

Improving Speech Translation Accuracy and Time Efficiency with Fine-tuned wav2vec 2.0-based Speech Segmentation

Ryo Fukuda, *Nonmember*, Katsuhito Sudoh, *Nonmember*, Satoshi Nakamura, *Fellow, IEEE*

Abstract—Speech translation (ST) automatically converts utterances in a source language into text in another language. Splitting continuous speech into shorter segments, known as speech segmentation, plays an important role in ST. Recent segmentation methods trained to mimic the segmentation of ST corpora have surpassed traditional approaches. Tsiamas et al. [1] proposed a segmentation frame classifier (SFC) based on a pre-trained speech encoder called wav2vec 2.0. Their method, named SHAS, retains 95-98% of the BLEU score for ST corpus segmentation. However, the segments generated by SHAS are very different from ST corpus segmentation and tend to be longer with multiple combined utterances. This is due to SHAS’s reliance on length heuristics, i.e., it splits speech into segments of easily translatable length without fully considering the potential for ST improvement by splitting them into even shorter segments. Longer segments often degrade translation quality and ST’s time efficiency. In this study, we extended SHAS to improve ST translation accuracy and efficiency by splitting speech into shorter segments that correspond to sentences. We introduced a simple segmentation algorithm using the moving average of SFC predictions without relying on length heuristics and explored wav2vec 2.0 fine-tuning for improved speech segmentation prediction. Our experimental results reveal that our speech segmentation method significantly improved the quality and the time efficiency of speech translation compared to SHAS.

Index Terms—End-to-end speech-to-text translation, speech segmentation, pretrained speech encoder.

I. INTRODUCTION

THE segmentation of continuous speech is a fundamental process required for speech translation (ST) and other spoken language applications. In text-to-text machine translation (MT), the input text is usually segmented into sentences using punctuation marks as boundaries. However, such explicit boundaries are unavailable in ST. ST corpora usually contain speech segments that are aligned to sentences. For example, the procedure for creating a multilingual ST corpus, MuST-C [2], first performs sentence alignment between English transcriptions and its translations and aligns the English speech and transcriptions with a forced aligner. Much ST research uses such sentence-aligned speech segments to train and evaluate

systems, although they cannot be used in practical situations. In addition, existing ST models cannot directly translate long continuous speech without segmentation. One reason is that the required computational resources increase with the length of the input speech. Even without any constraints on computational resources, an ST model trained on segmented short speech struggles to translate extremely long speech that is not included in its training data. For these reasons, several efforts have focused on speech segmentation for ST.

Pause-based segmentation with voice activity detection (VAD) is commonly used as preprocessing for automatic speech recognition (ASR) and ST. However, pauses in speech do not necessarily coincide with boundaries of semantic units such as sentences in text, e.g., there may be long pauses in an utterance corresponding to a sentence or almost no pauses between utterances. Over-segmentation, in which a silence interval fragments a sentence, and under-segmentation, in which multiple sentences are included in one segment while ignoring a short pause, reduce the ASR and ST performances [3]. Fixed-length segmentation is the simplest approach that segments audio at a predefined segment length [4]. There is also a combination method that concatenates speech segments generated by VAD up to a certain length. Such **length-based segmentation** methods are heuristic approaches that can split speech into segments of easily translatable length [5]. **Punctuation-based segmentation** methods are often used in cascade ST, re-segmenting the ASR results of segments produced by VAD with a punctuation restoration model or a language model [6], [7]. These methods can improve the translation accuracy of MT, but they cannot be used for end-to-end ST, where the source language is translated directly without ASR. In addition, these methods cannot prevent ASR errors due to improper segmentation.

As mentioned above, ST corpora usually have speech segments that correspond to sentences, which are suitable for translation. Recent **corpus-based segmentation** methods have been successful using a classification model trained to predict segmentation of ST corpora. A corpus-based method, SHAS [1], led to state-of-the-art results with a segmentation frame classifier (SFC) based on a pre-trained speech encoder called wav2vec 2.0 [8]. However, the segments generated by SHAS tend to be significantly longer than segments of ST corpus. Such long segments can decrease translation quality. In addition, the longer the segment is, the more computation time required for translation. These long segments are caused by using segmentation algorithms that place more importance

Ryo Fukuda is with the Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma 630-0192, Japan (e-mail: fukuda.ryo.fo3@is.naist.jp).

Katsuhito Sudoh is with the Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma 630-0192, Japan (e-mail: sudoh@is.naist.jp).

Satoshi Nakamura is with the Data Science Center and Graduate School of Science and Technology, Nara Institute of Science and Technology, Ikoma 630-0192, Japan (e-mail: s-nakamura@is.naist.jp).

on the lengths of segments than SFC prediction. This strategy allows SHAS to split speech into segments whose lengths are preferred by ST. However, the following potential remains unconsidered: improving ST translation accuracy and time efficiency by splitting these segments even shorter.

In this work, we extend SHAS to improve ST translation accuracy and efficiency by splitting speech into shorter segments that correspond to sentences, such as those included in ST corpus. We introduce a simple segmentation algorithm using the moving average of SFC predictions without relying on length heuristics to produce shorter segments. We also introduce an efficient fine-tuning of wav2vec 2.0 to improve SFC accuracy. We conducted experiments with an end-to-end ST on MuST-C v2 for English-to-German. Our experimental results showed that the proposed method retained 97.4% of BLEU score for MuST-C segmentation included in the corpus, surpassing 95.1% by SHAS. We also showed that the proposed method reduced the translation time by about 20% while improving translation accuracy by generating shorter segments. Our case analysis revealed that our proposed method sometimes outperformed MuST-C segmentation and produced competitive translation results. Furthermore, an evaluation using 8 language pairs from MuST-C v1 and Europarl-ST showed that the proposed method is effective for target languages and domains that differ from the dataset used to train the SFC.

II. RELATED WORK

Early studies on segmentation for ST considered modeling with the Markov decision process [4], [9], conditional random fields [10], [11], and support vector machines [12]–[15]. They focused on cascade ST systems that consist of an ASR model and a statistical machine translation model, which were superseded by newer ST systems based on neural machine translation.

In recent studies, many speech segmentation methods based on VAD have been proposed for ST. Gaido et al. [16] and Inaguma et al. [17] used the heuristic concatenation of VAD segments up to a fixed length to address the over-segmentation problem. Gállego et al. [18] used a pre-trained ASR model called wav2vec 2.0 [8] for silence detection. Yoshimura et al. [19] used an RNN-based ASR model to consider consecutive blank symbols (“_”) as a segment boundary in decoding using connectionist temporal classification (CTC). Such CTC-based speech segmentation has the following advantage; segment lengths can be intuitively controlled by adjusting the number of consecutive blank symbols that are regarded as segment boundaries. However, these methods often split speech at inappropriate boundaries for ST because they mainly segment speech based on long pauses.

Re-segmentation using ASR transcripts is widely used in cascade STs. Improvements in MT performance have been reported by re-segmenting transcriptions to sentence units using punctuation restoration [11], [15], [20]–[22] and language models [23], [24]. Unfortunately, they are difficult to use in end-to-end ST and cannot prevent the ASR errors caused by speech segmentation.

Corpus-based segmentation using manually or semi-manually segmented speech corpora is a leading recent ap-

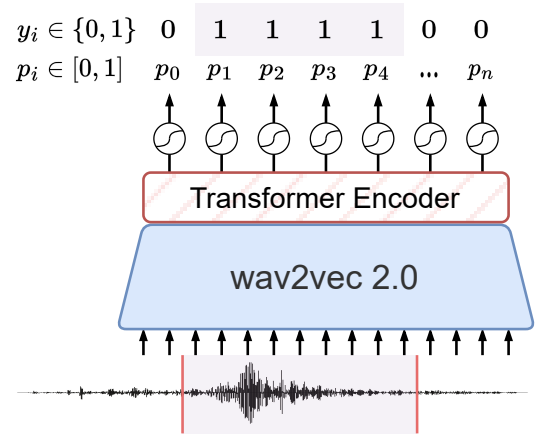


Fig. 1: SFC of SHAS: Value of $y = 1$ indicates that corresponding frame is part of a segment of ST corpus, and $y = 0$ indicates that it is part of a segment boundary.

proach. Wan et al. [3] introduced a re-segmentation model trained with movie and TV subtitle corpora to modify the segment boundaries in ASR output. Wang et al. [25] and Iranzo-Sánchez et al. [26] proposed an RNN-based text segmentation model trained with a bilingual speech corpus. Methods for directly segmenting speech with a segmentation model have also recently been proposed [1], [27]. Fukuda et al. [27] used a Transformer encoder to build a segmentation model and also proposed a hybrid method that combines VAD and the prediction of the segmentation model. Tsiamas et al. [1] built an SFC based on a pre-trained speech encoder called wav2vec 2.0. Their method, Supervised Hybrid Audio Segmentation (SHAS), is the current state-of-the-art method of speech segmentation for ST.

