi, Matteo Negri, Viet Nhat Nguyen, Amirhossein Tebbifakhr, and Marco Turchi. 2019. Data Augmentation for End-to-End Speech Translation: FBK@IWSLT '19. In *Proceedings of the 16th International Workshop on Spoken Language Translation*, Hong Kong. Publisher: Zenodo.

Charlie Frogner, Chiyuan Zhang, Hossein Mobahi, Mauricio Araya-Polo, and Tomaso Poggio. 2015. Learning with a wasserstein loss. In *Proceedings of the 28th International Conference on Neural Information Processing Systems - Volume 2*, NIPS'15, page 2053–2061, Cambridge, MA, USA. MIT Press.

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

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2020. End-to-end speech-translation with knowledge distillation: FBK@IWSLT2020. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 80–88, Online. Association for Computational Linguistics.

Marco Gaido, Sara Papi, Dennis Fucci, Giuseppe Fiameni, Matteo Negri, and Marco Turchi. 2022. Efficient yet competitive speech translation: FBK@IWSLT2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 177–189, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Gerard I. Gállego, Ioannis Tsiamas, Carlos Escolano, José A. R. Fonollosa, and Marta R. Costa-jussà. 2021. End-to-end speech translation with pre-trained models and adapters: UPC at IWSLT 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 110–119, Bangkok, Thailand (online). Association for Computational Linguistics.

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2019. Adapting Transformer to End-to-End Spoken Language Translation. In *Proc. Interspeech 2019*, pages 1133–1137.

Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. 2006. Connectionist temporal classification: Labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd International Conference on Machine Learning*, ICML '06, page 369–376, New York, NY, USA. Association for Computing Machinery.

Chi Han, Mingxuan Wang, Heng Ji, and Lei Li. 2021. Learning shared semantic space for speech-to-text translation. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2214–2225, Online. Association for Computational Linguistics.

Junxian He, Chunting Zhou, Xuezhe Ma, Taylor Berg-Kirkpatrick, and Graham Neubig. 2022. Towards a unified view of parameter-efficient transfer learning. In *International Conference on Learning Representations*.

Dan Hendrycks and Kevin Gimpel. 2020. Gaussian error linear units (gelus).

Geoffrey E. Hinton, Oriol Vinyals, and Jeffrey Dean. 2015. Distilling the knowledge in a neural network. *ArXiv*, abs/1503.02531.

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

Wei-Ning Hsu, Benjamin Bolte, Yao-Hung Hubert Tsai, Kushal Lakhotia, Ruslan Salakhutdinov, and Abdelrahman Mohamed. 2021a. Hubert: Self-supervised speech representation learning by masked prediction of hidden units. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 29:3451–3460.

Wei-Ning Hsu, Anuroop Sriram, Alexei Baevski, Tatiana Likhomanenko, Qiantong Xu, Vineel Pratap, Jacob Kahn, Ann Lee, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2021b. Robust wav2vec 2.0: Analyzing Domain Shift in Self-Supervised Pre-Training. In *Proc. Interspeech 2021*, pages 721–725.

Hirofumi Inaguma, Brian Yan, Siddharth Dalmia, Pengcheng Guo, Jiatong Shi, Kevin Duh, and Shinji Watanabe. 2021. ESPnet-ST IWSLT 2021 offline speech translation system. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 100–109, Bangkok, Thailand (online). Association for Computational Linguistics.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2020. Europarl-st: A multilingual corpus for speech translation of parliamentary debates.

J. Kahn, M. Rivière, W. Zheng, E. Kharitonov, Q. Xu, P. E. Mazaré, J. Karadayi, V. Liptchinsky, R. Collobert, C. Fuegen, T. Likhomanenko, G. Synnaeve, A. Joulin, A. Mohamed, and E. Dupoux. 2020. Libri-light: A benchmark for asr with limited or no supervision. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7669–7673. `https://github.com/facebookresearch/libri-light`.

Eugene Kharitonov, Morgane Rivière, Gabriel Synnaeve, Lior Wolf, Pierre-Emmanuel Mazaré, Matthijs Douze, and Emmanuel Dupoux. 2021. Data augmenting contrastive learning of speech representations in the time domain. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 215–222.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization.

