Learning Adaptive Segmentation Policy for End-to-End Simultaneous Translation

Ruiqing Zhang, Zhongjun He[*], Hua Wu and Haifeng Wang
Baidu Inc. No. 10, Shangdi 10th Street, Beijing, 100085, China
{zhangruiqing01, hezhongjun, wu_hua, wanghaifeng}@baidu.com

Abstract

End-to-end simultaneous speech-to-text translation aims to directly perform translation from streaming source speech to target text with high translation quality and low latency. A typical simultaneous translation (ST) system consists of a speech translation model and a policy module, which determines when to wait and when to translate. Thus the policy is crucial to balance translation quality and latency. Conventional methods usually adopt fixed policies, e.g. segmenting the source speech with a fixed length and generating translation. However, this method ignores contextual information and suffers from low translation quality. This paper proposes an adaptive segmentation policy for end-to-end ST. Inspired by human interpreters, the policy learns to segment the source streaming speech into meaningful units by considering both acoustic features and translation history, maintaining consistency between the segmentation and translation. Experimental results on English-German and Chinese-English show that our method achieves a good accuracy-latency trade-off over recently proposed state-of-the-art methods.

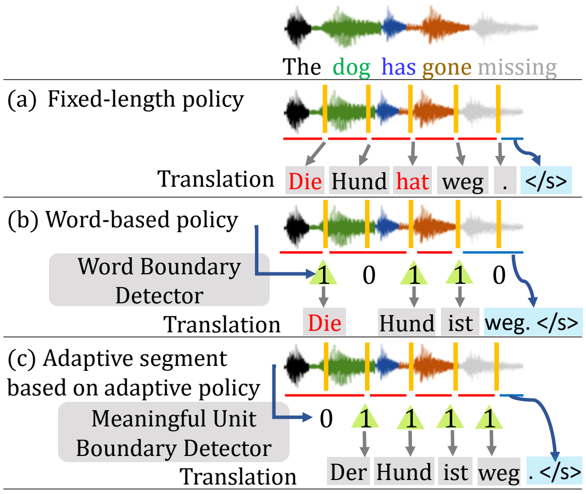


Figure 1: An En-De example illustrates three segmentation policies for end-to-end simultaneous translation. The yellow line indicates the sampling frequency of the source speech. (a) Fixed-length policy generates a target word for each interval. (b) Word-based policy detects word boundaries and generates target word once a source word is detected (green triangle). (c) Our method detects meaningful units and generates translation. Both the Fixed-length policy and Word-based policy incorrectly generate "Die" without consideration of following context[1]. The last blue blocks denote the translation after receiving the complete speech. Mistranslated words are annotated in red.

1 Introduction

Recent years have witnessed extensive studies and rapid progress of Simultaneous translation (ST). It aims to perform translation from source speech into the target language with high quality and low latency and is widely used in many scenarios, such as international conferences, press releases, etc.

Generally, the research of ST falls into two categories: the cascaded method, and the end-to-end method. The cascaded method consists of an automatic speech recognition (ASR) model which transcribes the source speech into source streaming text (Moritz et al., 2020; Wang et al., 2020b; Li et al., 2020a), and a followed-by machine translation (MT) model that generates translation based on the ASR output. Since there are no sentence or segment boundaries in the streaming source text output by ASR, a segmentation policy is required to link the ASR and the MT to determine when to read more source tokens and when to start translation (Oda et al., 2014; Dalvi et al., 2018; Ma et al., 2019; Arivazhagan et al., 2019; Zhang et al., 2020; Wilken et al., 2020). However, cascaded methods

[*] Corresponding author.

[1] In German, each singular noun is assigned a gender, either *masculine*, *feminine*, or *neuter*, which determines whether the definite article (like "The" in English) preceding the noun is "*Der*", "*Die*" or "*Das*". Therefore, translating "The" hastily without receiving the following noun may cause mistranslation.

Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics
Volume 1: Long Papers, pages 7862 - 7874
May 22-27, 2022 ©2022 Association for Computational Linguistics

face two main challenges. One is the error propagation that the ASR errors may hurt the translation quality. The other is the increase of latency because the translation model has to wait for the output of the ASR model.

To overcome these limitations, the end-to-end method attempts to directly translate from source speech to target text, without explicitly transcribing the source speech (Bansal et al., 2018; Di Gangi et al., 2019b; Jia et al., 2019). To balance the translation quality and latency, the key challenge lies in the segmentation policy that determines the translation boundaries of the speech frames.

Most of the previous work used fixed policies. Some of them take fixed-length policy (Nguyen et al., 2021; Ma et al., 2020b, 2021) that splits speech at a fixed frequency, for example, to generate one target word every T_s ms (Figure 1 (a)). Other work adopts word-based policy that splits the speech into words and generates one target word whenever a new source word is detected, which calls for an auxiliary source word detector (Ren et al., 2020; Elbayad et al., 2020; Ma et al., 2020b; Zeng et al., 2021; Chen et al., 2021), see Figure 1 (b). However, both the above methods are "hard" policies, which do not consider the contextual information and result in low translation quality (Arivazhagan et al., 2019; Zhang et al., 2020).

In this paper, we propose an adaptive segmentation policy for end-to-end simultaneous translation based on *Meaningful Unit* (MU). The idea is borrowed from human interpreters, who do interpretation based on a unit with clear meaning rather than fixed frame length or word. We model the speech segmentation policy as a binary classification that determines whether a speech segment is an MU. Once an MU is detected, it is fed into an end-to-end speech translation model, as illustrated in Figure 1 (c). We propose a supervised training method, using both acoustic features and translation features to train the policy. Besides, we propose an incremental decoding method to construct training data from speech and translation pairs. Concretely, we first train a full speech translation model $\mathcal{M}_{ST}$, and then gradually expand speech frames to simulate simultaneous translation. When the translation of the current speech segment is a prefix of the full speech translation, the segment is extracted as an MU. At inference time, we employ the same $\mathcal{M}_{ST}$ to maintain the consistency between segmentation and translation. Our method is more flexible than fixed policies, as it dynamically detects meaningful units according to contextual information. Experiments on two language pairs show that the proposed approach outperforms the strong baselines in balancing translation quality and latency.