In our approach, we improve the accuracy of the SFC by unfreezing parts or all of the wav2vec 2.0 parameters during training. Performing full fine-tuning can significantly increase the training cost compared to the original SHAS, so we use a method of parameter-efficient transfer learning (PETL). PETL is a research direction aimed at reducing the computational costs of applying large pre-trained models to new tasks [28]–[30].

III. REVIEW OF SHAS

In this section, we describe an SFC (III-A) and a probabilistic divide-and-conquer (pDAC) algorithm (III-B) of the state-of-the-art speech segmentation method called SHAS. Then we describe its drawback: producing lengthy segments (III-C).

A. Segmentation frame classifier

The SFC determines whether each input speech frame belongs to a segment or a segment boundary. It is implemented as a neural network model with a single Transformer encoder layer that is connected to the encoder of the pre-trained, self-supervised speech model, wav2vec 2.0 (Fig. 1). Given an ST corpus, each frame of speech is labeled as 1 or 0, depending on whether it is included in a segment. During training, speech segments of N seconds split at random positions are used with

sequences of labels $y \in \{0, 1\}$ that correspond to model output sequences. $y = 1$ indicates that the corresponding frame is inside a segment, and $y = 0$ indicates that it is outside of the segments, i.e., belonging to a segment boundary.

At the inference time, given an unlabeled audio waveform, it is split into contiguous segments of length N , which are then input to the SFC. These segments are arranged in such a way that there is no temporal overlap between consecutive segments. For each input with length N , SFC predicts probabilities corresponding to audio frames of length $n = N/320$ due to the convolutional feature extractor of wav2vec 2.0. The wav2vec 2.0 parameters are kept fixed during the training, and only the parameters of the final Transformer encoder layer and the output layer are updated.

B. Probabilistic divide-and-conquer

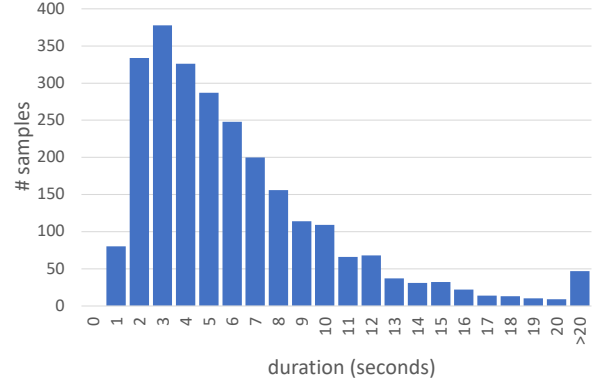
Algorithm 1 pDAC

```

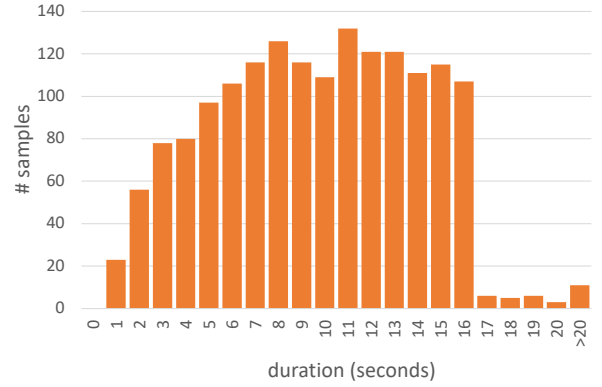
1: Inputs:  $probs, max, min, thr$ 
2: Initialize:
3:    $segments \leftarrow$  empty List
4:    $sgm \leftarrow \text{Tuple}(0, probs.length)$   $\triangleright$  Init single segment
5: RECURSIVE_SPLIT( $sgm$ )
6: return  $segments$ 
7:
8: Procedure RECURSIVE_SPLIT( $sgm$ )
9:   if  $sgm.length < max$  then
10:    append  $sgm$  to  $segments$ 
11:   else
12:      $j \leftarrow 0$ 
13:      $indices \leftarrow \text{argsort } probs[sgm]$ 
14:     while True do
15:        $sgm_a, sgm_b \leftarrow \text{split } sgm \text{ at } indices[j]$ 
16:        $sgm_a \leftarrow \text{trim}(probs[sgm_a], thr)$ 
17:        $sgm_b \leftarrow \text{trim}(probs[sgm_b], thr)$ 
18:       if  $sgm_a.length > min$  and
19:          $sgm_b.length > min$  then
20:         RECURSIVE_SPLIT( $sgm_a$ )
21:         RECURSIVE_SPLIT( $sgm_b$ )
22:       break
23:      $j \leftarrow j + 1$ 

```

During inference, pDAC divides speech based on the probability of each frame being included in a segment ($probs$) predicted by SFC. pDAC is a recursive algorithm that splits speech at the point least likely to be in a segment and applies the same split to the two resulting segments (Algorithm 1). The algorithm utilizes three hyperparameters: a maximum segment length (max) to regulate the length of the resulting segments, a minimum segment length (min) to prevent excessively small noisy segments, and a threshold (thr) to trim a segment's ends, which are classified as being excluded from segments. After a split, the resulting segments are trimmed to the first and last frames i, j with $p_i, p_j > thr$. A split is performed until the segment's length is less than max . This allows pDAC to keep the segments' length within a certain range.



(a) Sentence-aligned speech segmentation



(b) pDAC segmentation

Fig. 2: Histograms of segment length in each segmentation: Horizontal axis indicates length (seconds) of audio segment, and vertical axis indicates number of samples.

C. Lengthy segments by SHAS

SHAS outperformed the existing pause-based and length-based segmentation methods and consistently achieved better translation quality across multiple language pairs. However, the segments generated by SHAS tended to be significantly longer than the MuST-C segments. For example, the average length of the speech segments in MuST-C English-to-German was 5.79 seconds; the average length of segments generated by pDAC was 9.17 seconds. Fig. 2 shows the segment length distribution for each segmentation. The mode value for the sentence-aligned speech segmentation is about three seconds, whereas the mode for the pDAC segmentation is noticeably longer, about 11 seconds.

The cause of such long segments is that pDAC stops the segmentation once the segment length falls below a predefined value max , as mentioned. Thus, although SHAS is a corpus-based segmentation method using SFC, it also has aspects of length-based segmentation. Its advantage is that speech can be split into segments whose lengths are preferred by ST. Tsiamas et al. [1] found that a max of values between 14–18 seconds works well, which are considerably longer than the average MuST-C segments' length of 5.79 seconds.

On the other hand, longer segments by SHAS can degrade the translation quality and the time efficiency of ST. The longer

Algorithm 2 pTHR+MA

```

1: Inputs:  $probs, max, min, thr, n\_ma, lerp_{min}, lerp_{max}$ 
2: Initialize:
3:    $segments \leftarrow$  empty List
4:    $start \leftarrow 0$ 
5:    $thrs \leftarrow$  List with size  $max$ 
6:    $thrs[:min] \leftarrow 0, thrs[min:] \leftarrow thr$ 
7:    $\triangleright$  Set threshold filter  $thrs$ 
8:    $thrs \leftarrow Lerp(thrs, min, lerp_{min}, 0, thr)$ 
9:    $thrs \leftarrow Lerp(thrs, lerp_{max}, max, thr, 1)$ 
10:   $\triangleright$  Apply Linear Interpolation
11:   $probs \leftarrow MovingAverage(probs, n\_ma)$ 
12:   $\triangleright$  Apply Moving Average of  $n\_ma$  frames
13:  while  $start < probs.length$  do
14:    if  $probs[start] \leq thr$  then
15:       $start \leftarrow start + 1$ 
16:    else
17:       $end \leftarrow \min(start + max, probs.length)$ 
18:      for  $i = start \dots end$  do
19:        if  $probs[i] \leq thrs[i]$  then
20:           $end \leftarrow i$ 
21:        break
22:      append Tuple( $start, end$ ) to  $segments$ 
23:  return  $segments$ 

```

a segment is, the more likely that translation omissions will occur by neural machine translation [31]. In addition, the longer a segment is, the more computational time that is required for translation due to the increased time complexity, longer decoder outputs, etc.

