Efficient yet Competitive Speech Translation: FBK@IWSLT2022

Marco Gaido[1,2] 🐢, Sara Papi[1,2] 🐢, Dennis Fucci[1,2], Giuseppe Fiameni[3], Matteo Negri[1], Marco Turchi[1]
[1]Fondazione Bruno Kessler
[2]University of Trento
[3]NVIDIA AI Technology Center
{mgaido, spapi, dfucci, negri, turchi}@fbk.eu

Abstract

The primary goal of this FBK's systems submission to the IWSLT 2022 offline and simultaneous speech translation tasks is to reduce model training costs without sacrificing translation quality. As such, we first question the need of ASR pre-training, showing that it is not essential to achieve competitive results. Second, we focus on data filtering, showing that a simple method that looks at the ratio between source and target characters yields a quality improvement of 1 BLEU. Third, we compare different methods to reduce the detrimental effect of the audio segmentation mismatch between training data manually segmented at sentence level and inference data that is automatically segmented. Towards the same goal of training cost reduction, we participate in the simultaneous task with the same model trained for offline ST. The effectiveness of our lightweight training strategy is shown by the high score obtained on the MuST-C en-de corpus (26.7 BLEU) and is confirmed in high-resource data conditions by a 1.6 BLEU improvement on the IWSLT2020 test set over last year's winning system.

1 Introduction

The yearly IWSLT offline speech translation (ST) evaluation campaign aims at comparing the models produced by companies, universities, and research institutions on the task of automatically translating speech in one language into text in another language. Given a blind test set, participants' submissions are ranked according to the obtained SacreBLEU score (Post, 2018).

Over the years, the competition to achieve the highest score has driven to bigger and bigger models trained on large datasets: the 2021 winning model (Bahar et al., 2021b) has twice the number of encoder layers (12 vs 6), and a deeper (6 vs 4 layers) and larger (1024 vs 512 features) decoder compared to the 2019 winner (Potapczyk et al., 2019). In addition, most of the competitors have relied on knowledge transfer techniques (Ansari et al., 2020; Anastasopoulos et al., 2021b), such as the initialization of the ST model encoder with the encoder of an ASR system trained on large corpora (Bansal et al., 2019). All these practices have contributed to a remarkable increase in computational expenses and energy consumption that are antithetic with the recent rise of concerns on the social and environmental consequences of these costs (Strubell et al., 2019).

Among the harms inherent to the high computational cost of training ST systems, there is also the risk of restricting the participation in competitions like IWSLT to few big players from the industry sectors that can afford them. As part of a research institution, with this work we try to answer the question: *can we reduce the training cost of ST systems without sacrificing final translation quality?* Specifically, can we train a competitive direct ST model from scratch, without expensive pre-training (e.g. ASR pre-training or self-supervised learning on huge dataset – Baevski et al. 2020)?

To answer these questions, we perform a preliminary study on the English-German (en-de) section of MuST-C (Cattoni et al., 2021), one of the most widespread ST corpora and then we scale to the high-resource data condition allowed by the task organizers. On MuST-C, we show that with the aid of a Connectionist Temporal Classification (CTC) auxiliary loss (Graves et al., 2006) and compression (Gaido et al., 2021a) in the encoder, our Conformer-based (Gulati et al., 2020) model can outperform – to the best of our knowledge – the previous best reported value of 25.3 BLEU by Inaguma et al. (2021), even avoiding any additional pre-training or transfer learning. Moreover, with the addition of a simple data filtering method, we achieve the new state-of-the-art score of 26.7 BLEU for a direct ST model that does not exploit external (au-

🐢 The authors contributed equally.

Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022), pages 177 - 189
May 26-27, 2022 ©2022 Association for Computational Linguistics

dio or textual) resources. Scaling to high-resource data conditions, we notice that the gap between an ASR pre-trained system and a system trained from scratch is closed only after a fine-tuning on in-domain data. Our submission to the offline task consists of an ensemble of three models that scores 32.2 BLEU on MuST-C v2 and 27.6 on IWSLT tst2020.

In the same vein of reducing the overall training computational costs, we participated also in the simultaneous task using our best offline model and without performing any additional training do adapt it to the simultaneous scenario (Papi et al., 2022). The simultaneous version of our offline-trained model is realized by applying the wait-k strategy (Ma et al., 2019) with adaptive word detection from the audio input (Ren et al., 2020) that determines the number of words in a speech segment using the greedy prediction of the CTC. Our SimulST model achieves competitive results on the MuST-C v2 test set compared to the last year systems, scoring 25 BLEU at medium latency ($< 2s$) and 30 BLEU at high latency ($< 4s$) while keeping low ($300 - 400ms$) the computation overhead and requiring no dedicated training.

2 Competitive ST without Pre-training

Before training systems on huge corpora, we conduct preliminary experiments on the MuST-C benchmark to find a promising setting aimed at reducing the high computational costs of ST. First, we validate on different architectures the finding of previous works (Gaido et al., 2021a; Papi et al., 2021b) that ST models trained with an additional CTC loss do not need an initialization of the encoder with that of an ASR model. To this aim, we add a CTC loss (Gaido et al., 2021a) whose targets are the lowercase transcripts without punctuation.[1] Second, we explore data selection mechanisms to increase model quality and reduce training time. We always use the same hyper-parameters used in our final trainings for all systems (see Section 6) unless otherwise specified.

2.1 Model Selection

As a first step, we compare different architectures proposed for ST: ST-adapted Transformer (Wang et al., 2020b), Conformer (Gulati et al., 2020), and Speechformer (Papi et al., 2021b). In addition, we also test a composite architecture made of a first stack of 8 Speechformer layers and a second stack of 4 Conformer layers. Hereinafter, we refer to this architecture as Speechformer Hybrid. As a side note, we also experimented with replacing the ReLU activation functions in the decoder of our Conformer model with the squared ReLU, in light of the recent findings on language models (So et al., 2021) showing accelerated model convergence, decreased training time, and improved performance. Unfortunately, these benefits were not observed in our experiments, as the introduction of the squared ReLU caused a small performance drop (-0.2 BLEU) and did not improve the convergence speed of the model. So, we do not consider this change in the rest of the paper.