Paul Knopp and Richard Sinkhorn. 1967. Concerning nonnegative matrices and doubly stochastic matrices. *Pacific Journal of Mathematics*, 21(2):343 – 348.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

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

Phuong-Hang Le, Hongyu Gong, Changhan Wang, Juan Pino, Benjamin Lecouteux, and Didier Schwab. 2023. Pre-training for speech translation: Ctc meets optimal transport.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2021. Multilingual speech translation from efficient finetuning of pretrained models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 827–838.

Yuchen Liu, Hao Xiong, Jiajun Zhang, Zhongjun He, Hua Wu, Haifeng Wang, and Chengqing Zong. 2019. End-to-End Speech Translation with Knowledge Distillation. In *Proc. Interspeech 2019*, pages 1128–1132.

J. Niehues, R. Cattoni, S. Stüker, M. Negri, M. Turchi, Elizabeth Salesky, Ramon Sanabria, Loïc Barrault, Lucia Specia, and Marcello Federico. 2019. The iwslt 2019 evaluation campaign. In *Proceedings of the 16th International Workshop on Spoken Language Translation*.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. 2015. Librispeech: An asr corpus based on public domain audio books. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5206–5210.

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

Gabriel Peyré and Marco Cuturi. 2019. Computational optimal transport: With applications to data science.

Ngoc-Quan Pham, Tuan Nam Nguyen, Thai-Binh Nguyen, Danni Liu, Carlos Mullov, Jan Niehues, and Alexander Waibel. 2022. Effective combination of pretrained models - KIT@IWSLT2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 190–197, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Juan Pino, Liezl Puzon, Jiatao Gu, Xutai Ma, Arya D. McCarthy, and Deepak Gopinath. 2019. Harnessing Indirect Training Data for End-to-End Automatic Speech Translation: Tricks of the Trade. In *Proceedings of the 16th International Workshop on Spoken Language Translation*, Hong Kong. Publisher: Zenodo.

Maja Popović. 2017. chrF++: words helping character n-grams. In *Proceedings of the Second Conference on Machine Translation*, pages 612–618, Copenhagen, Denmark. Association for Computational Linguistics.

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

Tomasz Potapczyk and Pawel Przybysz. 2020. SR-POL's System for the IWSLT 2020 End-to-End Speech Translation Task. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 89–94, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Matthias Sperber and Matthias Paulik. 2020. Speech translation and the end-to-end promise: Taking stock of where we are. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7409–7421, Online. Association for Computational Linguistics.

Xu Tan, Yi Ren, Di He, Tao Qin, and Tie-Yan Liu. 2019. Multilingual neural machine translation with knowledge distillation. In *International Conference on Learning Representations*.

Yuqing Tang, Chau Tran, Xian Li, Peng-Jen Chen, Naman Goyal, Vishrav Chaudhary, Jiatao Gu, and Angela Fan. 2020. Multilingual translation with extensible multilingual pretraining and finetuning. *arXiv preprint arXiv:2008.00401*.

Ioannis Tsiamas, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022a. SegAugment: Maximizing the Utility of Speech Translation Data with Segmentation-based Augmentations.

Ioannis Tsiamas, Gerard I. Gállego, Carlos Escolano, José Fonollosa, and Marta R. Costa-jussà. 2022b. Pretrained speech encoders and efficient fine-tuning methods for speech translation: UPC at IWSLT 2022. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 265–276, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Ioannis Tsiamas, Gerard I. Gállego, José A. R. Fonollosa, and Marta R. Costa-jussà. 2022c. Shas: Approaching optimal segmentation for end-to-end speech translation.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020a. Fairseq S2T: Fast speech-to-text modeling with fairseq. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing: System Demonstrations*, pages 33–39, Suzhou, China. Association for Computational Linguistics.

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

Ruibin Xiong, Yunchang Yang, Di He, Kai Zheng, Shuxin Zheng, Chen Xing, Huishuai Zhang, Yanyan Lan, Liwei Wang, and Tie-Yan Liu. 2020. On layer normalization in the transformer architecture. In *Proceedings of the 37th International Conference on Machine Learning*, ICML'20. JMLR.org.

Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2021a. Stacked acoustic-and-textual encoding: Integrating the pre-trained models into speech translation encoders. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 2619–2630, Online. Association for Computational Linguistics.

Qiantong Xu, Alexei Baevski, Tatiana Likhomanenko, Paden Tomasello, Alexis Conneau, Ronan Collobert, Gabriel Synnaeve, and Michael Auli. 2021b. Self-training and pre-training are complementary for speech recognition. In *ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 3030–3034.