2 Related Work

Cascade Simultaneous Translation. To eliminate the impact of ASR errors, most previous work of cascade ST use golden transcript, rather than ASR result, to explore different read/write policies in ST. Existing policies can be classified into two categories: 1) The ***fixed policy*** segments the source text based on fixed lengths (Ma et al., 2019; Dalvi et al., 2018). For example, *wait-k* (Ma et al., 2019) is a typical *fixed policy* that first read k source words, then generates one target word immediately after receiving one subsequent source word. 2) The ***adaptive policy*** learns to segment the source text according to its context (Oda et al., 2014; Cho and Esipova, 2016; Gu et al., 2017; Arivazhagan et al., 2019; Ma et al., 2020a; Zhang et al., 2020). It has been proven that the adaptive policy is more effective than the fixed policy in balancing translation quality and latency (Zhang et al., 2020).

End-to-End Simultaneous Translation. The method has shown great potential over the cascaded method (Bérard et al., 2016; Weiss et al., 2017; Bansal et al., 2018; Jia et al., 2019; Wang et al., 2020a; Li et al., 2020b; Ansari et al., 2020). End-to-end ST contains a speech translation model, along with a policy to decide when to translate. However, most previous studies are based on fixed-length policy that translate every T_s ms (Nguyen et al., 2021; Ma et al., 2020b), or decide to translate whenever a fixed number of words are detected (Ren et al., 2020; Elbayad et al., 2020; Zeng et al., 2021; Ma et al., 2021; Chen et al., 2021), following the ***fixed policy*** of cascade ST systems.

This paper presents an adaptive policy for end-to-end simultaneous translation. We are motivated by an adaptive policy proposed for cascade ST (Zhang et al., 2020), which proposed to perform translation when a source text segment is detected to be a unit with clear meaning. However, there are three main differences. First, our method is proposed for end-to-end ST, while Zhang et al. (2020) is for cascade ST. Second, our method directly detects MU on speech rather than on the streaming text of the output of ASR. Third, we propose a multi-modal MU detection model using both acoustic features and translation history.

3 Adaptive Speech Segmentation Policy

The overall framework of our adaptive speech segmentation policy is illustrated in Figure 2. Given a streaming speech s, we incrementally detect whether a speech clip $s_{\leq t}$ $(t = 1, 2, ...)$ is an MU, where $s_{\leq t}$ denotes the head $t\mathcal{F}$ frames of s, and $\mathcal{F}$ is the detection interval. Once an MU is detected, the speech translation model produces its translation y' with the translation history y_p force decoded as a translation prefix. Meanwhile, y' is displayed to users and added to the translation history y_p to improve MU detection. In the following, we first introduce our MU detection model (Section 3.1), then propose a method to construct MU training data (Section 3.2). Finally, we describe the training details in Section 3.3.

3.1 Multi-modal MU Detection

We model the MU detection as a classification problem. Given a source speech s, the detector incrementally reads speech clips at each time t, to make a decision whether $s_{\leq t}$ is an MU.

We propose a multi-modal detector that uses both acoustic features and translation history. See the bottom green block of Figure 2 for illustration. For the acoustic feature extractor E_a^f, we use stacked temporal convolutional layers performed on raw speech features (80-channel log-mel filterbanks). Each of the convolutional layers is followed by layer normalization and a GELU activation function (Hendrycks and Gimpel, 2016), following Baevski et al. (2020). For the context feature extractor E_t^f, we use a trainable word embedding layer.

The outputs of the feature extractors are fed to the acoustic encoder E_a^e and context encoder E_t^e to generate latent representation, respectively. We add a position embedding to the textual embedding, as in BERT (Devlin et al., 2019), and add a convolutional layer as the relative positional embedding to the acoustic embedding, similar to Mohamed et al. (2019); Baevski et al. (2019). Both E_t^e and E_a^e follow the Transformer architecture (Sperber et al., 2018; Devlin et al., 2019).

$$\begin{aligned} c_y &= E_t^e(E_t^f(y_p)) \\ c_s &= E_a^e(E_a^f(s_{\leq t})) \end{aligned} \tag{1}$$

where $c_y = \{c_y^1, ..., c_y^N\}$ is the textual encoding of $y_p = \{y_p^1, y_p^2, ..., y_p^N\}$, and $c_s = \{c_s^1, ..., c_s^T\}$ is for speech encoding[2].

Next, we add different type embeddings e_{Type} ($e_{[TXT]}$ and $e_{[AUD]}$) to both text and audio embeddings to indicate their source type. These two sequences are then concatenated and fed to a 6-layer Transformer for cross-modal fusion. Special tokens [*CLS*] and [*SEP*] are added in this process following BERT (Devlin et al., 2019). The final hidden state corresponding to [*CLS*] is used as the aggregated sequence representation to predict the classification result l' using a fully-connected layer, followed by a softmax.

$$l' = \text{Softmax}(fc(E_{mm}([c_y; c_s] + e_{Type}))) \tag{2}$$

where E_{mm} denotes the multi-modal fusion Transformer and fc performs a fully-connection layer.

3.2 Constructing MU Training Data

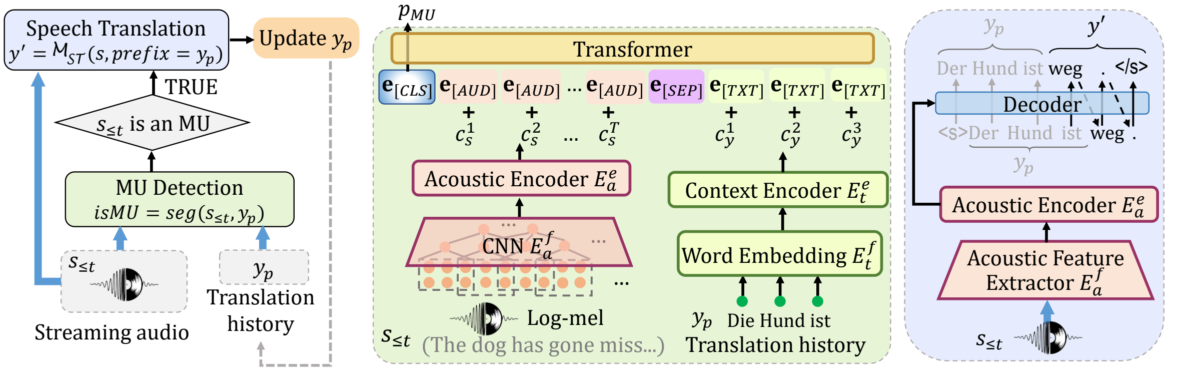
Since there are no standard MU segmentation training corpora, we propose a simple method to automatically extract meaningful speech units to construct MU training samples.