In all the architectures, the encoder starts with two 1D convolutions. These layers compress the input sequence by a factor of 4 except for the Speechformer, where they do not perform any downsampling. Indeed, the Speechformer relies on a modified self-attention mechanism (ConvAttention) with reduced memory requirements and shrinks the length of the input sequence only on top of 8 ConvAttention layers by means of the CTC-compression (Gaido et al., 2021a) mechanism before feeding the sequence to 4 Transformer layers. However, in a randomly initialized state, the CTC compression may actually not reduce the input sequence (or only slightly), leading to OOM errors caused by the quadratic memory complexity with respect to the sequence length of the Transformer layers. For this reason Papi et al. (2021b) initialize their encoder layers up to the CTC-compression module with a pre-trained model. Since we aim at reducing the computational cost avoiding any pre-training, we introduce two methods that ensure a minimal compression factor of the input sequence after the CTC-compression:

- **Max Output Length**: if the sequence produced by the CTC compression is longer than a threshold (a hyper-parameter that we set to 1/4 of the maximum input sequence length[2]), we merge (averaging them) an equal number of consecutive vectors so that the final length of the sequence is inferior of the defined threshold. For instance, if the maximum

[1] We add the CTC loss in the 8th encoder layer since (Gaido et al., 2021a; Papi et al., 2021a) has demonstrated that it compares favourably with adding the CTC on top of the encoder outputs or in other layers (Bahar et al., 2019).

[2] This ensures that the resulting sequences are not longer than the maximum length obtained by the Transformer and Conformer architectures after the two 1D convolutions.

input sequence length is 4,000, we set the threshold to 1,000; in this case, if a sample results in a sequence of length 2,346 after the CTC compression, we merge the first 3 vectors, then the vectors from the 4th to the 6th, and so on. We use 3 because it is the minimum compression factor that satisfies the length requirement.[3]

- **Fixed compression**: for a given number of epochs n_E (a hyperparameter) the CTC compression is disabled and replaced by a fixed compression that averages 4 consecutive vectors. In this way, we directly control the length of the sequence after the compression, resembling the fixed compression performed by the initial 1D convolutional layers of Transformer and Conformer ST models.

We choose the n_E parameter of the fixed compression method among the values 6, 8, 10, and 12 according to the BLEU score[4] on the dev set. The best score was achieved with $n_E = 10$ (24.16 BLEU), which was lower than the score obtained by the Max Output Length method (24.26 BLEU). As such, in Table 1 (*w/o pretrain* column) we report the results of Speechformer and Speechformer Hybrid with the Max Output Length method.

The results show that the Speechformer-based models do need pre-training to reach their best scores while Conformer and Transformer models achieve comparable translation quality avoiding the pre-training. Specifically, the Conformer architecture with CTC compression obtains the best score without pre-training (25.5 BLEU) and has a negligible gap from the best result with pre-training (25.7 of Speechformer Hybrid). We can hence confirm the statement that ASR pre-training can be avoided at barely no translation quality cost, and hereinafter we use the Conformer with CTC compression without pre-training unless noted otherwise. It is worth mentioning that the introduction of the CTC compression in the Conformer encoder does not only increase translation quality; also, it reduces the RAM requirements and speeds up both the inference and training phases. Indeed, as the sequence length is significantly reduced in the last encoder layers and in the encoder-decoder attention, less computations are required and the mini-batch size – the number of samples processed in parallel – can be increased. Overall, this leads to save $\sim 35\%$ of the training and inference time.

Model	w pretrain	w/o pretrain
Transformer	23.6	23.6
Speechformer	24.5	24.3
Conformer	24.8	24.8
+ CTC compr.	25.6	**25.5**
Speechformer Hybrid	**25.7**	24.9

Table 1: SacreBLEU on the *tst-COMMON* set of MuST-C v1 en-de.

2.2 Data Filtering

Easy methods to improve the quality of ST systems – and deep neural networks in general – consist in providing them with *more* data or *better* data. The first approach comes at the cost of longer training time and higher computational requirements. This makes the second approach more appealing and in line with the overall goal and spirit of this work. We hence focus on the definition of an efficient filtering strategy that improves the quality of our training data (and consequently of our models) without additional computational costs.

We start from the observation that ST models estimate the probability of an output text given an input audio $p(Y|X)$, and a good ST model assigns a low probability to erroneous samples, which are outliers of the $p(Y|X)$ distribution. Although training a ST model only to filter the training data would be extremely computationally expensive, we decided to adopt this method as an upper bound for comparison with easier and feasible strategies. In particular, for each sample in the training set, we computed the negative log-likelihood[5] (NLL) with a strong ST model trained on all the data available for the competition (see Section 5) as a proxy of the probability of the sample. A high NLL means that a sample is unlikely, while a NLL close to 0 means that the sample has a very high probability. Based on this, we can filter all the samples above a threshold to remove the least probable ones. To set the threshold, we draw an histogram on all the training sets (see Figure 2 in the Appendix) that leads to the following considerations: *i)* each dataset has a different distribution, making it difficult to define a threshold valid for all of them, and *ii)* MuST-C has the highest NLL, meaning that it is more complex to fit for the model.

[3] A compression factor 2 would result in a sequence of length 1,173 – higher than the 1,000 threshold – while 3 produces a sequence of length 782.

[4] BLEU+case.mixed+smooth.exp+tok.13a+version.1.5.1

[5] The negative log-likelihood is defined as $-log(p(Y|X))$.

Through the approach described above, we selected the data of MuST-C - the dataset we used in these preliminary experiments - with a NLL greater than 4.0. Upon a manual inspection of a sample of these selected data (5-10% of the total), we noticed that two main categories were present: *i)* bad source/target text alignments[6] (e.g. two sentences in the target translation are paired with only one in the transcript or vice versa), and *ii)* free (non-literal) translations. Instead, no cases of bad audio-transcript alignments were found (this was only a non-exhaustive manual inspection though), meaning that this problem is likely less widespread and impactful than the textual alignment errors in the corpus.