To bridge the gap between SHAS and ST corpus segmentation, we need a segmentation algorithm that does not heavily rely on length heuristics. In this case, SFC predictions become more critical to translation accuracy.

IV. PROPOSED METHOD

Next we propose an online decoding algorithm that focuses on SFC prediction rather than segment length to produce shorter segments (IV-A). We also introduce efficient fine-tuning to update the parameters of the upper layers of wav2vec 2.0 to improve SFC accuracy (IV-B) ¹.

A. Probability-first decoding algorithm with moving average

We introduce a segmentation algorithm based on probability thresholds (pTHR) that uses SFC predictions to find sentence boundaries. pTHR progressively determines where segments start and end using the probabilities of each speech frame that is included in a segment corresponding to a sentence ($probs$), predicted by SFC.

The algorithm simply takes the point at which the probability of being included in a segment exceeds the threshold as starting point s_i of segment i and the point at which it again falls below it as end point e_i . $thrs$ is a sequence of probability thresholds that

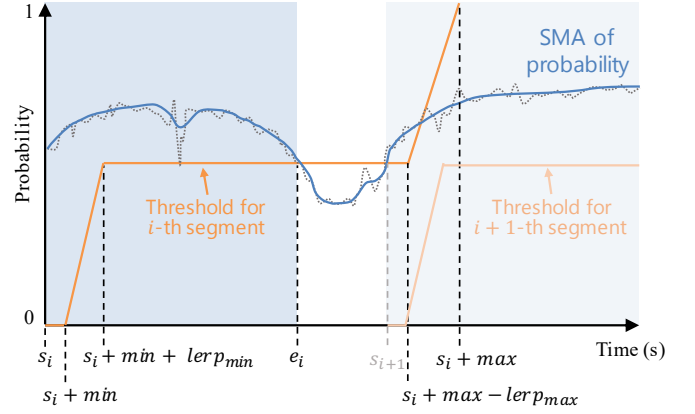


Fig. 3: Schematic diagram of proposed decoding algorithm.

determine the end of a segment, and its length is the number of speech frames corresponding to max seconds. The values contained in $thrs$ are almost thr (e.g., 0.5), although they are set to 0 at positions below min to ensure that the segment length is greater than or equal to min . We also applied linear interpolation between $thrs[min : lerp_{min}]$ and $thrs[max - lerp_{min} :]$ to bias the segment lengths to fall within the normal range. In Algorithm 2, $Lerp(list, start, end, a, b)$ linearly interpolates the values in $list$ between $start$ and end , transitioning smoothly from value a at $start$ to value b at end . The segmentation procedure is as follows:

- 1) The algorithm sequentially looks at the $probs$ values, starting with 0. A point at which the value first exceeds the threshold thr is taken as the starting position of the first segment, s_1 .
- 2) $probs[s_1 : s_1 + max]$ and $thrs$ are compared, and first point j , where $probs[j] < thrs[j]$, is taken as the end position of the first segment e_1 . If no position j is found, $s_1 + max$ is set to e_1 .
- 3) A point where the probability exceeds thr again is taken as the starting position of the second segment s_2 . The positions of the second, third, \dots segments are identically determined as the first segment.

Many existing VAD methods detect speech segments by such thresholding values such as acoustic power or CTC probabilities, etc. Our algorithm, pTHR, differs from them by taking SFC predictions as input. We automate the sentence-level segmentation as given in the ST corpus segmentation instead of performing VAD.

Since pTHR performs thresholding without past information, it can be computed in parallel and at high speed. On the other hand, pTHR is a less stable method than pDAC because its results are highly dependent on SFC accuracy. To stabilize the SFC's prediction, we first tried to use an autoregressive model as SFC, but the model could not be trained due to data imbalance. Then, inspired by classical time-series analysis methods, we incorporated a simple moving average (SMA) [32] into pTHR to smooth the SFC predictions. We applied an SMA with a window size of n_{ma} frames to $probs$, which are the pTHR inputs. Specifically, in line 11 of Algorithm 2, $probs[i]$ is

¹The source code is available at <https://github.com/ahclab/Wav2VecSegmenter>

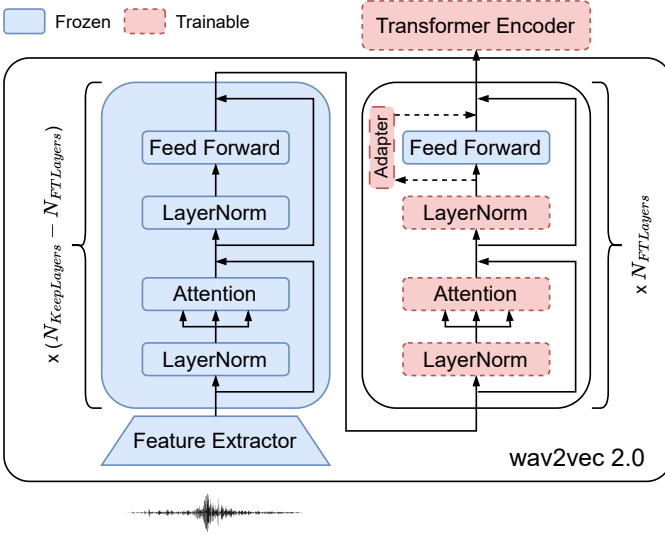


Fig. 4: Segmentation frame classifier with efficient fine-tuning

updated as follows:

$$\begin{aligned} probs'[i] &= \frac{1}{n_{ma}} \sum_{k=\max(i-n_{ma}, 0)}^i probs[k] \\ probs[i] &= probs'[i] \end{aligned}$$

This allows pTHR to stably perform segmentation even when the SFC prediction accuracy is low while maintaining high speed. We discuss the relationship between SFC prediction accuracy and the stability of each algorithm in Section VI-B. Hereafter, we refer to our proposed algorithm with $n_{ma} = 0$ as pTHR and with $n_{ma} > 0$ as pTHR+MA. Pseudocode and a schematic diagram of the pTHR+MA algorithm are shown in Algorithm 2 and Fig. 3.

B. SFC with memory efficient fine-tuning

In our proposed algorithm, the segmentation heavily relies on the SFC prediction, and the translation quality is expected to be affected by SFC accuracy. In SHAS, the wav2vec 2.0 parameters were frozen during the SFC training. In contrast, we introduce *SHAS + FTPT*, which updates the wav2vec 2.0 parameters of SFC (Fig. 4). The parameters of the upper $N_{FTLayers}$ encoder layers are updated out of $N_{AllLayers}$ layers inherited from the wav2vec 2.0. While the self-attention mechanism has quadratic complexity with respect to input length, our model operates with fixed-length inputs during both training and inference, ensuring consistent memory allocation for self-attention. To further optimize memory usage, we froze the parameters of the feed-forward layers, which can be substantial in terms of parameter count. In their place, we introduced parallel adapters [29]. These parallel adapters, as demonstrated by He et al. [29], outperform sequentially inserted adapters and have been validated for their efficacy in ST [33].

V. EXPERIMENTAL SETTINGS

We investigated the effectiveness of our proposed method by conducting speech translation experiments and compared several speech segmentation methods.

TABLE I: Number of segments of MuST-C v2 used in experiments

Language pair	train	dev	tst-COMMON
English-to-German	250,942	1,415	2,580

A. Data

We conducted experiments with the English-to-German (En-De) ST as our primary focus. We used MuST-C v2 for the experiments, which consisted of triplets of segmented English speech, transcripts, and target language translations. Table I shows the statistics of the datasets used in the experiments. MuST-C train and dev split were used to build the SFC models. tst-COMMON was used as a test set for evaluation. The average SNRs of the tst-COMMON are 37dB for the period including only ambient noise only and 1.52dB for the period including applause and laughter.

To further validate the effectiveness of the SFC models across different languages, we performed supplementary tests using the 8 language pairs available in MuST-C v1. Specifically, these tests were conducted from English to German, Spanish (En-Es), French (En-Fr), Italian (En-It), Dutch (En-Nl), Portuguese (En-Pt), Romanian (En-Ro), and Russian (En-Ru). Additionally, to assess the applicability of the method in different domains, we performed tests using the Europarl-ST En-De dataset [34].

B. Evaluation

The evaluation process followed [35]. First, the test set audio files were split using one of the segmentation methods (described in V-C). Then the newly created segments were translated using an ST model (V-D), and the translations were aligned to the references in the test set using mwerSegmenter [35]. Finally, the BLEU scores [36] were calculated with SacreBLEU [37]²³. We also measured BERTScore [38]⁴ and BLEURT [39]⁵.