We expect that MUs can be translated properly without waiting for future speech. Therefore, we define MU as *the minimum speech segment whose translation will not be changed by subsequent speech*. This requires MUs to contain enough information to generate stable translation. Accordingly, we propose to extract meaningful speech units by comparing the translation of every speech prefix segment and the full-speech translation with a pre-trained speech translation model $\mathcal{M}_{\mathcal{ST}}$. For a speech segment $s_{\leq t}$, if its translation $y' = \mathcal{M}_{\mathcal{ST}}(s_{\leq t})$ is a prefix of the full-speech translation $\widetilde{y} = \mathcal{M}_{\mathcal{ST}}(s)$, we identify that $s_{\leq t}$ is sufficient to provide a stable translation and annotate it to an MU.

We propose an incremental-translation paradigm. We incrementally translate $s_{\leq t}$, $t = 1, 2, ...$, to judge whether its translation y' is a prefix of $\widetilde{y}$. If so, we extract $s_{\leq t}$ as an MU, and force-decode its translation y' as a prefix in detecting subsequent speech segments. This is to keep consistent with the force-decoding strategy at the inference stage.

Moreover, while comparing y' with $\widetilde{y}$, as illustrated in Figure 3, we propose a *tail-truncation* strategy that discards the last k words from the partial decoding results y'. This is to avoid translation

[2] Note that the length of acoustic encoding T is not equal to the number of source frames $t\mathcal{F}$ for the temporal sampling of convolutional feature extractor layers.



(a). The inference framework (b). The multi-modal MU Detection (c). The Speech Translation

Figure 2: (a). The overall framework of our proposed adaptive policy for end-to-end ST includes an MU detection module and a speech translation module. Once an MU $s_{\le t}$ is detected, it will be translated with the translation history y_p decoded first, then decode until EOS (</s>). The new translation y' will be added to the translation history for detecting subsequent MUs. (b). The MU detection module performs a multi-modal classification based on acoustic features c_s^i and translation context c_y^i. (c). The speech translation module shares E_a^f and E_a^e with the MU Detection module. It first force decodes y_p (in gray) and then generates y' (in black) at inference time.

errors caused by ambiguous speech fragments. For example, $s_{\le 3}$ pronounces "*The dog has*" is translated to "Der Hund hat", which is not a prefix of the full-speech translation $\widetilde{y}$ due to the current received speech is ambiguous[3]. However, after truncated the tail word, y' turn to be "Der Hund" and becomes a prefix of $\widetilde{y}$. Therefore, discarding tail words from y' enables the model to discover translation that partially matches $\widetilde{y}$ in advance, thus shortening the granularity of extracted MUs and reducing latency. At the inference stage, we also remove tail k words from the translation of detected MUs.

With the incremental-translation paradigm, we can extract four MUs for the example in Figure 3, i.e., $s_{\le 2}$, $s_{\le 3}$, $s_{\le 4}$ and $s_{\le 5}$.

3.3 Training with Shared Acoustic Encoders

Our pre-trained speech translation model $\mathcal{M}_{ST}$ includes an acoustic feature extractor E_a^f, an acoustic encoder E_a^e, and a textual translation decoder, as shown in Figure 2(c). E_a^f and E_a^e are shared with the multi-modal MU detection model so that the acoustic forward computation can be shared at inference time. The translation decoder is based on Transformer, which links the acoustic encoder through cross-attention (Vaswani et al., 2017).

Note that to keep MU detection consistency in both training and decoding, we initialize the acoustic feature extractor and context encoder with the weights from $\mathcal{M}_{ST}$, and keep them frozen in training MU detection, instead of joint training with $\mathcal{M}_{ST}$.

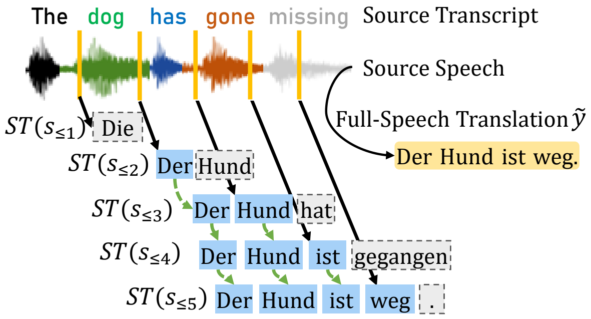

The dog has gone missing Source Transcript
Source Speech
Full-Speech Translation $\widetilde{y}$
Der Hund ist weg.
$ST(s_{\le 1})$ Die
$ST(s_{\le 2})$ Der Hund
$ST(s_{\le 3})$ Der Hund hat
$ST(s_{\le 4})$ Der Hund ist gegangen
$ST(s_{\le 5})$ Der Hund ist weg .

Figure 3: A running example of extracting MUs. While translating each speech segment $s_{\le t}$, the last k words will be discarded (in gray) from the generated translation (k=1 in this example). If the rest decoding result (in blue) is a prefix of the full-speech translation (in yellow), then $s_{\le t}$ is annotated as an MU, and the tail-truncated translation is force-decoded (green dashed lines) in translating subsequent speech segments. Note that the *Source Transcript* is invisible in extracting MUs and here is shown only for illustration.

Formally, the $\mathcal{M}_{ST}$ model optimizes the two acoustic encoders and the translation decoder first with auto-regressive loss $\mathcal{L}_{ST}$:

$$\mathcal{L}_{ST} = -\sum_{(s,y)\in\mathcal{D}_{ST}} \sum_{i=1}^{N} \log p(y_i | s, y_{<i}; \theta_{ae}, \theta_{td}) \tag{3}$$

Then the MU detection model is optimized without gradients back-propagated to the acoustic encoders:

[3] The English word "has" can be translated to either "ist" or "hat" in most cases, depending on what follows.

$$\mathcal{L}_{MU} = -\sum_{(s,y_p,l)\in\mathcal{D}_{MU}} \log p(l|s,y_p;\theta_{te},\theta_{mm}) \tag{4}$$

where θ_{ae} denotes weights of the two acoustic encoders, θ_{td} is the translation decoder of $\mathcal{M}_{ST}$, θ_{te} is the textual encoders for y_p and θ_{mm} is the weights of the multi-modal fusion. $\mathcal{D}_{ST}$ is the speech translation dataset and $\mathcal{D}_{MU}$ contains training triplets generated by the pre-trained $\mathcal{M}_{ST}$ model.

4 Experiments

We carry out experiments on English-German (En-De) and Chinese-English (Zh-En) simultaneous translation. We use sacreBLEU (Post, 2018) to evaluate the translation performance and the acoustic average lagging (AL) (Ma et al., 2019, 2020b) as the latency metric. The AL measures the system lagging behind an "ideal policy", which produces translation at the same speed as the audio received.