These considerations motivated us to search for a feasible strategy that filter out the bad source/target text alignments. We first considered a simple method that discards samples with too high or low ratio between the target translation length (in characters) and the duration of the source audio.[7] The corresponding histogram on the training data can be found in Figure 3 in the Appendix. Looking at the plots, it emerges that this ratio is strongly dataset-dependent, likely due to the high variability in speaking rate for different domains and conditions, thus making it hard to set good thresholds. For this reason, also supported by the finding of the manual inspection on the good quality of audio-text alignments discussed above, we turn to examine the ratio between the target translation length and the *source transcript length*.[8] Figure 4 in the Appendix shows its histogram: in this case, the behavior is consistent on all datasets, making it easy to determine good values for the minimum and maximum ratio to admit (we set them to 0.8 and 1.6).

In Table 2 we report the results of our filtering method and we compare it with the upper bound of the NLL-based filtering strategy as well as with previous works both under the same data condition and with additional external data. First, we can notice that our simple method based on the target/source character ratio leads to a 1.2 BLEU gain, and has a very small gap (0.2 BLEU) with respect to the upper bound exploiting a strong ST model for filtering. Second, our score (26.7 BLEU) is significantly higher than those reported by previous direct ST works in the same data condition and is on par or even outperforms those of models trained with the addition of external resources. Finally, we compare the results of our model with those of the best cascade models reported in the same data conditions (Bahar et al., 2021a): the tightly-integrated cascade is close to our model (-0.2 BLEU), but ours also benefits from the data filtering technique we just discussed.

Model	BLEU
Cascade (Bahar et al., 2021a)	25.9
Tight Integrated Cascade (Bahar et al., 2021a)	26.5
Without external data	
SATE (Xu et al., 2021)	25.2
BiKD (Inaguma et al., 2021)	25.3
With external data	
JT-ST (Tang et al., 2021)	26.8
Chimera (Han et al., 2021)	26.3
This work	
Conformer + CTC compr.	25.5
+ char-ratio filter.	26.7
+ NLL-based filter.	26.9

Table 2: SacreBLEU on the *tst-COMMON* set of MuST-C v1 en-de. Chimera uses additional speech and WMT14 (Bojar et al., 2014), while JT-ST uses only WMT14 as external resource.

To sum up, we managed to define a training recipe that enables reaching state-of-the-art ST results on MuST-C en-de (26.7 BLEU) with a single training step and involves: *i)* the Conformer architecture, *ii)* an auxiliary CTC loss and CTC-compression in the 8th encoder layer, and *iii)* a simple yet effective filtering strategy based on the ratio between source and target number of characters. In the following section, we discuss the application of this procedure in high-resource data conditions.

3 Audio Segmentation Strategy

ST models are usually trained and evaluated in the ideal and unrealistic condition of audio utterances split at sentence level. As such, when fed with an unsegmented audio stream, they suffer from the mismatch between the training and inference data, which often results in significant performance drops. Accordingly, our last year submission (Papi et al., 2021a) focused on reducing the impact of this distributional shift, both by increasing the robustness of the model with a fine-tuning on a random re-segmentation of the MuST-C training set (Gaido et al., 2020a), and by means of a hybrid method for

[6] In the MuST-C corpus, the alignments between transcripts and translations of the training set are automatically produced, hence misalignments and textual differences can be present.

[7] In practice, we compute the number of characters divided by the number of $10ms$ audio frames.

[8] We used normalized transcript without punctuation, so the length of the target translation is on average 1.2X that of the source transcript.

audio segmentation (Gaido et al., 2021c), which considers both the audio content and the desired length of the resulting speech segments. The experiments showed that the two approaches accounted for complementary gains, both contributing to obtain our best scores.

Recently, Tsiamas et al. (2022) presented a novel Supervised Hybrid Audio Segmentation (SHAS) with excellent results in limiting the translation quality drop. SHAS adopts a probabilistic version of the Divide-and-Conquer algorithm by Potapczyk and Przybysz (2020) that progressively splits the audio at the frame with highest probability of being a splitting point until all segments are below a specified length. The probability of being a splitting point is estimated by a classifier fed with audio representations generated by wav2vec 2.0 (Baevski et al., 2020) and trained to approximate the manual segmentation of the existing corpora, i.e. to emit 1 for frames representing splitting points and 0 otherwise. Since this approach involves a prediction with neural models of considerable size, its superiority over the VAD-based ones comes with a significant computational cost and overhead. In addition, SHAS is not applicable to audio streams, as it requires the full audio to be available before start splitting. In the context of this competition, however, these limitations do not represent a significant issue.

Tsiamas et al. (2022) compare SHAS with previous segmentation methods only using models trained on well-formed sentence-utterance pairs. In this work, we validate their findings also on models fine-tuned on randomly segmented data to check: *i)* whether this fine-tuning brings benefits also with audio segmented with SHAS, and *ii)* whether the gap between SHAS and other segmentation is closed or not by the fine-tuning.

4 Simultaneous

In light of the recent work that questions the necessity of a dedicated training procedure for simultaneous model (Papi et al., 2022), we participate in the Simultaneous task with the same model used for the Offline task. Their finding is perfectly aligned with the spirit of this submission toward the reduction of training computational costs. We determine when to start generating the output translation adopting the wait-k strategy (Ma et al., 2019) that simply prescribes to wait for k words before starting to generate the translation, where k is a hyper-parameter controlled by the user that can be increased or decreased to directly control the latency of the system. The number of words in a given input speech is determined with an adaptive word detection strategy (Ren et al., 2020), because of its superiority over the fixed strategy (Ma et al., 2020b) in strong models trained in high-resource data conditions (Papi et al., 2022). Our adaptive word detection mechanism exploits the predicted output of CTC module in the encoder (Ren et al., 2020; Zeng et al., 2021) to count the number of words in the source speech.