Biao Zhang, Ivan Titov, Barry Haddow, and Rico Sennrich. 2020. Adaptive feature selection for end-to-end speech translation. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 2533–2544, Online. Association for Computational Linguistics.

Ziqiang Zhang and Junyi Ao. 2022. The YiTrans speech translation system for IWSLT 2022 offline shared task. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, pages 158–168, Dublin, Ireland (in-person and online). Association for Computational Linguistics.

Ziqiang Zhang, Sanyuan Chen, Long Zhou, Yu Wu, Shuo Ren, Shujie Liu, Zhuoyuan Yao, Xun Gong, Lirong Dai, Jinyu Li, et al. 2022. Speechlm: Enhanced speech pre-training with unpaired textual data. *arXiv preprint arXiv:2209.15329*.

Jinming Zhao, Hao Yang, Gholamreza Haffari, and Ehsan Shareghi. 2022. M-Adapter: Modality Adaptation for End-to-End Speech-to-Text Translation. In *Proc. Interspeech 2022*, pages 111–115.

A Appendix

A.1 Implementation Details

This section presents the implementation details of our proposed model architecture.

As an ASR model, we are using wav2vec 2.0[2] which is composed of a 7-layer convolutional feature extractor and 24-layer Transformer encoder. It is pretrained with 60k hours of non-transcribed speech from Libri-Light (Kahn et al., 2020), and fine-tuned for ASR with 960 hours of labeled data from Librispeech (Panayotov et al., 2015). The wav2vec 2.0 version we use was also fine-tuned with pseudo-labels (Xu et al., 2021b).

As an MT model, we are using mBART50 (Tang et al., 2020), which is already fine-tuned on En-Xx multilingual machine translation[3]. We further pretrain it for two reasons. Firstly, we are only interested in the En-De direction, and thus we would like a more specialized model on that direction. Secondly, due to the 2nd step of encoder matching, we would like the text encoder to have a very good representation of our data. For MT fine-tuning, we use the original parameters of mBART50 (Tang et al., 2020), and the datasets listed in Table 6.

The acoustic encoder has 24 Transformer layers, while the semantic encoder and the decoder have 12 layers each. All layers have an embedding dimensionality of 1024, a feed-forward dimensionality of 4098, GELU activations (Hendrycks and Gimpel, 2020), 16 attention heads, and pre-layer normalization (Xiong et al., 2020). The vocabulary for the CTC has a size of 32 characters, while the one for the ST model has a size of 250,000.

The model takes waveforms with a 16kHz sampling rate as input, which are normalized to zero mean and unit variance. The models are trained using the data presented in Table 1, with maximum source length of 400,000 and target length of 1024 tokens. Gradient accumulation and data parallelism are employed to achieve an effective batch size of approximately 32 million tokens.

For the Siamese pre-training we use Adam (Kingma and Ba, 2014) with a base learning rate of $2 \cdot 10^{-4}$, a warm-up of 1,000 steps and an inverse square root scheduler. We follow a reduced regularization approach, as compared to the original configuration of wav2vec 2.0 and mBART50, which we found to work the best in our preliminary experiments. Thus, we use 0.1 activation dropout in the acoustic encoder, as well as time masking with probability of 0.2 and channel masking with probability of 0.1. For the context encoder, we use 0.1 dropout and 0.1 attention dropout. All other dropouts are inactive. All the weights in the loss function were set to 1.0 (Eq. 1). We train until the $\mathcal{L}^{OT_2}$ term of the loss does not improve for 5,000 steps, and then average the 10 best checkpoints according to the same loss term.

For ST fine-tuning, we use Adam with a base learning rate of $5 \cdot 10^{-5}$, fixed for the 20% of the training before decaying to $5 \cdot 10^{-7}$ for the rest. In the semantic encoder, we apply a dropout of 0.1 and an attention dropout of 0.1, while for the decoder we use a dropout of 0.3 and an attention dropout of 0.1. Neither dropout nor masking is applied in the frozen acoustic encoder. The loss is the cross-entropy with label smoothing of 0.2.