4.1 Data Settings

We evaluate our method on MuST-C (Di Gangi et al., 2019a) En-De dataset and BSTC (Zhang et al., 2021a) Zh-En dataset. To compare with the previous methods, we carried out experiments under two settings: the **Limited-training-corpora setting** that constrains the training data to a limited set of corpora, and the **Open-training-corpora setting** uses more data. For En-De, we set the training data of Limited-training-corpora setting as the training set of MuST-C, a dataset consisting of 408 hours of speech with transcription and translation, while the experiments with the Open-training-corpora setting use unlimited datasets of up to 1,302 hours of speech. See appendix A.1 for detail.

4.2 Model Settings

We compare our method with previous strong ST approaches. Methods listed with "*" are carried out under the Open-training-corpora setting, while others use the Limited-training-corpora setting.

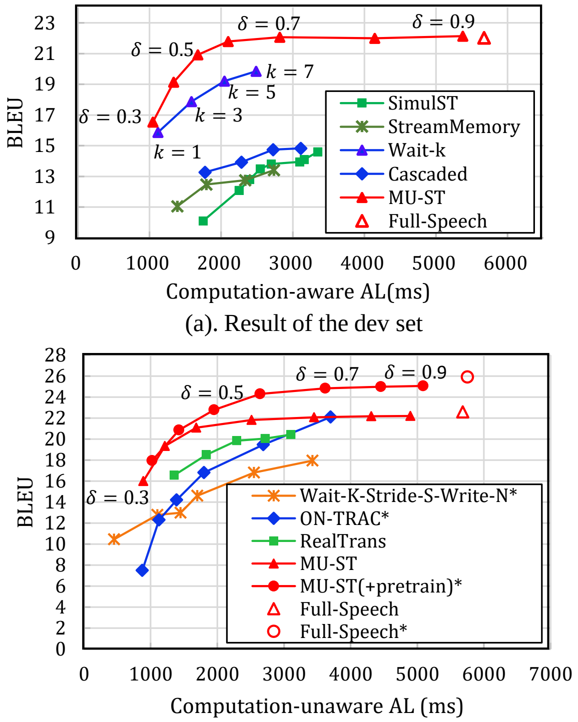
- *Wait-k* (Chen et al., 2021) integrates the *wait-k* (Ma et al., 2019) policy into end-to-end speech translation with an additional ASR module to detect the number of source words within the streaming speech.

Arch	**Speech Translator**			**Speech Segmentor**	
	Feature Extractor	Context Encoder	ST Decoder	Context Encoder	Fusion Encoder
Base	CNN-2	T-12	T-6	T-6	T-6
Big	CNN-7	T-24	T-12	T-6	T-6

Table 1: The two architectures of our method. "n" in "T-n" and "CNN-n" represents the number of stacked Transformer and stacked CNN layers, respectively.

- *SimulST* (Ma et al., 2020b) takes the fixed-length policy that translates out one token every T_s ms. We set T_s to 280 following their best experimental settings.
- *StreamMemory* (Ma et al., 2021) proposes an end-to-end speech translation model with augmented memory, which stores previous states of streaming speech to reduce the computation cost. They use the same fixed-length policy as in *SimulST*.
- *RealTranS* (Zeng et al., 2021) proposes a fixed policy (Wait-k-Stride-N) for an end-to-end ST that triggers translation based on the number of words within the streaming speech, which is detected based on a CTC module built on top of the speech translation encoder.
- *Wait-K-Stride-S-Write-N** (Nguyen et al., 2021) proposes a fixed-length policy for end-to-end ST that first wait for *K* frames, then alternatively decoding *N* target words and reading *S* frames.
- *ON-TRAC** (Elbayad et al., 2020): A cascade system that achieved the **first-place** of the IWSLT2020 En-De simultaneous translation shared task (Federico et al., 2020). It takes a fixed policy (*wait-k*) to link the ASR output and the MT module.
- *MU-ST*: Our proposed method that triggers the speech translator with an MU-based adaptive policy. The $\mathcal{M}_{ST}$ model is trained from scratch.
- *MU-ST*(*+pretrain*)*: To compare with methods of the Open-training-corpora setting, we take pre-training techniques in training the speech translator $\mathcal{M}_{ST}$.

We train *MU-ST* and *MU-ST*(*+pretrain*)* with different model architectures. The $\mathcal{M}_{ST}$ model of *MU-ST* is first pre-trained with an ASR task



(a). Result of the dev set

(b). Result of the tst-COMMON set

Figure 4: The Translation quality (BLEU) vs. latency (AL) evaluated on MuST-C En-De. The results marked with * are of the Open-training-corpora setting, while others only use the MuST-C as the training corpus. The results of all the comparison methods are excerpted from the corresponding papers[7].

to enhance the acoustic representations, as performed in *SimulST* (Ma et al., 2020b), *StreamMemory* (Ma et al., 2021) and *RealTranS* (Zeng et al., 2021). For *MU-ST(+pretrain)**, we follow the recently proposed speech translation pre-training method (Li et al., 2020b) to initialize the encoder with wav2vec2.0[4] (Baevski et al., 2020) and initialize the decoder with mBART50[5] (Tang et al., 2020), then fine-tune with speech translation corpora. As listed in Table 1, *MU-ST* takes the *Base* model and *MU-ST(+pretrain)** takes the *Big* version[6]. We set the number of truncated words *k*=2 in *tail-truncation* and the length of speech clips $T_s = 250ms$ as default.

4.3 En-De Experiment

Figure 4 shows the results on MuST-C dev set and tst-COMMON set. We calculate the computation-aware latency on the MuST-C dev set and computation-unaware latency on the MuST-C tst-COMMON set, to be consistent with previous work. The difference between them is whether the model inference time is taken into account.

MU-ST achieves higher translation quality under the same latency on both datasets. δ denotes the probability threshold of MU detector, i.e., $\delta = [0.3, 0.4, ..., 0.9]$ corresponds to the results of taking $p(l = 1|s, y_p) > \delta$ as the criterion of determining s to be an MU. Small δ produces fine-grained speech segments and small delay, but if some ambiguous speech segments are incorrectly recognized as MUs, it will result in poor translation quality.