The number of words to wait – k – is not the only hyper-parameter that controls the wait-k strategy. Another important factor is *how often* we check the number of uttered words that is the length of the *speech segment*. A short speech segment means that the system decides more frequently whether to wait for more input or to produce a part of output. This can reduce the latency, but it increases the number of forward passes through the encoder and hence the computational cost. In addition, a longer speech segment implies that the system takes decision with more context at disposal, possibly improving the quality. For this reason, we performed preliminary experiments exploring different speech segment dimensions (every $40ms$ ranging from $120ms$ to $720ms$) and we found $320ms$ and $640ms$ to be superior to other values. Accordingly, we report the results of our systems for these two speech segment durations varying the value of k to achieve different latency. In particular, we test our model with $k = \{1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14\}$ in order to lie in the latency intervals prescribed by the Simultaneous Shared Task.[9] The latency intervals are determined by the Average Lagging (Ma et al., 2020b) – or AL – on MuST-C v2 tst-COMMON and are: *Low Latency* with $AL \leq 1000ms$, *Medium Latency* with $AL \leq 2000ms$, and *High Latency* with $AL \leq 4000ms$. We use a standard AL-BLEU graph to report the system performance, where in the x axis we find the AL values ranging from $700ms$ to $4000ms$ and in the y axis the corresponding BLEU values. Moreover, we also report the AL_{CA}, the computational aware version of the AL metric (Ma et al., 2020b) accounting also for the computational time spent by the model during inference, in an AL_{CA}-BLEU graph that will be used to additionally score the

[9] https://iwslt.org/2022/simultaneous

performance in the simultaneous task.

5 Data

As training set, we use the ASR and ST datasets allowed for the offline task,[10] which are the same allowed for the simultaneous one. The ASR data consist in *(speech, transcript)* pairs that, in our case, are in English. The ST data consist in *(speech, transcript, translation)* triplets from a source language (here English) to a target language (here German). The ASR data we used are: LibriSpeech (Panayotov et al., 2015), TEDLIUM version 3 (Hernandez et al., 2018), Voxpopuli (Wang et al., 2021), and Mozilla Common Voice.[11] The ST data we used are: MuST-C version 2 (Cattoni et al., 2021), CoVoST version 2 (Wang et al., 2020a), and Europarl-ST (Iranzo-Sánchez et al., 2020).

The ASR-native corpora were included in our ST training by applying Sequence Knowledge Distillation (Kim and Rush, 2016; Gaido et al., 2021b) – or SeqKD –, a popular data augmentation method used in the past IWSLT editions (Ansari et al., 2020; Anastasopoulos et al., 2021a) in which a teacher MT model is used to translate the source transcripts into the target language. To avoid additional computational costs, we choose as MT teacher the freely available pre-trained model by Tran et al. (2021) for WMT2021 that was trained on the corresponding WMT2021 dataset (Akhbardeh et al., 2021), allowed by the IWSLT2022 Offline Task. The SeqKD method was also applied to MuST-C v2 in order to augment the scarce ST available data. As such, our training set comprised the synthetic data built using SeqKD and the native ST data, both filtered with the method described in Section 2.2. The two types of data were distinguished by means of a tag pre-pended to the target text (Gaido et al., 2020b; Papi et al., 2021a).

6 Experimental Settings

All the models used for our participation were implemented on Fairseq-ST (Wang et al., 2020b).[12] All the architectures (Transformer, Speechformer, Speechformer Hybrid, and Conformer) consist in 12 encoder layers and 6 decoder layers, 512 features for the attention layers and 2,048 hidden units in the feed-forward layers. We used 0.1 dropout for the feed-forward layer and attention layer. For Conformer convolutional layers we also apply 0.1 dropout and we set the kernel size to 31 for the point- and depth-wise convolutions. We trained with the Adam optimizer (Kingma and Ba, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.98$). The learning rate was set to increase linearly from 0 to $2e-3$ for the first 25,000 warm-up steps and then to decay with an inverse square root policy. Differently, it was kept constant for model fine-tuning, with a value of $1e{-}3$. The vocabularies are built via SentencePiece models (Sennrich et al., 2016). In our preliminary experiments only on MuST-C, the number of merge operations was set to 8,000 (Di Gangi et al., 2020) for the German translations and 5,000 (Wang et al., 2020b) for the lowercase punctuation-free English transcripts. In the experiments on high-resource data condition, we doubled these values. We normalize the audio features before passing them to our models with Cepstral Mean and Variance Normalization. Specifically, in offline ST the mean and variance are estimated at utterance level, while for simultaneous ST inference the normalization is based on the global mean and variance estimated on the MuST-C version 2 training set.

Trainings were performed on 4 NVIDIA A100 GPUs with 40GB RAM. We set the maximum number of tokens to 40k per mini-batch and 2 as update frequency for the Conformer with CTC-compression. The other models were trained with 20k tokens per mini-batch and 4 as update frequency. We trained each model for 100,000 updates, corresponding to about 28 hours for the Conformer with CTC-compression. For offline generation, the maximum number of tokens was decreased to 25k, since we used a single K80 GPU with 12GB RAM and we applied the beam search strategy with `num_beams=5`. For simultaneous generation based on SimulEval (Ma et al., 2020a), we used a K80 GPU and greedy search.

7 Results

In this section, we report our experiments in high-resource data conditions and we discuss our submission to the Offline (section 7.1 and Simultaneous (section 7.2) tasks.

7.1 Offline

Fine-tuning on in-domain data. In addition to training our models in the high-resource data con-

[10] `https://iwslt.org/2022/offline`

[11] `https://commonvoice.mozilla.org/en/datasets`

[12] Code available at: `https://github.com/hlt-mt/FBK-fairseq`.

	Model	BLEU
I.	Conformer	30.6
II.	+ in-domain fn	31.6
III.	Conformer_pretrain	31.5
IV.	+ in-domain fn	**31.7**
V.	Ensemble (II, III)	32.0
VI.	Ensemble (III, IV)	31.7
VII.	Ensemble (II, IV)	**32.2**

Table 3: BLEU on MuST-C v2 tst-COMMON for Conformer with pretraining (*Conformer_pretrain*) and without it (*Conformer*). We also report the scores after fine-tuning on in-domain data (+ *in-domain fn*).

dition, we also investigate whether fine-tuning on in-domain data brings advantages or not. The results are reported in Table 3. As we can notice, the Conformer with pre-training outperforms its version trained from scratch by 0.9 BLEU. However, when both the systems are fine-tuned on the in-domain data (rows II and IV), this difference becomes negligible (0.1 BLEU) meaning that the pre-training phase can be skipped in favor of a single fine-tuning step. This might also suggest that the learning rate scheduler and the hyper-parameters we used – tuned on MuST-C corpus – may be sub-optimal when a large amount of data is available. For time reasons, we did not investigate this aspect, which we leave to future work. In addition, we compared several model ensembles: the Conformer with fine-tuning (II) and the pre-trained Conformer (III); the pre-trained Conformer (III) and the pre-trained Conformer with fine-tuning (IV); the Conformer with fine-tuning (II) and the pre-trained Conformer with fine-tuning (IV). Our results show that ensembling the pre-trained Conformer and its fine-tuned version (VI) does not bring benefits, while selecting the Conformer without pre-training fine-tuned on in-domain data and the Conformer with pre-training (V) leads to some improvements, which are enhanced when the two fine-tuned models are used (VII). We also tested ensembles with more than 2 models without obtaining any advantage in terms of translation quality.