C. Segmentation method

1) *SFC*: We trained the SFC models with random segments of 20 seconds of audio samples extracted from the training data, following Tsiamas et al. [1]. As a pre-trained speech encoder of wav2vec 2.0, we used an XLS-R model [40] of 300 million parameters⁶, with 24 layers and a dimensionality of 1024. The Transformer encoder has a single layer, 1024 model dimensions, 2048 feed-forward dimensions, eight heads, pre-layer normalization, GELU activation, and 0.1 dropout. Prior to being mapped to probabilities through a linear sigmoid layer, an additional layer of normalization and 0.1 dropout were applied. Models were trained for 16 epochs using Adam with an initial learning rate of $2.5 \cdot 10^{-4}$ (decayed with cosine annealing). After training, the best checkpoint was selected based on the prediction performance of the dev set.

As SFCs of the baseline method SHAS (these models are also called *SHAS*), we built a *middle* model that inherited the

²³<https://github.com/mjpost/sacrebleu>

³signature: BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.5.0

⁴bert-base-multilingual-cased

⁵<https://storage.googleapis.com/bleurt-oss-21/BLEURT-20.zip>

⁶<https://huggingface.co/facebook/wav2vec2-xls-r-300m>

lower 16 layers of the XLS-R encoder and a *large* model that inherited 24 layers. In their preliminary experiments, Tsiamas et al. [1] found that it is beneficial to inherit the lower 14 layers from XLS-R. We also quoted the scores they reported for comparison.

We created the following six variations of *SHAS + FTPT* shown in Section IV-B. The model settings are shown in brackets in the format $N_{FTLayers}/N_{AllLayers}$.

- *middle+quarter* (4/16)
- *middle+half* (8/16)
- *middle+all* (16/16)
- *large+quarter* (6/24)
- *large+half* (12/24)
- *large+all* (24/24)

2) *Segmentation algorithm*: We used pDAC (Section III-B) and pSTRM [5] as baseline segmentation algorithms. pSTRM splits on the longest pause in the interval (*min* and *max*), if any, and otherwise it splits at *max*. pSTRM also emphasizes the target segments' length like pDAC, although it is an online decoding algorithm like our proposed algorithms. The proposed algorithms are pTHR+MA with moving average and pTHR without it. We tuned hyperparameters of each segmentation algorithm using dev set. For pDAC and pSTRM, both of which prioritize segment length, we fixed the threshold at 0.5, *min* = 0.2, and tried *max* = {28, 26, 24, 22, 20, 18, 16, 14, 12, 10}. For pTHR and pTHR+MA, we fixed *max* = 28, *min* = 0.2, and tried *thr* = {0.9, 0.8, 0.7, 0.6, 0.5, 0.4, 0.3, 0.2, 0.1} and a moving average of {0, 0.1, 0.2, 0.4, 0.8, 1} seconds. As per the above settings, our proposed algorithm also imposes length constraints with *max* and *min*. However, these are merely safeguards to avoid extreme lengths, and usually, the segmentation positions are determined by probability and threshold *thr*.

D. Speech translation model

Following Tsiamas et al. [1], we used the joint speech-to-text model [41] from fairseq [42] for MuST-C v2 En-De and Europarl-ST En-De. This joint model is a Transformer encoder-decoder that can take both speech and text as input and share the top layer of the encoder between the two modalities. It performs knowledge distillation from the text-to-text translation task as a guide for the ST task [43], [44] and applies cross-attention regularization to the encoder representations to bridge the gap between the two modalities. We used a model trained on MuST-C En-De⁷. This model has 12 encoder and 6 decoder layers, with a dimensionality of 512 and 2048 feedforward dimensions. For the tests on 8 language pairs using MuST-C v1, we employed a multilingual ST model trained on the MuST-C v1⁸. This model also has 12 encoder and 6 decoder layers, with a dimensionality of 512 and 2048 feedforward dimensions. During inference, decoding was performed with a beam search of beam size 5.

VI. EXPERIMENTAL RESULTS

A. Translation quality

Table II shows the overall results of MuST-C v2 En-De across BLEU, BERTScore F1, and BLEURT metrics. The

leftmost columns in the table present the BLEU results. From the perspective of the SFC model, *SHAS + FTPT* generally achieved higher translation accuracy than *SHAS*. In terms of algorithms, when using *SHAS + FTPT* for SFC, pTHR and pTHR+MA tended to be either comparable to or even better than the baseline. The best BLEU score (26.30) was obtained when using the *large + all* of the *SHAS + FTPT* for SFC and pTHR+MA for the segmentation algorithm. We discuss the effects of the proposed segmentation algorithm in Section VI-B and fine-tuning the wav2vec 2.0 in Section VI-C. The middle columns in the table display the results for BERTScore F1. While the trend of the results was similar to the previously discussed BLEU scores, differences between our proposed algorithms and the baselines were more pronounced. This could potentially be attributed to the longer segments produced by the baseline algorithms. Specifically, as the translation segments become longer, there is an increased risk of misalignment during BERTScore computation, which can lead to a decrease in the score. The rightmost columns of Table II present the BLEURT results. These showed a similar trend to BERTScore, suggesting that shorter segments might receive higher evaluations in embedding-based automatic evaluations.

Table III shows the translation qualities of the segmentations of MuST-C, SHAS, and the proposed method. The SHAS's score is that reported by Tsiamas et al., and the proposed method's score is the best result from Table II. The proposed method retained 97.4% of the BLEU score for sentence-aligned speech segmentation, surpassing the 95.1% by SHAS.

B. Effectiveness of segmentation algorithm

In table II, for the baseline SFC models *SHAS* trained with fixed wav2vec 2.0 parameters (*middle* and *large*), the pTHR results had significantly lower BLEU than those of the conventional segmentation algorithms, pDAC and pSTRM. This result implies that the SFC performance that predicted the ST corpus segmentation was insufficient, and in such cases, the conventional segmentation algorithms, which heavily rely on length heuristics, had an advantage. On the other hand, as the number of layers to be trained $N_{FTLayers}$ increased, the pTHR results improved, achieving BLEU scores that were comparable to those of pDAC and pSTRM. While other algorithms demonstrated a statistically significant difference between *SHAS* and *SHAS + FTPT* at a $p < 0.05$ level, only pTHR exhibited significance at $p < 0.001$, underscoring its substantial improvement. It suggests that the need to consider segment length decreases with higher SFC accuracy.

Moreover, pTHR with a moving average (pTHR+MA) obtained the best BLEU score in most models. In particular, for a *large* model, it outperformed pTHR by more than 3 BLEU points, demonstrating the effect of smoothing the probability using the moving average to compensate for the model's low prediction accuracy. However, we found no significant difference between pTHR and pTHR+MA for *large + half*, *large + all*, etc., where there are many trainable parameters. The best parameters for each segmentation algorithm are shown in Appendix A.

⁷https://github.com/facebookresearch/fairseq/blob/main/examples/speech_text_joint_to_text/docs/ende-mustc.md

⁸https://github.com/facebookresearch/fairseq/blob/main/examples/speech_to_text/docs/mustc_example.md

TABLE II: Results by baseline model *SHAS* and proposed model *SHAS + FTPT*, and four decoding algorithms on MuST-C v2 En-De. Numbers in brackets are the number of encoder layers (fine-tuned/inherited from wav2vec 2.0). For BLEU, † and ‡ indicate statistical significance ($p < 0.05$ and $p < 0.001$, respectively) in comparison with the top row.