On the dev set, we compare *MU-ST* with *Wait-k* (Chen et al., 2021), *SimulST* (Ma et al., 2020b) and *StreamMemory* (Ma et al., 2021). *SimulST* and *StreamMemory* takes fixed-length speech policy (Figure 1 (a)) while *Wait-k* performs wait-k based on the number of words detected from an ASR module (Figure 1 (b)). We also plot the result of a Cascaded system based on textual Wait-K (Chen et al., 2021) . We observed that:

- Our adaptive policy outperforms the *Wait-k* methods, and the *Wait-k* approaches are superior to the fixed-length methods.

- We report the result of translating the whole speech without segmentation in the "full-speech translation". *MU-ST* approaches the BLEU of full-speech translation as early as $\delta = 0.6$, indicating that our method can achieve comparable BLEU with full-speech translation with a very small latency (about 2100ms), while other methods still have a large gap with the full-speech translation under corresponding delay.

On the tst-COMMON set, we compare *MU-ST* with three fixed-policy methods. *ON-TRAC** and *RealTranS* follow the wait-k policy, while *Wait-K-Stride-S-Write-N** takes the fixed-length policy. We observed that:

- Our proposed *MU-ST* trained with MuST-C achieves higher BLEU at all latency regimes than other approaches. In particular, it even superior to the cascade method *ON-TRAC**,

[4] https://dl.fbaipublicfiles.com/fairseq/wav2vec/wav2vec_vox_960h_pl.pt

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

[6] The architecture of *MU-ST(+pretrain)** is determined by its inherited pre-training models. *MU-ST* adopts smaller models because of the low-resource training data。

[7] Following previous studies, we evaluate the computation-aware AL and computation-unaware AL on MuST-C dev set and test set, respectively.

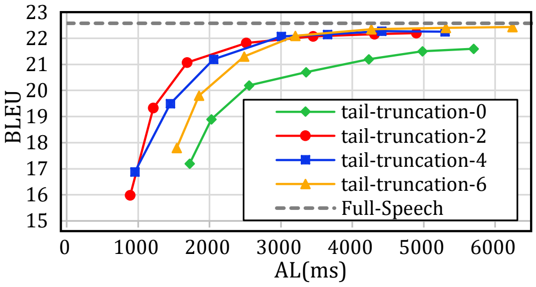


Figure 5: Impact of the number of truncated words in *tail-truncation*.

which utilized large-scale ASR and MT corpora for training.

- *MU-ST*(*+pretrain*)* outperforms *MU-ST* in BLEU by taking advantage of pre-trained models and more training data for $\mathcal{M}_{\mathcal{ST}}$. The full-speech translation of *MU-ST*(*+pretrain*)* has 3.32 BLEU points improvement (22.58→25.90) over *MU-ST*. Meanwhile, the latency of *MU-ST*(*+pretrain*)* is longer than *MU-ST* under the same δ. This may be because the pre-trained large speech translation model of *MU-ST*(*+pretrain*)* enables the high translation diversity of its speech translation model $\mathcal{M}_{\mathcal{ST}}$, which is not conducive in constructing fine-grained meaningful units. Strong translation diversity will reduce the probability of prefix matching between partial translation y'_i and full-speech translation $\widetilde{y}$ in MU construction, thus bringing in longer MUs and higher latency.

4.4 Ablation Studies

We conduct experiments concerning various aspects of our MU-based policy in this section. All ablation results are trained on the Limited-training-corpora setting and evaluated on MuST-C tst-COMMON set.

4.4.1 Do we need *tail-truncation*?

When constructing the data for MU detection, we proposed a *tail-truncation* strategy, which removes the last k words from the translation of each speech segment to avoid translation errors caused by ambiguous speech segments. Now we verify its significance. We compare models with different numbers of truncated words k in *tail-truncation*, with results shown in Figure 5. It is observed that without *tail-truncation* ($k = 0$), the translation quality is worse and the latency is longer, compared with $k = 2$. This corroborates our motivation specified in Section 3.2 that *tail-truncation* enables the model to discover fine-grained meaningful units. Moreover, it also facilitates producing context-aware translation by taking longer context in translation. Therefore, *tail-truncation* strategy plays an important role in extracting meaningful units.

Increasing k from 2 to 4 and 6 generally brings higher latency, along with a tiny improvement in translation quality. According to the I-MOS ranking mechanism (Zhang et al., 2021b) for ST systems, $k = 2$ and $k = 6$ ranks tied, both better than $k = 4$. $k = 2$ is superior at low-latency regime and $k = 6$ performs better at high-latency regime. This is because according to our MU extraction algorithm, we can always guarantee the consistency between the MU translation and the full-speech translation, regardless of the value of k. Larger k makes it easier to match partial translation and full-speech translation, thus producing more fine-grained MUs. But at the same time, truncating more translated words can avoid displaying problematic translations at the tail. So the translation quality of large k will not degrade. On the contrary, using a larger k improves the translation accuracy because it receives more source speech when performing translation.

4.4.2 Multi-modal vs. Single-modal MU Detection

To further study the effect of the translation history for MU detection, we remove the previously generated translation y_p from the multi-modal MU detection model. Without y_p, the segmentation model detects MUs only based on the input speech clip $s_{\leq t}$, and only optimizes the top 6-layer Transformer.

We build golden segmentation on tst-COMMON based on the meaningful speech units construction algorithm (Section 3.2), then evaluate different models on MU segmentation, translation quality, and latency. The results are shown in Table 2. It is observed that for both limited and open training corpora settings, the multi-modal method which combines speech features and translation history outperforms the single-modal method on MU segmentation in terms of F1 score (absolute improvements of 1.6-2 percentage points). However, there are only slight improvements in translation quality and a slight delay in latency. This is because incorrect segmentation does not necessarily lead to the decline of BLEU, which also depends on

Model	**setting**	**F1** (%)	**BLEU**	**AL** (ms)
MU-ST	Single-Modal	72.6	20.92	1642.2
	Multi-Modal	**74.6**	**21.07**	1684.8
MU-ST (*+pretrain*)*	Single-Modal	72.5	22.73	1925.7
	Multi-Modal	**74.1**	**22.78**	1952.5

Table 2: The performance of single-modal and multi-modal MU detection models evaluated on MuST-C tst-COMMON at $\delta = 0.5$.

the robustness of the speech translation model. For some small errors brought by wrong segmentation, a robust speech translation model may ignore them and generate correct translation when translating subsequent MUs. In such cases, the overall BLEU will not be largely affected.