Fine-tuning on re-segmented data. As introduced in Section 3, we tested two audio segmentation methods: the *Hybrid* segmentation (Gaido et al., 2021c), and the *SHAS* segmentation (Tsiamas et al., 2022). Also, we fine-tuned our ST models on automatically re-segmented data to reduce the mismatch between train and evaluation conditions. The results are shown in Table 4. First, we notice that the SHAS segmentation method improves over the Hybrid one, with gains from 0.7 to 3.4 BLEU. Secondly, we see that the fine-tuning on resegmented data – useful with the Hybrid segmentation – becomes useless if using SHAS. In fact, the best overall results are obtained using SHAS on a model that is not fine-tuned on resegmented data (row 2), which scores 30.4 BLEU on the MuST-C v2 tst-COMMON and 26.8 BLEU on the IWSLT 2020 test set. As such, we can conclude that fine-tuning on resegmented data is not needed if the audio is segmented with SHAS.

Ensembles. Since in the experiments on in-domain fine-tuning the best overall score was obtained by an ensemble of models, we compared the best combination (Ensemble VII in Table 3) with other ensembles obtained by combining models fine-tuned on re-segmented data and models without this fine-tuning. As we can see from rows 7-10 of Table 4, the best scores are realized by adding a model fine-tuned on re-segmented data (6) to Ensemble VII, although the gap between all the ensembles is small on both test sets (≤ 0.4 BLEU). This 3-models ensemble (10) obtained the best overall BLEU of 31.3 on MuST-C v2 tst-COMMON and 27.6 on IWSLT 2020 test set, outperforming by 1.6 BLEU the best result reported last year (Inaguma et al., 2021).

Offline Submissions. Given the results of the Ensemble (1, 2, 6), we chose its output as our *primary* submission for the Offline Shared task. On the basis of the small performance drop on both test sets (0.4 BLEU) and to verify the possibility of avoiding the fine-tuning on re-segmented data, we choose the Ensemble (1, 2) as *contrastive* submission. Lastly, we can notice that the single Conformer model without pre-training (1) falls behind the best Ensemble by only 1 BLEU for MuST-C v2 tst-COMMON and 1.2 BLEU for IWSLT 2020 test set. This suggests that users can be served with sound and competitive translations even with a single model obtained with less than 30 hours of total training time on 4 GPUs. To test this hypothesis, we sent the translations generated by the latter system as additional *contrastive* submission. We report in Table 5 the official results for the tst2022 and tst2021 sets. The scores confirm our findings that the gap between the best ensemble and the single model without pre-training is limited to less than 1 BLEU. Most significantly, this single model outperforms the best direct system reported last

	Model	Hybrid		SHAS	
		tst-COMMON	iwslt2020	tst-COMMON	iwslt2020
1.	Conformer + in-domain fn	27.4	23.8	30.3	26.4
2.	Conformer_pretrain + in-domain fn	28.1	24.4	**30.4**	**26.8**
	with fine-tuning on resegmented data				
3.	Conformer + resegm. fn	28.3	25.2	29.3	26.1
4.	Conformer + in-domain fn + resegm. fn	29.1	25.0	29.9	26.2
5.	Conformer_pretrain + resegm. fn	29.0	25.9	29.8	26.7
6.	Conformer_pretrain + in-domain fn + resegm. fn	29.0	25.7	29.7	**26.8**
	Ensembles				
7.	Ensemble (1, 2)	28.6	24.7	30.9	27.2
8.	Ensemble (4, 6)	29.7	26.0	30.5	27.2
9.	Ensemble (2, 6)	28.9	25.7	30.8	27.4
10.	Ensemble (1, 2, 6)	28.9	25.8	**31.3**	**27.6**

Table 4: BLEU scores of Hybrid and SHAS audio segmentation methods of the models with and without fine-tuning on re-segmented data (*resegm. fn*) on the MuST-C v2 tst-COMMON and the IWSLT2020 test set.

Model		tst2022			tst2021		
		ref2	ref1	both	ref2	ref1	both
Best direct IWSLT 2021	(Bahar et al., 2021b)	-	-	-	22.6	18.3	31.0
Best cascade IWSLT 2021	HW-TSC (Anastasopoulos et al., 2021b)	-	-	-	24.6	20.3	34.0
This work							
primary	Ensemble (1, 2, 6)	**23.6**	**21.0**	**32.9**	**25.5**	**21.3**	**35.6**
contrastive1	Ensemble (1, 2)	23.4	20.6	32.5	25.4	20.9	35.2
contrastive2	Conformer + in-domain fn	22.8	20.1	31.6	24.5	20.2	33.9

Table 5: BLEU scores on the official blind tst2022 and tst2021 sets of our primary and contrastive submissions.

year (Bahar et al., 2021b) by 1.9 BLEU on the two single references and 2.9 BLEU on both references. Our primary submission increases these gains to 2.9-3.0 BLEU on the single references and 4.6 BLEU on both references, and beats the best cascade system from last year campaign (HW-TSC – Anastasopoulos et al. 2021b) by 0.9-1.0 BLEU on the single references and 1.6 BLEU on both references. All in all, we can conclude that this work has shown that a lightweight training procedure is possible without dramatically sacrificing the quality and competitiveness of the system. We believe that our results are promising for future works in this direction.

7.2 Simultaneous

For the SimulST task participation we use the best performing offline model, namely the Conformer with pre-training and fine-tuning on in-domain data, to which we apply the wait-k policy with adaptive word detection. The AL- and AL$_{\text{CA}}$-BLEU graphs are shown in Figure 1.