Model \ Decoding	BLEU				BERTScore F1				BLEURT			
	pDAC	pSTRM	pTHR	+MA	pDAC	pSTRM	pTHR	+MA	pDAC	pSTRM	pTHR	+MA
<i>SHAS</i>												
<i>middle</i> (0/16)	25.42	25.11	23.54	25.73	0.5201	0.5233	0.5324	0.5418	0.4958	0.4941	0.4992	0.5050
<i>large</i> (0/24)	24.41	25.18	21.15	24.78	0.5072	0.5378	0.5087	0.5381	0.4850	0.4975	0.4730	0.4988
<i>SHAS + FTPT</i>												
<i>middle+quarter</i> (4/16)	25.84	25.57†	25.67‡	25.96	0.5381	0.5342	0.5641	0.5592	0.5082	0.5035	0.5232	0.5213
<i>middle+half</i> (8/16)	25.75	25.52	25.92‡	26.17	0.5344	0.5343	0.5724	0.5703	0.5046	0.5046	0.5287	0.5267
<i>middle+all</i> (16/16)	25.73	25.71†	26.13‡	26.27†	0.5394	0.5401	0.5697	0.5634	0.5054	0.5029	0.5264	0.5211
<i>large+quarter</i> (6/24)	25.73	25.74†	25.73‡	26.18	0.5369	0.5420	0.5651	0.5560	0.5058	0.5054	0.5238	0.5183
<i>large+half</i> (12/24)	25.89†	25.58	26.26‡	26.15	0.5363	0.5358	0.5751	0.5623	0.5077	0.5049	0.5317	0.5205
<i>large+all</i> (24/24)	25.95†	25.70†	26.28‡	26.30†	0.5518	0.5345	0.5657	0.5698	0.5161	0.5007	0.5239	0.5257

TABLE III: BLEU scores of sentence-aligned segmentation, *SHAS*, and our method in English-to-German Translation. Numbers in parentheses are the percentages of retained BLEU scores of sentence-aligned speech segmentation

	MuST-C v2 En-De
Sentence-aligned	26.99 (100%)
<i>SHAS</i> (Tsiamas+22)	25.67 (95.1%)
Proposed method	26.30 (97.4%)

TABLE IV: Number of trainable parameters and maximum GPU memory usage for each SFC

Model	Trainable / Non-trainable parameters	GPU memory (MB)
<i>middle</i> (0/16)	8M / 215M	4,469
<i>middle+quarter</i> (4/16)	38M / 189M	15,511
<i>middle+half</i> (8/16)	59M / 173M	15,877
<i>middle+all</i> (16/16)	101M / 139M	17,053
<i>large</i> (0/24)	8M / 315M	5,716
<i>large+quarter</i> (6/24)	48M / 282M	21,570
<i>large+half</i> (12/24)	80M / 257M	23,058
<i>large+all</i> (24/24)	143M / 206M	25,272

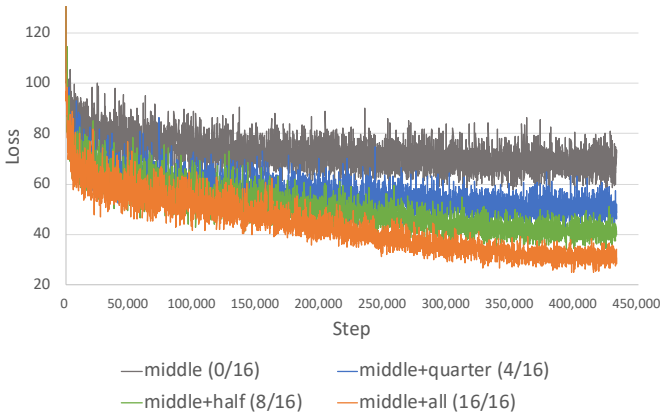


Fig. 5: Loss curves of middle SFC models

C. Effectiveness of wav2vec 2.0 fine-tuning

Table IV shows the number of trainable and non-trainable parameters and the maximum GPU memory usage for each SFC. The translation quality by pTHR is somewhat proportional to the number of trainable parameters in SFC. The explanation is that the higher the percentage of parameters that can be updated, the easier it is to fit a pre-trained speech model to the segmentation task, as shown by the loss curves in Fig. 5. Table V shows

TABLE V: Prediction performance for each SFC model

Model	Precision	Recall	F1
<i>middle</i> (0/16)	0.9894	0.9046	0.9449
<i>middle+quarter</i> (4/16)	0.9879	0.9194	0.9524
<i>middle+half</i> (8/16)	0.9861	0.9282	0.9563
<i>middle+all</i> (16/16)	0.9834	0.9344	0.9583
<i>large</i> (0/24)	0.9802	0.8532	0.9123
<i>large+quarter</i> (6/24)	0.9908	0.9074	0.9472
<i>large+half</i> (12/24)	0.9896	0.9166	0.9517
<i>large+all</i> (24/24)	0.9812	0.9381	0.9591

the prediction performance for the dev set by each SFC model. After determining the output to be 0 or 1 with a threshold value of 0.5 from the probability for each frame, we calculated the Precision, Recall, and F1 for the correct label. *SHAS* models (*middle* and *large*) have a high Precision of about 98%, but a low Recall of about 85% to 90%. Label $y = 1$ indicates that the corresponding frame is inside the segment, while $y = 0$ indicates that it is outside of it. Therefore, low Recall shows that in-segment frames are often incorrectly judged as out-of-segment. In *SHAS*, the length heuristics with pDAC mitigated the over-segmentation due to this low Recall. On the other hand, *SHAS + FTPT* models (*middle + all*, *large + all*), with fine-tuning of all the layers, improved the Recall by 3% to 7% with almost no drop in Precision. The reduction in the need for length heuristics proportional to model size, mentioned in Section VI-B, can be explained by this improvement in Recall.

D. Improving time efficiency of ST

Figure 6 shows the trade-off between time efficiency and translation quality for each segmentation algorithm with SFC model *large + all*. The number of tokens per mini-batch was set to 100,000, and ST inference was performed using NVIDIA GeForce RTX 3090 on the same computer. Segments were sorted by length before batching. The horizontal axis shows the average ST inference times of five inferences, and the vertical axis shows the BLEU. For pDAC and pSTRM, the conditions were set in the range of $max = [2, 28]$, and for pTHR and pTHR+MA, they were set in the range of $threshold = [0.1, 0.9]$. The proposed algorithms (pTHR and pTHR+MA) achieved higher translation accuracy with better time efficiency than the baseline algorithms (pDAC and pSTRM). In particular, the segments generated by pTHR were processed about 25% faster than the pDAC segments while

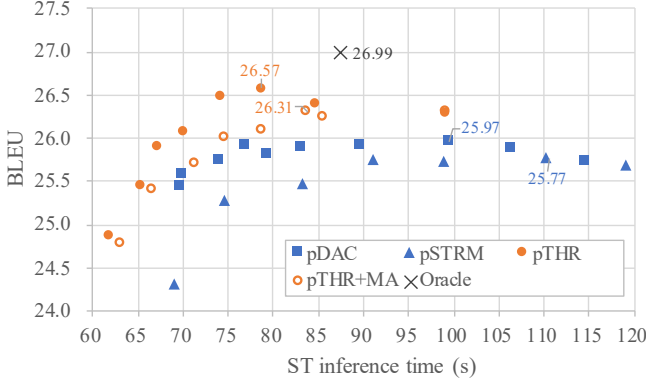


Fig. 6: ST inference time and BLEU for each segmentation method. (model=large+all)

retaining about 97% of the translation accuracy of the sentence-aligned speech segmentation. The shorter average segment length of 5.67 seconds for pTHR compared to 9.17 seconds for pDAC contributed to the higher speed because shorter segment lengths are easier to parallelize and reduce the number of autoregressions.

E. Segment length distribution

Figure 7 shows the length distribution of the segments generated by pSTRM, pTHR, and pTHR+MA using an SFC model *large + all*. pDAC (Fig. 2b) tends to produce longer segments compared to the sentence-aligned speech segmentation (Fig. 2a), as shown in Section III-C; the same is true for pSTRM (Fig. 7a). On the other hand, the pTHR (Fig. 7b) and pTHR+MA (Fig. 7c) distributions resemble that of the sentence-aligned speech segmentation, producing relatively short segments, suggesting that the segmentation of the proposed method was more faithful to sentence segmentation than SHAS.

In another aspect, longer pDAC and pSTRM segments reduced the contextual dependencies between segments, perhaps improving the translation accuracy. Therefore, combining pTHR and pTHR+MA with a context-aware ST [45], [46] might further improve the translation performance.

F. Automatic segmentation repairs improper segmentation in ST corpus segments

Although the prediction accuracy of the best SFC model has room for improvement (Table V), it maintains a BLEU score as high as 97% of the sentence-aligned speech segmentation. We conducted a case study, hypothesizing that automatic segmentation can repair the incorrect segmentation in the ST corpus. Fig. 8 shows examples of MuST-C and automatic segmentation and their ST results. Compared to the sentence segments, it can be seen that MuST-C segmentation often overlooked sentence boundaries. This was caused by audience laughter and short pauses between utterances. Such under-segmentation increases segment lengths, which often degrade the translation accuracy. In contrast, the proposed method predicted the sentence segment boundaries more accurately than MuST-C segmentation. Of course, there are over- and under-segmentations by the proposed method. However, in some cases,

the proposed method outperformed MuST-C segmentation, leading to competitive translation results.