4.5 Experiments on Zh-En ST

We also evaluate our method on Zh-En ST using BSTC (Zhang et al., 2021a) dataset. BSTC is the largest Zh-En public speech translation corpus, but contains only 66 hours of speech, corresponding to 37k sentences. To alleviate the data scarcity, we first construct pseudo speech translation data by translating the transcript of ASR corpora (AISHELL-1 (Bu et al., 2017), AISHELL-3 (Shi et al., 2020), and aidatatang_200zh[8]) with a Zh-En machine translation model trained on a translation corpus, CCMT2019 (Yang et al., 2019). Then the pseudo speech translation data, together with the BSTC, are assigned as the training corpus for the Zh-En end-to-end speech translation model. The combined training set contains a total of 529 hours of speech, corresponding to 478k sentence pairs of transcript and translation.

We implement three methods for comparison:

- ***Cascade***: we use an adaptive policy (Zhang et al., 2020) to connect an ASR model and an MT model. The ASR model is trained on 529 hours of speech, and the MT model based on Transformer big is pre-trained on CCMT2019 and fine-tuned on BSTC. The adaptive policy based on textual MU (Zhang et al., 2020) is trained on BSTC.

- ***Cascade****: Similar to *Cascade*, the only difference is that it adopts a public real-time ASR API[9] that uses more than 9400 hours of ASR training data (Amodei et al., 2016).

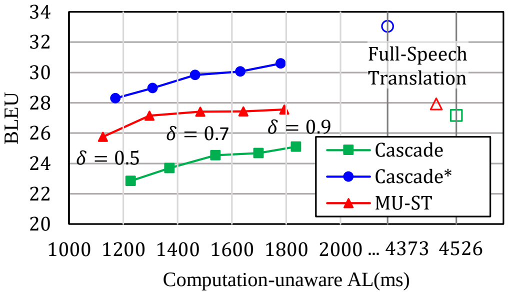


Figure 6: The performance on the Zh-En BSTC testset.

- ***MU-ST***: Our proposed adaptive speech segmentation policy for end-to-end ST. The acoustic encoders and target decoders of our speech translation model are initialized by the ASR and MT model of *Cascade* method, respectively. Then we fine-tune with speech translation data following Li et al. (2020b).

The results in Figure 6 show that: **1)** *Cascade** has a significant advantage over the other two methods. This is because the word error rate of the ASR model is 10.32% and 21.58% for *Cascade** and *Cascade*, respectively, leading to 5.9 BLEU points of gap between their full-speech translation results (27.2 vs. 33.1)[10]. **2)** *Cascade* and our end-to-end method *MU-ST* are optimized with identical training data, but *MU-ST* outperforms *Cascade*. We attribute this improvement to two reasons. First, the end-to-end method avoids ASR error propagation, with the full-speech translation of *MU-ST* surpassing *Cascade* by 0.7 BLEU points (27.9 vs. 27.2). Second and more important, *MU-ST* detects MUs directly from speech, thus avoiding loss of information. The average gap between *Cascade* and *MU-ST* at five ST results is 2.9 BLEU points, much larger than that of full-speech translation (0.7). This represents that segmentation from the source speech is superior to segmentation from noisy ASR results. Accordingly, we expect our *MU-ST* to have greater potential based on large-scale training data.

5 Conclusion

We present an adaptive speech segmentation policy for end-to-end simultaneous translation, which triggers translation with a meaningful speech unit de-

[8] a free Chinese Mandarin speech corpus by Beijing DataTang Technology Co., Ltd (www.datatang.com)

[9] https://ai.baidu.com/tech/speech/realtime_asr

[10] Note that, our proposed *MU-ST* surpassed the cascade method *ON-TRAC** in En-De experiments, but it failed to surpass *Cascade** in Zh-En because the ASR training data of *ON-TRAC** in En-De is only three times that of *MU-ST* (in hours), while the training data of *Cascade** is thousands of times that of *MU-ST* in Zh-En experiments.

tector. Experiments across two language pairs show that our method outperforms state-of-the-art methods with constrained training corpus, suggesting the effectiveness of our adaptive policy. Ablation studies reveal key factors that lead to its success, including tail-truncation, multi-modal segmentation, and speech-text pre-training.

References

Dario Amodei, Sundaram Ananthanarayanan, Rishita Anubhai, Jingliang Bai, Eric Battenberg, Carl Case, Jared Casper, Bryan Catanzaro, Qiang Cheng, Guoliang Chen, et al. 2016. Deep speech 2: End-to-end speech recognition in english and mandarin. In *International conference on machine learning*, pages 173–182. PMLR.

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondřej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, et al. 2020. Findings of the iwslt 2020 evaluation campaign. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 1–34.

Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. 2019. Monotonic infinite lookback attention for simultaneous machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1313–1323, Florence, Italy. Association for Computational Linguistics.

Alexei Baevski, Michael Auli, and Abdelrahman Mohamed. 2019. Effectiveness of self-supervised pretraining for speech recognition. *arXiv preprint arXiv:1911.03912*.

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

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018. Pretraining on high-resource speech recognition improves low-resource speech-to-text translation.

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *NIPS workshop on End-to-end Learning for Speech and Audio Processing*.

Hui Bu, Jiayu Du, Xingyu Na, Bengu Wu, and Hao Zheng. 2017. Aishell-1: An open-source mandarin speech corpus and a speech recognition baseline. In *2017 20th Conference of the Oriental Chapter of the International Coordinating Committee on Speech Databases and Speech I/O Systems and Assessment (O-COCOSDA)*, pages 1–5. IEEE.

Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. 2021. Direct simultaneous speech-to-text translation assisted by synchronized streaming asr.

Kyunghyun Cho and Masha Esipova. 2016. Can neural machine translation do simultaneous translation? *arXiv preprint arXiv:1606.02012*.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 2018. Incremental decoding and training methods for simultaneous translation in neural machine translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 493–499, New Orleans, Louisiana. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding. 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 4171–4186.

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019a. 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)*, pages 2012–2017, Minneapolis, Minnesota. Association for Computational Linguistics.

Mattia Antonino Di Gangi, Matteo Negri, Roldano Cattoni, Dessi Roberto, and Marco Turchi. 2019b. Enhancing transformer for end-to-end speech-to-text translation. In *Machine Translation Summit XVII*, pages 21–31. European Association for Machine Translation.

Maha Elbayad, Ha Nguyen, Fethi Bougares, Natalia Tomashenko, Antoine Caubrière, Benjamin Lecouteux, Yannick Estève, and Laurent Besacier. 2020. On-trac consortium for end-to-end and simultaneous speech translation challenge tasks at iwslt 2020. *arXiv preprint arXiv:2005.11861*.