As we can see from the AL-BLEU graph, the systems with speech segment $320ms$ and $640ms$ have similar behaviour in terms of quality. The main difference between them is the minimum latency from which they start: the system with speech segment $320ms$ starts at an AL of about $800ms$ while the system with speech segment $640ms$ starts at about $900ms$. On average, if the k value increases, the AL increases by $300ms$ for both systems, with a wider latency interval at the beginning that progressively shrinks at high latency values. In spite of this, the system with speech segment $320ms$ achieves the highest BLEU slightly before the *Medium Latency* (25.1) and *High Latency* thresholds (30.1), making it the best candidate for submission. If we look at the AL$_{\text{CA}}$-BLEU graph, the results partially change because the system with speech segment $640ms$ has a lower computational burden, achieving up to 2 BLEU points improvement at low latency against the other system. Thus, looking at the computational aware metric, the best candidate is the system with speech segment $640ms$. We can conclude that $320ms$ is the best speech segment value for the AL ranking while $640ms$ is the best for the AL computational aware version. Since the organizers encourage multiple submissions, we participate with both the speech segment values.

8 Conclusions

We described the FBK participation in the IWSLT 2022 offline and simultaneous tasks (Anastasopoulos et al., 2022). Our focus was to build a system with the least number of training steps but capable of obtaining competitive results with state-of-

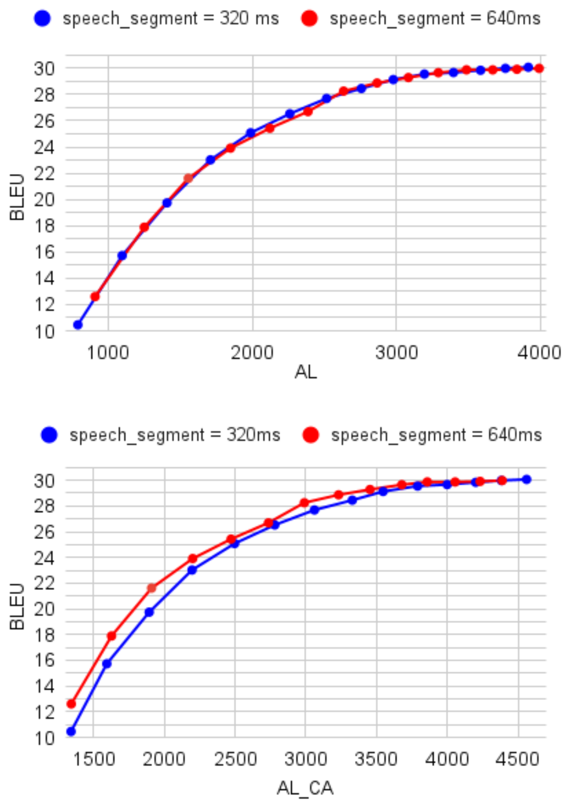

Figure 1: AL- and AL_{CA}-BLEU curves on MuST-C v2 tst-COMMON.

the-art models, which typically undergo complex and longer training procedures. To this aim, we *i)* showed that ASR pre-training of the encoder can be avoided without a significant impact on the final system performance, *ii)* proposed a simple yet effective data filtering technique to enhance translation quality while reducing the training time, and *iii)* compared different solutions to deal with automatic audio segmentation at inference time. Our results on the IWSLT2020 test set indicate that a single Conformer-based model without pre-training falls behind our best model ensemble by only 1.2 BLEU and outperforms the best score reported last year by 0.4 BLEU. The same trend occurs on the blind tst2021 and tst2022 sets, with a 0.8-1.1 BLEU gap from our best model ensemble, which in turn beats by ∼1 BLEU the best reported result last year. These promising results are also confirmed in the simultaneous scenario in which, using the offline-trained model without any adaptation for the simultaneous task, we reach good quality-latency balancing, especially in the more realistic computational aware evaluation setting.

9 Acknowledgements

This work has been supported by the ISCRA-B project *DireSTI* granted by CINECA, and by Amazon Web Services. The submission to the simultaneous track has been carried out as part of the project Smarter Interpreting (`https://kunveno.digital/`) financed by CDTI Neotec funds.

References

Farhad Akhbardeh, Arkady Arkhangorodsky, Magdalena Biesialska, Ondřej Bojar, Rajen Chatterjee, Vishrav Chaudhary, Marta R. Costa-jussa, Cristina España-Bonet, Angela Fan, Christian Federmann, Markus Freitag, Yvette Graham, Roman Grundkiewicz, Barry Haddow, Leonie Harter, Kenneth Heafield, Christopher Homan, Matthias Huck, Kwabena Amponsah-Kaakyire, Jungo Kasai, Daniel Khashabi, Kevin Knight, Tom Kocmi, Philipp Koehn, Nicholas Lourie, Christof Monz, Makoto Morishita, Masaaki Nagata, Ajay Nagesh, Toshiaki Nakazawa, Matteo Negri, Santanu Pal, Allahsera Auguste Tapo, Marco Turchi, Valentin Vydrin, and Marcos Zampieri. 2021. Findings of the 2021 conference on machine translation (WMT21). In *Proceedings of the Sixth Conference on Machine Translation*, pages 1–88, Online. Association for Computational Linguistics.

Antonios Anastasopoulos, Luisa Bentivogli, Marcely Z. Boito, Ondřej Bojar, Roldano Cattoni, Anna Currey, Georgiana Dinu, Kevin Duh, Maha Elbayad, Marcello Federico, Christian Federmann, Hongyu Gong, Roman Grundkiewicz, Barry Haddow, Benjamin Hsu, Dávid Javorský, Věra Kloudová, Surafel M. Lakew, Xutai Ma, Prashant Mathur, Paul McNamee, Kenton Murray, Maria Nădejde, Satoshi Nakamura, Matteo Negri, Jan Niehues, Xing Niu, Juan Pino, Elizabeth Salesky, Jiatong Shi, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Yogesh Virkar, Alex Waibel, Changhan Wang, and Shinji Watanabe. 2022. FINDINGS OF THE IWSLT 2022 EVALUATION CAMPAIGN. In *Proceedings of the 19th International Conference on Spoken Language Translation (IWSLT 2022)*, Dublin, Ireland. Association for Computational Linguistics.

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alexander Waibel, Changhan Wang, and Matthew Wiesner. 2021a. FINDINGS OF THE IWSLT 2021 EVALUATION CAMPAIGN. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 1–29, Bangkok, Thailand (online). Association for Computational Linguistics.