G. Testing on other datasets

Table VI shows the results of applying a combination of two SFC models (baseline SHAS and the proposed SHAS+FTPT) and four algorithms (pDAC, pSTRM, pTHR, and pTHR+MA) to the 8 languages of MuST-C v1. No hyperparameters were fine-tuned in these tests, and all segmentation methods were applied with exactly the same configuration as used in his MuST-C v2 en-de. From the model perspective, SHAS+FTPT consistently outperformed SHAS. Observations from the algorithm side were also in line with our main experiments: the translation accuracy of the SHAS model was particularly low when using pTHR, while pTHR+MA showed translation accuracy comparable to pDAC. These results lead us to conclude that our proposed method is effective for different target languages. Table VII shows the test results for a different domain, Europarl-ST. The improvement in translation accuracy when using pTHR (from 24.06 to 25.68) suggests that our approach of unfreezing wav2vec brings about an enhancement in generalization ability, rather than mere overfitting to the task.

VII. CONCLUSION

In this study, we addressed a problem caused by the state-of-the-art speech segmentation method, SHAS, which tends to generate overly long segments, degrading the quality and time efficiency of speech translation (ST). We extended SHAS to improve ST translation accuracy and efficiency by splitting speech into shorter segments that correspond to sentences. We introduced a simple segmentation algorithm using the moving average of SFC predictions without relying on length heuristics. We also introduced efficient fine-tuning of wav2vec 2.0 to improve the SFC of SHAS and investigated the effects of model size and trainable parameters on prediction performance. Experiments using ST corpora showed that the proposed segmentation algorithm improves ST's time efficiency by generating shorter segments while maintaining translation quality comparable to existing algorithms. Experimental results also showed that fine-tuning wav2vec 2.0 improves the accuracy of SFC, with a concomitant significant improvement in ST quality.

Future research will focus on the proposed method for simultaneous speech translation. It will also investigate adapting to the noisy and multi-speaker environments, optimizing segmentation to maximize translation accuracy, and combining with context-aware ST.

ACKNOWLEDGMENTS

Part of this work was supported by JST SPRING Grant Number JPMJSP2140 and JSPS KAKENHI Grant Numbers JP21H05054 and JP21H03500.

REFERENCES

- [1] I. Tsiamas, G. I. Gállego, J. A. R. Fonollosa, and M. R. Costa-jussà, "SHAS: Approaching optimal Segmentation for End-to-End Speech Translation," in *Proc. Interspeech 2022*, 2022, pp. 106–110.

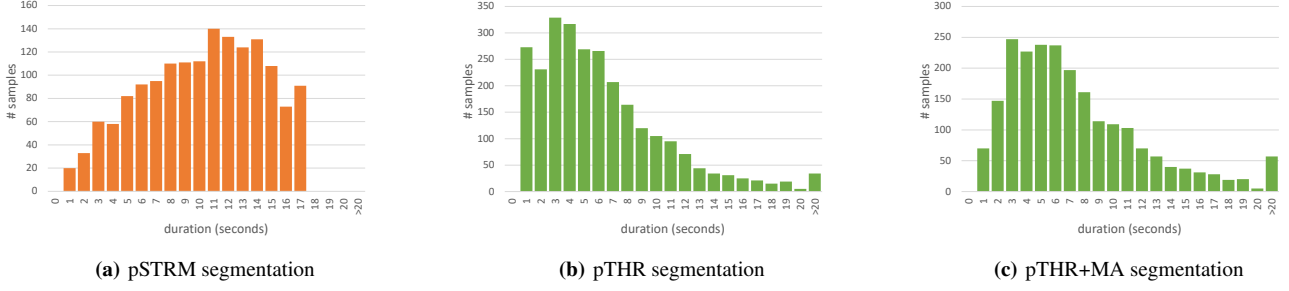


Fig. 7: Histograms of segment length in each segmentation

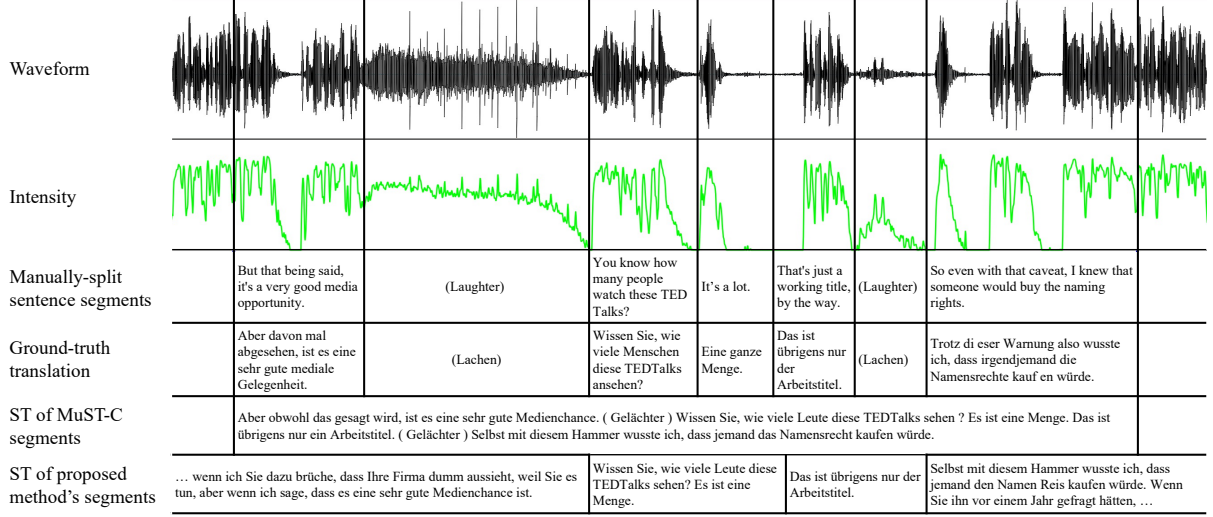


Fig. 8: Examples of segmentation and its ST result: Vertical lines indicate the split position of a waveform.

TABLE VI: Results in BLEU by *SHAS* and *SHAS + FTPT* for the 8 language pairs of MuST-C v1 tst-COMMON. Four numbers separated by slashes are the results of different algorithms (pDAC/pSTRM/pTHR/pTHR+MA).

	MuST-C v1 En-De	MuST-C v1 En-Es	MuST-C v1 En-Fr	MuST-C v1 En-It
Sentence-aligned	23.31	28.12	33.93	24.20
SHAS (<i>middle</i>)	21.75/21.42/19.69/21.93	26.69/26.43/22.84/25.73	31.36/30.84/28.69/31.70	22.67/22.22/19.68/21.93
SHAS+FTPT (<i>large+all</i>)	22.36/21.85/22.61/22.65	26.78/26.68/26.75/26.86	31.96/31.50/32.03/32.07	23.04/22.61/22.79/22.95

	MuST-C v1 En-Nl	MuST-C v1 En-Pt	MuST-C v1 En-Ro	MuST-C v1 En-Ru
Sentence-aligned	27.92	30.38	21.93	15.57
SHAS (<i>middle</i>)	26.28/25.97/22.97/25.20	28.55/28.30/25.85/28.39	20.31/19.96/18.62/20.38	14.59/14.35/12.57/14.19
SHAS+FTPT (<i>large+all</i>)	26.74/26.44/26.48/26.20	29.15/28.74/29.02/29.28	21.05/20.50/20.91/20.86	14.61/14.31/14.43/14.53

TABLE VII: Results in BLEU by *SHAS* and *SHAS + FTPT* for En-De of Europarl-ST. Four numbers separated by slashes are the results of different algorithms (pDAC/pSTRM/pTHR/pTHR+MA).