Marcello Federico, Alex Waibel, Kevin Knight, Satoshi Nakamura, Hermann Ney, Jan Niehues, Sebastian Stüker, Dekai Wu, Joseph Mariani, and François Yvon. 2020. Proceedings of the 17th international conference on spoken language translation. In *Proceedings of the 17th International Conference on Spoken Language Translation*.

Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK Li. 2017. Learning to translate in real-time with neural machine translation. In *Proceedings of* *the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 1, Long Papers*, pages 1053–1062.

Dan Hendrycks and Kevin Gimpel. 2016. Gaussian error linear units (gelus). *arXiv preprint arXiv:1606.08415*.

Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 2019. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7180–7184. IEEE.

Bo Li, Shuo-yiin Chang, Tara N Sainath, Ruoming Pang, Yanzhang He, Trevor Strohman, and Yonghui Wu. 2020a. Towards fast and accurate streaming end-to-end asr. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6069–6073. IEEE.

Xian Li, Changhan Wang, Yun Tang, Chau Tran, Yuqing Tang, Juan Pino, Alexei Baevski, Alexis Conneau, and Michael Auli. 2020b. Multilingual speech translation with efficient finetuning of pretrained models. *arXiv preprint arXiv:2010.12829*.

Mingbo Ma, Liang Huang, Hao Xiong, Kaibo Liu, Chuanqiang Zhang, Zhongjun He, Hairong Liu, Xing Li, and Haifeng Wang. 2019. STACL: simultaneous translation with integrated anticipation and controllable latency. In *ACL 2019*, volume abs/1810.08398.

Xutai Ma, Juan Pino, James Cross, Liezl Puzon, and Jiatao Gu. 2020a. Monotonic multihead attention. In *ICLR 2020*.

Xutai Ma, Juan Pino, and Philipp Koehn. 2020b. Simulmt to simulst: Adapting simultaneous text translation to end-to-end simultaneous speech translation. *arXiv preprint arXiv:2011.02048*.

Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, Philipp Koehn, and Juan Pino. 2021. Streaming simultaneous speech translation with augmented memory transformer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7523–7527. IEEE.

Abdelrahman Mohamed, Dmytro Okhonko, and Luke Zettlemoyer. 2019. Transformers with convolutional context for asr. *arXiv preprint arXiv:1904.11660*.

Niko Moritz, Takaaki Hori, and Jonathan Le. 2020. Streaming automatic speech recognition with the transformer model. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6074–6078. IEEE.

Ha Nguyen, Yannick Estève, and Laurent Besacier. 2021. An empirical study of end-to-end simultaneous speech translation decoding strategies. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7528–7532. IEEE.

Yusuke Oda, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. 2014. Optimizing segmentation strategies for simultaneous speech translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, volume 2, pages 551–556.

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

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.

Yao Shi, Hui Bu, Xin Xu, Shaoji Zhang, and Ming Li. 2020. Aishell-3: A multi-speaker mandarin tts corpus and the baselines. *arXiv preprint arXiv:2010.11567*.

Matthias Sperber, Jan Niehues, Graham Neubig, Sebastian Stüker, and Alex Waibel. 2018. Self-attentional acoustic models. *arXiv preprint arXiv:1803.09519*.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

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

Chengyi Wang, Yu Wu, Shujie Liu, Jinyu Li, Liang Lu, Guoli Ye, and Ming Zhou. 2020b. Low latency end-to-end streaming speech recognition with a scout network. *arXiv preprint arXiv:2003.10369*.

Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. 2017. Sequence-to-sequence models can directly translate foreign speech. *arXiv preprint arXiv:1703.08581*.

Patrick Wilken, Tamer Alkhouli, Evgeny Matusov, and Pavel Golik. 2020. Neural simultaneous speech translation using alignment-based chunking. In *Proceedings of the 17th International Conference on Spoken Language Translation*, pages 237–246, Online. Association for Computational Linguistics.

Muyun Yang, Xixin Hu, Hao Xiong, Jiayi Wang, Yiliyaer Jiaermuhamaiti, Zhongjun He, Weihua Luo, and Shujian Huang. 2019. Ccmt 2019 machine translation evaluation report. In *China Conference on Machine Translation*, pages 105–128. Springer.

Xingshan Zeng, Liangyou Li, and Qun Liu. 2021. Realtrans: End-to-end simultaneous speech translation with convolutional weighted-shrinking transformer. *arXiv preprint arXiv:2106.04833*.

Ruiqing Zhang, Xiyang Wang, Chuanqiang Zhang, Zhongjun He, Hua Wu, Zhi Li, Haifeng Wang, Ying Chen, and Qinfei Li. 2021a. BSTC: A large-scale Chinese-English speech translation dataset. In *Proceedings of the Second Workshop on Automatic Simultaneous Translation*, pages 28–35, Online. Association for Computational Linguistics.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2020. Learning adaptive segmentation policy for simultaneous translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2280–2289.

Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2021b. Findings of the second workshop on automatic simultaneous translation. In *Proceedings of the Second Workshop on Automatic Simultaneous Translation*, pages 36–44, Online. Association for Computational Linguistics.

A Appendix

A.1 Experimental Corpora

We list the corpora used by different methods of En-De in Table 3.

	Type	Corpus	#Sents / #Hours	Limited-Training-Corpora	Open-Training-Corpora		
					MU-ST(+*pretrain*)	*ON-TRAC*	*Wait-K-Stride-S -Write-N*
Train							
	ST	MuST-C	229k(408h)	✓	✓	✓	✓
		Covost2	290k(430h)		✓		
		Europarl-ST	32k(77h)		✓	✓	✓
	ASR	How2	365h			✓	✓
		TED-LIUM3	452h			✓	
	MT	CommonCrawl + Europarl + News Commentary	1543k + 1730k + 320k			✓	
Dev & Test							
	Dev	MuST-C dev	1423(2.5h)	✓	✓	✓	✓
	Test	MuST-C tst-COMMON	2641(4.0h)	✓	✓	✓	✓

Table 3: The statistics of the corpora used in the simultaneous translation experiments. We list four data settings, one belongs to the Limited-training-corpora setting, and the other three belong to the Open-training-corpora setting.

The statistics of the training data used in Zh-En experiments is listed in Table 4.

Type	Corpus	#Hours	#Sent Pairs
ASR	AISHELL-1	178	141k
	AISHELL-3	85	63k
	aidatatang	200	237k
ST	BSTC	66	37k
MT	CCMT2019	/	9.1M

Table 4: The audio duration and the number of sentences of the corpora used in Zh-En ST experiments. The corpora with gray background are used as the final speech translation datasets, in which transcripts of the ASR corpora are translated by an NMT model to construct pseudo speech translation.