Antonios Anastasopoulos, Ondřej Bojar, Jacob Bremerman, Roldano Cattoni, Maha Elbayad, Marcello Federico, Xutai Ma, Satoshi Nakamura, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Sebastian Stüker, Katsuhito Sudoh, Marco Turchi, Alex Waibel, Changhan Wang, and Matthew Wiesner. 2021b. Findings of the IWSLT 2021 Evaluation Campaign. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, Online.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej 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 Waibel, and Changhan Wang. 2020. FINDINGS OF THE IWSLT 2020 EVALUATION CAMPAIGN. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34, Online. Association for Computational Linguistics.

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 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Parnia Bahar, Tobias Bieschke, and Hermann Ney. 2019. A Comparative Study on End-to-end Speech to Text Translation. In *Proceedings of International Workshop on Automatic Speech Recognition and Understanding (ASRU)*, pages 792–799, Sentosa, Singapore.

Parnia Bahar, Tobias Bieschke, Ralf Schlüter, and Hermann Ney. 2021a. Tight Integrated End-to-End Training for Cascaded Speech Translation. In *2021 IEEE Spoken Language Technology Workshop (SLT)*, pages 950–957.

Parnia Bahar, Patrick Wilken, Mattia A. Di Gangi, and Evgeny Matusov. 2021b. Without further ado: Direct and simultaneous speech translation by AppTek in 2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 52–63, Bangkok, Thailand (online). Association for Computational Linguistics.

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, Volume 1 (Long and Short Papers)*, pages 58–68, Minneapolis, Minnesota. Association for Computational Linguistics.

Ondřej Bojar, Christian Buck, Christian Federmann, Barry Haddow, Philipp Koehn, Johannes Leveling, Christof Monz, Pavel Pecina, Matt Post, Herve Saint-Amand, Radu Soricut, Lucia Specia, and Aleš Tamchyna. 2014. Findings of the 2014 workshop on statistical machine translation. In *Proceedings of the Ninth Workshop on Statistical Machine Translation*, pages 12–58, Baltimore, Maryland, USA. Association for Computational Linguistics.

Roldano Cattoni, Mattia A. 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.

Mattia A. Di Gangi, Marco Gaido, Matteo Negri, and Marco Turchi. 2020. On Target Segmentation for Direct Speech Translation. In *Proceedings of the 14th Conference of the Association for Machine Translation in the Americas (AMTA 2020)*, pages 137–150, Virtual.

Marco Gaido, Mauro Cettolo, Matteo Negri, and Marco Turchi. 2021a. 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, Mauro Cettolo, and Marco Turchi. 2020a. Contextualized Translation of Automatically Segmented Speech. In *Proc. Interspeech 2020*, pages 1471–1475.

Marco Gaido, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2020b. 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, Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2021b. On Knowledge Distillation for Direct Speech Translation . In *Proceedings of CLiC-IT 2020*, Online.

Marco Gaido, Matteo Negri, Mauro Cettolo, and Marco Turchi. 2021c. Beyond voice activity detection: Hybrid audio segmentation for direct speech translation. In *Proceedings of The Fourth International Conference on Natural Language and Speech Processing (ICNLSP 2021)*, pages 55–62, Trento, Italy. Association for Computational Linguistics.

Alex Graves, Santiago Fernández, Faustino J. 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)*, pages 369–376, Pittsburgh, Pennsylvania.

Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, and Ruoming Pang.

2020. Conformer: Convolution-augmented Transformer for Speech Recognition. In *Proc. Interspeech 2020*, pages 5036–5040.

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.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia A. Tomashenko, and Yannick Estève. 2018. TED-LIUM 3: Twice as much data and corpus repartition for experiments on speaker adaptation. In *Speech and Computer - 20th International Conference, SPECOM 2018, Leipzig, Germany, September 18-22, 2018, Proceedings*, volume 11096 of *Lecture Notes in Computer Science*, pages 198–208. Springer.

Hirofumi Inaguma, Tatsuya Kawahara, and Shinji Watanabe. 2021. Source and target bidirectional knowledge distillation for end-to-end speech translation. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 1872–1881, 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. In *ICASSP 2020 - 2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 8229–8233.

Yoon Kim and Alexander M. Rush. 2016. Sequence-level knowledge distillation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1317–1327, Austin, Texas. Association for Computational Linguistics.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Mingbo Ma, Liang Huang, Hao Xiong, Renjie Zheng, Kaibo Liu, Baigong Zheng, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, Hua Wu, and Haifeng Wang. 2019. STACL: Simultaneous translation with implicit anticipation and controllable latency using prefix-to-prefix framework. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3025–3036, Florence, Italy. Association for Computational Linguistics.

Xutai Ma, Mohammad Javad Dousti, Changhan Wang, Jiatao Gu, and Juan Pino. 2020a. SIMULEVAL: An evaluation toolkit for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 144–150, Online. Association for Computational Linguistics.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020b. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. 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*, pages 582–587, Suzhou, China. Association for Computational Linguistics.

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

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2021a. Dealing with training and test segmentation mismatch: FBK@IWSLT2021. In *Proceedings of the 18th International Conference on Spoken Language Translation (IWSLT 2021)*, pages 84–91, Bangkok, Thailand (online). Association for Computational Linguistics.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2021b. Speechformer: Reducing information loss in direct speech translation. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1698–1706, Online and Punta Cana, Dominican Republic.

Sara Papi, Marco Gaido, Matteo Negri, and Marco Turchi. 2022. Does simultaneous speech translation need simultaneous models?

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

Tomasz Potapczyk and Pawel Przybysz. 2020. SRPOL'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.

Tomasz Potapczyk, Pawel Przybysz, Marcin Chochowski, and Artur Szumaczuk. 2019. Samsung's system for the IWSLT 2019 end-to-end speech translation task. In *Proceedings of the 16th International Conference on Spoken Language Translation*, Hong Kong. Association for Computational Linguistics.

Yi Ren, Jinglin Liu, Xu Tan, Chen Zhang, Tao Qin, Zhou Zhao, and Tie-Yan Liu. 2020. SimulSpeech: End-to-end simultaneous speech to text translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3787–3796, Online. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

David R. So, Wojciech Mańke, Hanxiao Liu, Zihang Dai, Noam Shazeer, and Quoc V. Le. 2021. Primer: Searching for efficient transformers for language modeling.