For the experiments incorporating Knowledge Distillation (KD) during ST fine-tuning, the loss is calculated as a weighted sum of the standard cross-entropy (no label smoothing) and the KL divergence between the teacher and student distributions, controlled by a hyperparameter λ, set to 0.5. The teacher distribution for each step is obtained offline using the fine-tuned mBART50, where we keep the top-8 indices, and both the teacher and student distributions are additionally modified with

[2] `https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec2_vox_960h_new.pt`

[3] `https://dl.fbaipublicfiles.com/fairseq/models/mbart50/mbart50.ft.1n.tar.gz`

temperature $T = 1.3$ (Gaido et al., 2020).

After ST fine-tuning, we pick the 10 best checkpoints according to the BLEU (Papineni et al., 2002) computed with sacreBLEU (Post, 2018) on the development set of MuST-C and average them. For generation, we use a beam search of 5. All models are implemented in FAIRSEQ (Ott et al., 2019), and experiments were run on a cluster of 8 NVIDIA GeForce RTX 3090. Our code is available at a public repository[4].

A.2 MT fine-tuning

For the MT fine-tuning, we use the parallel text of the ST datasets, as well as Europarl v10 En-De (Koehn, 2005) (Table 6). We perform text normalization and remove pairs with extremely short text segments (fewer than 4 characters) or extreme source-to-target length ratio (less than 0.5 or larger than 2).

	Original	Filtered
ST datasets		
MuST-C v3	270	235
Europarl-ST	33	26
CoVoST 2	231	203
MT datasets		
Europarl v10	$1,829$	$1,566$
Total	$2,363$	$2,030$

Table 6: Filtered training data (thousands of sentences) for MT fine-tuning stage.

	MuST-C		**Europarl-ST**	**CoVoST2**
	v2	**v3**		
Off-the-shelf				
mBART50	31.4	30.9	35.0	33.6
Fine-tuned				
MuST-C v2	35.3	34.4	34.6	35.3
All (§3.1)	34.9	34.2	40.3	39.9

Table 7: BLEU scores on MT test sets.

A.3 Preliminary experiments

Before starting the primary experiments for the IWSLT evaluation campaign, we conducted an array of preliminary tests, building on top of previous years' submissions (Gállego et al., 2021; Tsiamas et al., 2022b). These explorations were intended to examine the impact of system configuration variations on the performance metrics on the MuST-C v2 dev set, such as BLEU (Papineni et al., 2002), chrF2 (Popović, 2017), and COMET (Rei et al., 2020). To ensure the robustness of our findings, we estimated statistical significance using the bootstrap resampling method (Koehn, 2004).

In our initial experiment, we examined the impact of various fine-tuning strategies used in our last years' participations, specifically *LNA* (Li et al., 2021) and *LNA-Adapters* (Tsiamas et al., 2022b), in comparison to full fine-tuning. The goal was to verify whether these approaches inadvertently hurt the system's performance. As demonstrated in Table 8, these strategies indeed had a detrimental effect, leading to reductions of 1.9 BLEU points when applied to both the encoder and the decoder. Consequently, we opted to adopt a conventional full fine-tuning strategy for subsequent experiments.

Following this, we conducted a comparative analysis of various speech encoders, including different variations of *wav2vec 2.0* (Baevski et al., 2020; Xu et al., 2021b; Hsu et al., 2021b; Conneau et al., 2021), *HuBERT* (Hsu et al., 2021a), and *SpeechLM* (Zhang et al., 2022) (Table 9). Our baseline was the wav2vec 2.0 fine-tuned with pseudo-labels (Xu et al., 2021b), and intriguingly, most encoders exhibited a comparable level of performance. A marginal decrease was observed with the wav2vec 2.0 pretrained on a large pool of datasets (LV-60 + CV + SWBD + FSH) (Hsu et al., 2021b), and the multilingual version of wav2vec 2.0, XLSR (Conneau et al., 2021). The SpeechLM results were noticeably below expectations, leading us to suspect a bug in our implementation.

Upon noting that the hyperparameters were optimized for a specific speech encoder, we hypothesized that a reduction in the learning rate might boost HuBERT's performance. However, as demonstrated in Table 11, the performance was adversely affected, prompting us to retain the original wav2vec 2.0 as the primary speech encoder due to the lack of substantial improvements offered by other alternatives.