A.2 Numeric Results for the figures

δ	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Full-Speech
AL	1049	1336	1674	2099	2819	4144	5377	5673
BLEU	16.54	19.14	20.92	21.79	22.07	22.00	22.14	22.04

Table 5: Numeric Results of *MU-ST* for Figure 4(a).

A.3 Case Study

We showcase an example in Zh-En from the test sets in Figure 7.

δ	0.3	0.4	0.5	0.6	0.7	0.8	0.9	Full-Speech
MU-ST								
AL	888	1211	1685	2516	3452	4310	4896	5682
BLEU	15.98	19.33	21.07	21.82	22.08	22.16	22.2	22.58
*MU-ST(+pretrain)**								
AL	1023	1424	1953	2642	3621	4453	5089	5754
BLEU	17.94	20.85	22.78	24.3	24.82	24.99	25.05	25.9

Table 6: Numeric Results of *MU-ST* and *MU-ST(+pretrain)** for Figure 4(b).

δ	0.5	0.6	0.7	0.8	0.9	Full-Speech
Cascade						
AL	1228	1370	1539	1698	1836	2426
BLEU	22.84	23.69	24.54	24.68	25.1	27.17
*Cascade**						
AL	1171	1308	1464	1631	1780	2173
BLEU	28.30	28.97	29.84	30.06	30.60	33.05
*MU-ST**						
AL	1125	1296	1484	1642	1793	2352
BLEU	25.76	27.15	27.42	27.43	27.56	27.93

Table 7: Numeric Results for Figure 6.

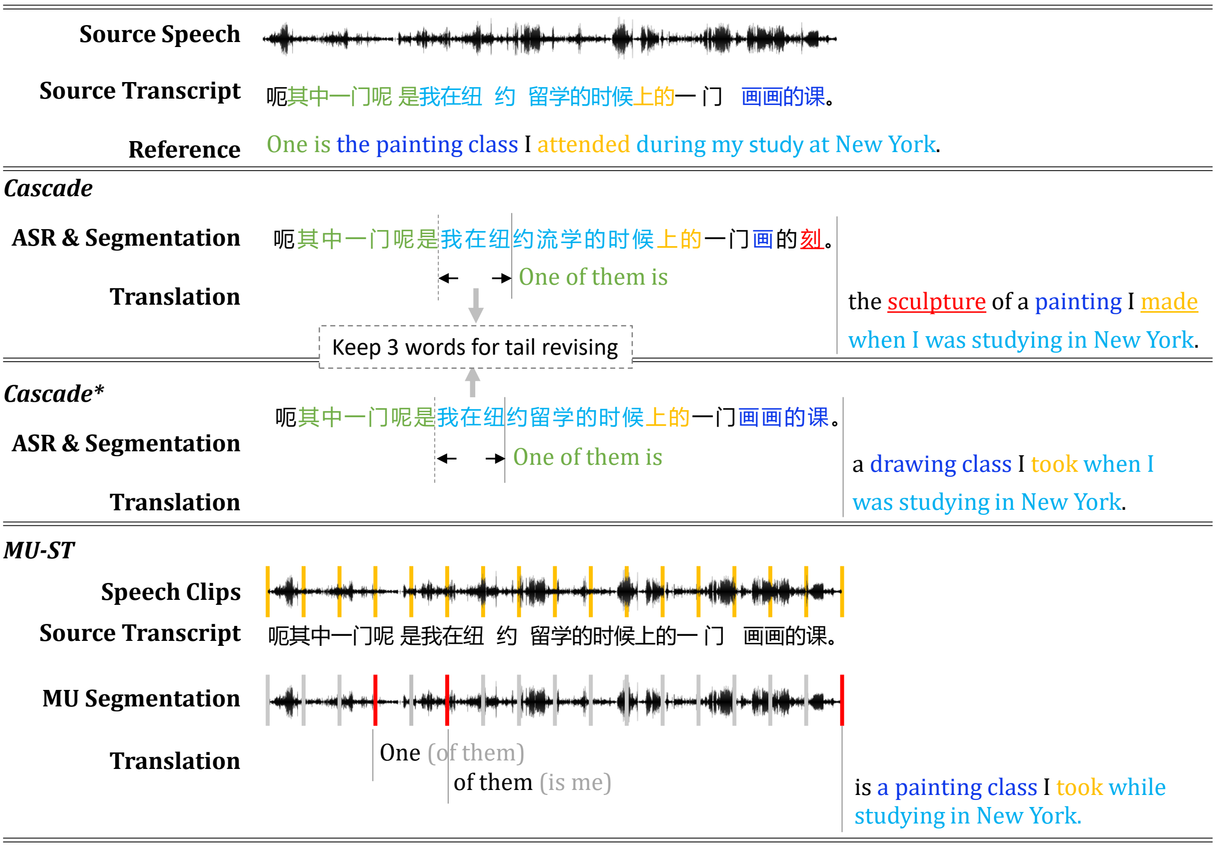

Source Speech
Source Transcript
呃其中一门呢 是我在纽 约 留学的时候上的一 门 画画的课。
Reference
One is the painting class I attended during my study at New York.
Cascade
ASR & Segmentation
呃其中一门呢是我在纽约流学的时候上的一门画的刻。
Translation
One of them is
the sculpture of a painting I made when I was studying in New York.
Keep 3 words for tail revising
Cascade*
ASR & Segmentation
呃其中一门呢是我在纽约留学的时候上的一门画画的课。
Translation
One of them is
a drawing class I took when I was studying in New York.
MU-ST
Speech Clips
Source Transcript
呃其中一门呢 是我在纽 约 留学的时候上的一 门 画画的课。
MU Segmentation
Translation
One (of them)
of them (is me)
is a painting class I took while studying in New York.

Figure 7: A Chinese-English example in the BSTC test set. In the ASR result of *Cascade*, two characters are incorrectly recognized: "留学"("study abroad")→"流学" ("rheological") and "画画的课" ("painting class") →"画画的刻" ("sculpture of painting"), which makes the translation distort the meaning of the source speech, see the underlined translation. On the contrary, our proposed *MU-ST* avoids the error propagated from ASR by end-to-end training and generates a correct translation. Moreover, both the cascade methods keep revising some tail words for better accuracy, but causing translation delay, denoted by the 3-words lagging. *MU-ST* remove this extra delay through end-to-end speech translation.