Emma Strubell, Ananya Ganesh, and Andrew McCallum. 2019. Energy and policy considerations for deep learning in NLP. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3645–3650, Florence, Italy. Association for Computational Linguistics.

Yun Tang, Juan Pino, Xian Li, Changhan Wang, and Dmitriy Genzel. 2021. Improving speech translation by understanding and learning from the auxiliary text translation task. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4252–4261, Online. Association for Computational Linguistics.

Chau Tran, Shruti Bhosale, James Cross, Philipp Koehn, Sergey Edunov, and Angela Fan. 2021. Facebook ai's wmt21 news translation task submission. In *Proc. of WMT*.

Ioannis Tsiamas, Gerard I Gállego, José AR Fonollosa, and Marta R Costa-jussà. 2022. Shas: Approaching optimal segmentation for end-to-end speech translation. *arXiv preprint arXiv:2202.04774*.

Changhan Wang, Juan Pino, Anne Wu, and Jiatao Gu. 2020a. CoVoST: A diverse multilingual speech-to-text translation corpus. In *Proceedings of The 12th Language Resources and Evaluation Conference*, pages 4197–4203, Marseille, France. European Language Resources Association.

Changhan Wang, Morgane Riviere, Ann Lee, Anne Wu, Chaitanya Talnikar, Daniel Haziza, Mary Williamson, Juan Pino, and Emmanuel Dupoux. 2021. VoxPopuli: A large-scale multilingual speech corpus for representation learning, semi-supervised learning and interpretation. 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 993–1003, Online. Association for Computational Linguistics.

Changhan Wang, Yun Tang, Xutai Ma, Anne Wu, Dmytro Okhonko, and Juan Pino. 2020b. fairseq s2t: Fast speech-to-text modeling with fairseq. In *Proceedings of the 2020 Conference of the Asian Chapter of the Association for Computational Linguistics (AACL): System Demonstrations*.

Chen Xu, Bojie Hu, Yanyang Li, Yuhao Zhang, Shen Huang, Qi Ju, Tong Xiao, and Jingbo Zhu. 2021. 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.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021. RealTranS: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 2461–2474, Online. Association for Computational Linguistics.

A Dataset Statisctics for Data Filtering

In this Section we report the histograms created when defining our data filtering mechanism (Section 2.2).

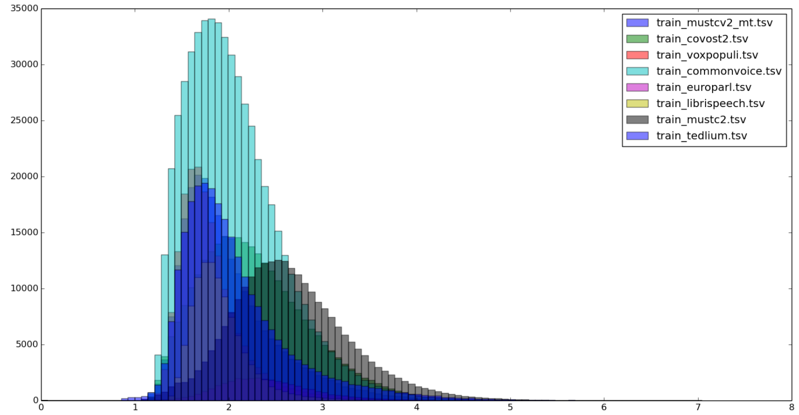

Figure 2: Histogram of the negative log-likelihood (NLL) of the samples for all the training set of the competition. The ST model used to estimate the NLL has been trained on all the data and was scoring 29.6 BLEU on MuST-C.

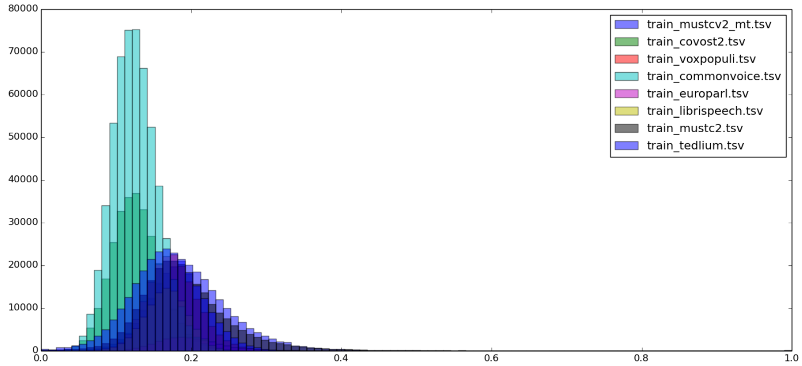

Figure 3: Histogram of the ratio between the number of target translation character and 10ms audio frames for all the training set of the competition.

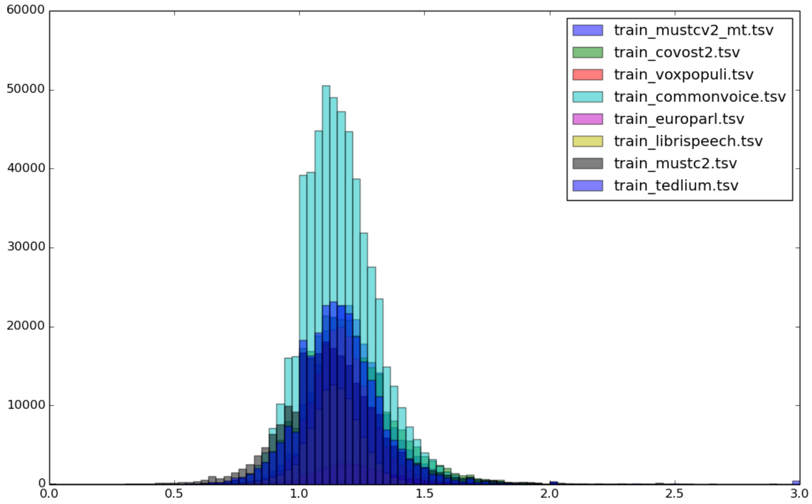

Figure 4: Histogram of the ratio between the number of characters in the target translation and the source punctuation-free transcript for all the training set of the competition.