	Europarl-ST En-De
Sentence-aligned	26.17
SHAS (<i>middle</i>)	24.35/22.77/24.06/24.65
SHAS+FTPT (<i>large+all</i>)	25.13/24.15/25.68/25.48

- [2] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "MuST-C: a Multilingual Speech Translation Corpus," 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)*. Minneapolis, Minnesota: Association for Computational Linguistics, Jun. 2019, pp. 2012–2017. [Online]. Available: <https://aclanthology.org/N19-1202>
- [3] D. Wan, C. Kedzie, F. Ladhak, E. Turcan, P. Galuščáková, E. Zotkina, Z. P. Jiang, P. Bell, and K. McKeown, "Segmenting subtitles for correcting asr segmentation errors," in *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, 2021, pp. 2842–2854.
- [4] M. Sinclair, P. Bell, A. Birch, and F. McInnes, "A semi-markov model for speech segmentation with an utterance-break prior," in *Fifteenth Annual Conference of the International Speech Communication Association*, 2014.
- [5] M. Gaido, M. Negri, M. Cettolo, and M. Turchi, "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)*. Trento, Italy: Association for Computational Linguistics, 12–13 Nov. 2021, pp. 55–62. [Online]. Available: <https://aclanthology.org/2021.icnlp-1.7>
- [6] M. Paulik, S. Rao, I. Lane, S. Vogel, and T. Schultz, "Sentence segmentation and punctuation recovery for spoken language translation," in *2008 IEEE International Conference on Acoustics, Speech and Signal Processing*, 2008, pp. 5105–5108.
- [7] E. Cho, J. Niehues, and A. Waibel, "Segmentation and punctuation prediction in speech language translation using a monolingual translation system," in *Proceedings of the 9th International Workshop on Spoken Language Translation: Papers*, Hong Kong, Table of contents, Dec. 6-7 2012, pp. 252–259. [Online]. Available: <https://aclanthology.org/2012.iwslt-papers.15>

- [8] A. Baevski, Y. Zhou, A. Mohamed, and M. Auli, “wav2vec 2.0: A framework for self-supervised learning of speech representations,” *Advances in Neural Information Processing Systems*, vol. 33, 2020.
- [9] S. Mansour, “Morphotagger: Hmm-based arabic segmentation for statistical machine translation,” in *Proceedings of the 7th International Workshop on Spoken Language Translation: Papers*, 2010.
- [10] T. Nguyen and S. Vogel, “Context-based Arabic morphological analysis for machine translation,” in *CoNLL 2008: Proceedings of the Twelfth Conference on Computational Natural Language Learning*. Manchester, England: Coling 2008 Organizing Committee, Aug. 2008, pp. 135–142. [Online]. Available: <https://aclanthology.org/W08-2118>
- [11] W. Lu and H. T. Ng, “Better punctuation prediction with dynamic conditional random fields,” in *Proceedings of the 2010 conference on empirical methods in natural language processing*, 2010, pp. 177–186.
- [12] M. Diab, K. Hacioglu, and D. Jurafsky, “Automatic tagging of Arabic text: From raw text to base phrase chunks,” in *Proceedings of HLT-NAACL 2004: Short Papers*. Boston, Massachusetts, USA: Association for Computational Linguistics, May 2 - May 7 2004, pp. 149–152. [Online]. Available: <https://aclanthology.org/N04-4038>
- [13] F. Sadat and N. Habash, “Combination of Arabic preprocessing schemes for statistical machine translation,” in *Proceedings of the 21st International Conference on Computational Linguistics and 44th Annual Meeting of the Association for Computational Linguistics*. Sydney, Australia: Association for Computational Linguistics, Jul. 2006, pp. 1–8. [Online]. Available: <https://aclanthology.org/P06-1001>
- [14] E. Matusov, D. Hillard, M. Magimai-Doss, D. Hakkani-Tur, M. Ostendorf, and H. Ney, “Improving speech translation with automatic boundary prediction,” in *Proceedings of Interspeech 2007*, 2007, pp. 2449–2452.
- [15] V. K. Rangarajan Sridhar, J. Chen, S. Bangalore, A. Ljolje, and R. Chengalvarayan, “Segmentation strategies for streaming speech translation,” in *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. Atlanta, Georgia: Association for Computational Linguistics, Jun. 2013, pp. 230–238. [Online]. Available: <https://aclanthology.org/N13-1023>
- [16] M. Gaido, M. Negri, M. Cettolo, and M. Turchi, “Beyond voice activity detection: Hybrid audio segmentation for direct speech translation,” *CoRR*, vol. abs/2104.11710, 2021. [Online]. Available: <https://arxiv.org/abs/2104.11710>
- [17] H. Inaguma, B. Yan, S. Dalmia, P. Guo, J. Shi, K. Duh, and S. Watanabe, “ESPnet-ST IWSLT 2021 offline speech translation system,” in *Proceedings of the 18th International Conference on Spoken Language Translation*. Bangkok, Thailand (online): Association for Computational Linguistics, Aug. 2021, pp. 100–109. [Online]. Available: <https://aclanthology.org/2021.iwslt-1.10>
- [18] G. I. Gállego, I. Tsiamas, C. Escolano, J. A. Fonollosa, and M. R. Costa-jussà, “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)*, 2021, pp. 110–119.
- [19] T. Yoshimura, T. Hayashi, K. Takeda, and S. Watanabe, “End-to-end automatic speech recognition integrated with ctc-based voice activity detection,” in *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2020, pp. 6999–7003.
- [20] E. Cho, J. Niehues, K. Kilgour, and A. Waibel, “Punctuation insertion for real-time spoken language translation,” in *Proceedings of the Eleventh International Workshop on Spoken Language Translation*, 2015.
- [21] T.-L. Ha, J. Niehues, E. Cho, M. Mediani, and A. Waibel, “The kit translation systems for iwslt 2015,” in *Proceedings of the Eleventh International Workshop on Spoken Language Translation*, 2015.
- [22] E. Cho, J. Niehues, and A. Waibel, “NMT-Based Segmentation and Punctuation Insertion for Real-Time Spoken Language Translation,” in *Proceedings of Interspeech 2017*, 2017, pp. 2645–2649.
- [23] A. Stolcke and E. Shriberg, “Automatic linguistic segmentation of conversational speech,” in *Proceeding of Fourth International Conference on Spoken Language Processing. ICSLP’96*, vol. 2. IEEE, 1996, pp. 1005–1008.
- [24] X. Wang, A. Finch, M. Utiyama, and E. Sumita, “An efficient and effective online sentence segmenter for simultaneous interpretation,” in *Proceedings of the 3rd Workshop on Asian Translation (WAT2016)*, 2016, pp. 139–148.
- [25] X. Wang, M. Utiyama, and E. Sumita, “Online sentence segmentation for simultaneous interpretation using multi-shifted recurrent neural network,” in *Proceedings of Machine Translation Summit XVII Volume 1: Research Track*, 2019, pp. 1–11.
- [26] J. Iranzo-Sánchez, A. Giménez Pastor, J. A. Silvestre-Cerdà, P. Baquero-Arnal, J. Civera Saiz, and A. Juan, “Direct segmentation models for streaming speech translation,” in *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*. Online: Association for Computational Linguistics, Nov. 2020, pp. 2599–2611. [Online]. Available: <https://aclanthology.org/2020.emnlp-main.206>
- [27] R. Fukuda, K. Sudoh, and S. Nakamura, “Speech Segmentation Optimization using Segmented Bilingual Speech Corpus for End-to-end Speech Translation,” in *Proc. Interspeech 2022*, 2022, pp. 121–125.
- [28] N. Houlsby, A. Giurghi, S. Jastrzebski, B. Morrone, Q. De Laroussilhe, A. Gesmundo, M. Attariyan, and S. Gelly, “Parameter-efficient transfer learning for nlp,” in *International Conference on Machine Learning*. PMLR, 2019, pp. 2790–2799.
- [29] J. He, C. Zhou, X. Ma, T. Berg-Kirkpatrick, and G. Neubig, “Towards a unified view of parameter-efficient transfer learning,” in *International Conference on Learning Representations*, 2021.
- [30] Y.-L. Sung, J. Cho, and M. Bansal, “Lst: Ladder side-tuning for parameter and memory efficient transfer learning,” *Advances in Neural Information Processing Systems*, vol. 35, pp. 12 991–13 005, 2022.
- [31] K. Cho, B. van Merriënboer, D. Bahdanau, and Y. Bengio, “On the properties of neural machine translation: Encoder–decoder approaches,” in *Proceedings of SSS-8, Eighth Workshop on Syntax, Semantics and Structure in Statistical Translation*. Doha, Qatar: Association for Computational Linguistics, Oct. 2014, pp. 103–111. [Online]. Available: <https://aclanthology.org/W14-4012>
- [32] G. U. Yule, “The applications of the method of correlation to social and economic statistics,” *Journal of the Royal Statistical Society*, vol. 72, no. 4, pp. 721–730, 1909.
- [33] I. Tsiamas, G. I. Gállego, C. Escolano, J. Fonollosa, and M. R. Costa-jussà, “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)*. Dublin, Ireland (in-person and online): Association for Computational Linguistics, May 2022, pp. 265–276. [Online]. Available: <https://aclanthology.org/2022.iwslt-1.23>
- [34] J. Iranzo-Sánchez, J. A. Silvestre-Cerdà, J. Jorge, N. Roselló, A. Giménez, A. Sanchis, J. Civera, and A. Juan, “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)*. IEEE, 2020, pp. 8229–8233.
- [35] E. Matusov, G. Leusch, O. Bender, and H. Ney, “Evaluating machine translation output with automatic sentence segmentation,” in *Proceedings of the Second International Workshop on Spoken Language Translation*, 2005.
- [36] K. Papineni, S. Roukos, T. Ward, and W.-J. Zhu, “Bleu: a method for automatic evaluation of machine translation,” in *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics, Jul. 2002, pp. 311–318. [Online]. Available: <https://aclanthology.org/P02-1040>
- [37] M. Post, “A call for clarity in reporting BLEU scores,” in *Proceedings of the Third Conference on Machine Translation: Research Papers*. Brussels, Belgium: Association for Computational Linguistics, Oct. 2018, pp. 186–191. [Online]. Available: <https://aclanthology.org/W18-6319>
- [38] T. Zhang, V. Kishore, F. Wu, K. Q. Weinberger, and Y. Artzi, “Bertscore: Evaluating text generation with bert,” *arXiv preprint arXiv:1904.09675*, 2019.
- [39] T. Sellam, D. Das, and A. Parikh, “BLEURT: Learning robust metrics for text generation,” in *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Online: Association for Computational Linguistics, Jul. 2020, pp. 7881–7892. [Online]. Available: <https://aclanthology.org/2020.acl-main.704>
- [40] A. Babu, C. Wang, A. Tjandra, K. Lakhotia, Q. Xu, N. Goyal, K. Singh, P. von Platen, Y. Saraf, J. Pino, A. Baevski, A. Conneau, and M. Auli, “XLS-R: Self-supervised Cross-lingual Speech Representation Learning at Scale,” in *Proc. Interspeech 2022*, 2022, pp. 2278–2282.
- [41] Y. Tang, J. Pino, X. Li, C. Wang, and D. Genzel, “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)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 4252–4261. [Online]. Available: <https://aclanthology.org/2021.acl-long.328>
- [42] C. Wang, Y. Tang, X. Ma, A. Wu, D. Okhonko, and J. Pino, “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*. Suzhou, China: Association for Computational Linguistics, Dec. 2020, pp. 33–39. [Online]. Available: <https://aclanthology.org/2020.aacl-demo.6>