Our focus then shifted towards examining the influence of varying regularization and data augmentation strategies on system performance (Table 10). We explored a range, from our traditionally used setup (*base*), to the one employed in the *original* foundation model fine-tuning, and a *reduced* version. Implementing the *original* regularization within the speech encoder, as opposed to the *base* variant, significantly boosted performance, leading

[4] https://github.com/mt-upc/iwslt-2023

Encoder	Decoder	BLEU	chrF2	COMET
-	-	29.0	54.7	0.8001
LNA	-	28.0 *	54.1 *	0.7949 *
-	LNA	27.9 *	54.0 *	0.7882 *
LNA	LNA	27.1 *	53.2 *	0.7800 *
LNA-Adapt	-	28.2 *	54.3 *	0.7960 *
-	LNA-Adapt	27.6 *	53.6 *	0.7889 *
LNA-Adapt	LNA-Adapt	27.1 *	53.5 *	0.7847 *

Table 8: Performance comparison of fine-tuning strategies w.r.t. to full fine-tuning, evaluated on the MuST-C v2 dev set (en-de). *LNA* and *LNA-Adapters* represent the strategies proposed by (Li et al., 2021) and (Tsiamas et al., 2022b) respectively. $*$ indicates significance w.r.t. baseline (full fine-tuning).

us to select this configuration. We also explored the effectiveness of WavAugment (Kharitonov et al., 2021), ultimately finding that, despite its training speed slowdown, it did not enhance the results. Consequently, we opted to stop using it.

Lastly, we evaluated the potential benefits of employing the new MuST-C v3 training data on system performance (Table 12). Unexpectedly, no significant improvements were observed upon transitioning from MuST-C v2 to v3. Despite this, we decided to utilize v3, since it's specifically prepared for the IWSLT evaluation campaign.

These preliminary investigations have not only provided a more profound understanding of the role of each system's component and setting, but also have yielded us with a better starting point for the subsequent experiments of our work.

Learning Rate	**BLEU**	**chrF2**	**COMET**
$5 \cdot 10^{-4}$	30.3	56.1	0.8099
$2 \cdot 10^{-4}$	30.3	56.0	0.8069
$1 \cdot 10^{-4}$	30.2	55.9	0.8085
$5 \cdot 10^{-5}$	29.5*	55.3*	0.8047

Table 11: Learning rate search for HuBERT encoder, with MuST-C v2 dev set (en-de). $*$ indicates significance w.r.t. baseline (1st row).

Training Data	**BLEU**	**chrF2**	**COMET**
MuST-C v2	30.7	56.4	0.8127
MuST-C v3	30.5	56.6	0.8118

Table 12: Performance of the systems trained with different versions of MuST-C, evaluated with MuST-C v2 dev set (en-de). No significant improvements found.

System	**ASR FT**	**BLEU**	**chrF2**	**COMET**
Wav2Vec 2.0 Large (LV-60) + Self Training	✓	30.2	56.1	0.8087
Wav2Vec 2.0 Large (LV-60)	✓	30.1	55.9	0.8098
Wav2Vec 2.0 Large (LV-60)	✗	30.3	55.9	–
Wav2Vec 2.0 Large (LV-60 + CV + SWBD + FSH)	✓	29.7*	55.7*	0.8083
Wav2Vec 2.0 Large (LV-60 + CV + SWBD + FSH)	✗	30.0	55.9	–
Wav2Vec 2.0 Large conformer - rope (LV-60) †	✓	29.8	55.4*	–
XLSR-53	✗	28.9*	55.0*	–
HuBERT Large	✓	30.3	56.1	0.8099
HuBERT Large	✗	30.3	56.2	0.8110
SpeechLM-P Large ‡	✗	23.6*	50.2*	–

Table 9: Speech encoders exploration with MuST-C v2 dev set (en-de). $*$ indicates significance w.r.t. baseline (1st row). † uses *LNA-Adapters* (Tsiamas et al., 2022b). ‡ indicates a possible bug in our implementation.

Encoder Reg.	**Decoder Reg.**	**WavAugm.**	**BLEU**	**chrF2**	**COMET**
base	base	✓	30.2	56.1	0.8087
base	original	✓	30.5	56.4*	0.8149*
base	original	✗	30.7	56.4*	0.8127*
base	reduced	✓	30.1	55.9	0.8078
original	base	✓	29.8	55.8	0.8100
reduced	base	✓	30.1	55.9	0.8108
original	original	✓	30.4	56.2	0.8138*
reduced	reduced	✓	30.1	56.0	0.8122*

Table 10: Variations of the regularization and data augmentation strategies, with MuST-C v2 dev set (en-de). $*$ indicates significance w.r.t. baseline (1st row).