- [43] Y. Liu, H. Xiong, J. Zhang, Z. He, H. Wu, H. Wang, and C. Zong, “End-to-end speech translation with knowledge distillation,” *Proc. Interspeech 2019*, pp. 1128–1132, 2019.
- [44] M. Gaido, M. A. Di Gangi, M. Negri, and M. Turchi, “On knowledge distillation for direct speech translation,” *Computational Linguistics CLiC-it 2020*, p. 211, 2020.
- [45] M. Gaido, M. A. D. Gangi, M. Negri, M. Cettolo, and M. Turchi, “Contextualized Translation of Automatically Segmented Speech,” in *Proc. Interspeech 2020*, 2020, pp. 1471–1475. [Online]. Available: <http://dx.doi.org/10.21437/Interspeech.2020-2860>
- [46] B. Zhang, I. Titov, B. Haddow, and R. Sennrich, “Beyond sentence-level end-to-end speech translation: Context helps,” in *Proceedings of the 6522th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*. Online: Association for Computational Linguistics, Aug. 2021, pp. 2566–2578. [Online]. Available: <https://aclanthology.org/2021.acl-long.200>

APPENDIX

A. Hyperparameters

TABLE VIII: Best parameters of each algorithm chosen by dev set and results in BLEU and number of segments (#seg).

(a) pDAC

Model	BLEU	#seg	max
l _{na} _116_ft0	25.42	1073	26
l _{na} _116_ft4	25.84	1656	16
l _{na} _116_ft8	25.75	1268	22
l _{na} _116_ft16	25.73	1855	14
l _{na} _124_ft0	24.41	1031	28
l _{na} _124_ft6	25.73	1636	16
l _{na} _124_ft12	25.89	1389	20
l _{na} _124_ft24	25.95	2279	10

(b) pSTRM

Model	BLEU	#seg	max
l _{na} _116_ft0	25.11	953	28
l _{na} _116_ft4	25.57	1186	22
l _{na} _116_ft8	25.52	1294	20
l _{na} _116_ft16	25.71	1300	20
l _{na} _124_ft0	25.18	1648	16
l _{na} _124_ft6	25.7	1584	16
l _{na} _124_ft12	25.58	1304	20
l _{na} _124_ft24	25.70	1292	20

(c) pTHR

Model	BLEU	#seg	thr	ma
l _{na} _116_ft0	23.54	3639	0.1	0
l _{na} _116_ft4	25.67	2353	0.1	0
l _{na} _116_ft8	25.92	2703	0.2	0
l _{na} _116_ft16	26.13	2525	0.2	0
l _{na} _124_ft0	21.15	4860	0.1	0
l _{na} _124_ft6	25.73	2588	0.2	0
l _{na} _124_ft12	26.26	2638	0.2	0
l _{na} _124_ft24	26.28	2149	0.1	0

(d) pTHR+MA

Model	BLEU	#seg	thr	ma
l _{na} _116_ft0	25.73	1944	0.1	0.2
l _{na} _116_ft4	25.96	2055	0.1	0.1
l _{na} _116_ft8	26.17	2464	0.2	0.1
l _{na} _116_ft16	26.27	1981	0.1	0.1
l _{na} _124_ft0	24.78	1816	0.1	0.4
l _{na} _124_ft6	26.13	1914	0.1	0.1
l _{na} _124_ft12	26.15	2005	0.1	0.1
l _{na} _124_ft24	26.30	2044	0.1	0.1



Ryo Fukuda received a B.S. degree in engineering from Toyohashi University of Technology in 2019 and master's degree in engineering from the Nara Institute of Science and Technology in 2021. He is currently working toward the Ph.D. degree with the Nara Institute of Science and Technology. His research interests include speech and natural language processing with speech translation.



Katsuhito Sudoh received a B.S. degree in engineering and master's and Ph.D. degrees in informatics from Kyoto University in 2000, 2002, and 2015.

He was with NTT Communication Science Laboratories from 2002 to 2017 and is currently an Associate Professor with the Nara Institute of Science and Technology. His research interests include machine translation and natural language processing.

Dr. Sudoh is a member of the Association for Computational Linguistics (ACL) and the International Speech Communication Association (ISCA).



Satoshi Nakamura (Fellow, IEEE) received a B.S. degree from the Kyoto Institute of Technology in 1981 and a Ph.D. degree from Kyoto University in 1992.

From 1994 to 2000, he was an Associate Professor with the Graduate School of Information Science, Nara Institute of Science and Technology, Japan. He was the Director of the ATR Spoken Language Communication Research Laboratories from 2000 and 2008 and the Vice President of ATR from 2007 to 2008. He was the Director General of the Keihanna Research Laboratories and the Executive Director of

the Knowledge Creating Communication Research Center, National Institute of Information and Communications Technology, Japan from 2009 to 2010. He is currently a Professor with the Graduate School of Science and Technology, Nara Institute of Science and Technology, a Project Leader of the Tourism Information Analytics Team, RIKEN, Center for Advanced Intelligence Project AIP, an Honorary Professor with the Karlsruhe Institute of Technology, Germany, and an ATR Fellow. He is also a Director of the Augmented Human Communication Laboratory and a Full Professor with the Graduate School of Information Science, Nara Institute of Science and Technology. His research interests include modeling and systems of speech-to-speech translation and speech recognition.

Prof. Nakamura is one of the leaders of speech-to-speech translation research and has been serving on various worldwide speech-to-speech translation research projects, including C-STAR, IWSLT, and A-STAR. He has been an Elected Board Member of the International Speech Communication Association (ISCA) since June 2011, an IEEE Signal Processing Magazine Editorial Board Member since April 2012, and an IEEE SPS Speech and Language Technical Committee Member since 2013. He has received the following awards: the Yamashita Research Award, the Kiyasu Award from the Information Processing Society of Japan, the Telecom System Award, the AAMT Nagao Award, the Docomo Mobile Science Award in 2007, and the ASJ Award for Distinguished Achievements in Acoustics. He received a Commendation for Science and Technology from the Minister of Education, Science and Technology and the Commendation for Science and Technology by the Minister of Internal Affairs and Communications. He also received the LREC Antonio Zampolli Award in 2012.