AN ABSTRACT OF THE DISSERTATION OF

Junkun Chen for the degree of Doctor of Philosophy inComputer Science and Computer Science presented on December 6, 2022.

Title: Towards Direct Simultaneous Speech Translation

Abstract approved: ______________________________

Liang Huang Prasad Tadepalli

Simultaneous speech translation (SimulST) is widely useful in many cross-lingual communication scenarios, including multinational conferences and international traveling. Since text-based simultaneous machine translation (SimulMT) has achieved great success in recent years. The conventional cascaded approach for SimulST uses a pipeline of streaming ASR followed by simultaneous MT but suffers from error propagation and extra latency. Recent efforts attempt to directly translate the source speech into the target text or speech simultaneously, but this is much harder due to the combination of separate tasks. In this dissertation, we focus on improving simultaneous translation model, enabling it to handle speech input and directly generate the translated text in the target language. First, we investigate how to improve simultaneous translation by incorporating generated more monotonic pseudo references in training. These pseudo references with fewer reorderings cause fewer anticipations and can substantially improve simultaneous translation quality. Then, we propose an ASR-assisted direct SimulST framework. The model can directly translate from the given speech with a wait-k policy guided by a synchronized streaming ASR. However, speech translation tasks suffer from data scarcity problems. To alleviate the issue, we next introduce a Fused Acoustic and Text Masked Language Model (FAT-MLM), which jointly learns a unified representation for both acoustic and text input from various types of corpora, including parallel data for speech recognition and machine translation, and even pure speech and text data. By finetuning from FAT, the speech translation model can be substantially improved. Besides that, we further extend FAT to cross-lingual speech synthesis. Our proposed model can clone the voice of the source speaker and generate the corresponding speech in the target language.

Towards Direct Simultaneous Speech Translation

by

Junkun Chen

A DISSERTATION

submitted to

Oregon State University

in partial fulfillment of
the requirements for the
degree of

Doctor of Philosophy

Presented December 6, 2022
Commencement June 2023

Doctor of Philosophy dissertation of Junkun Chen presented on December 6, 2022.

APPROVED:

Co-Major Professor, representing Computer Science

Co-Major Professor, representing Computer Science

Head of the School of Electrical Engineering and Computer Science

Dean of the Graduate School

I understand that my dissertation will become part of the permanent collection of Oregon State University libraries. My signature below authorizes release of my dissertation to any reader upon request.

Junkun Chen, Author

ACKNOWLEDGEMENTS

I would first like to express my deepest of appreciation for my advisor, Professor Liang Huang. Throughout my time at OSU, he offered unwavering support, dedication, and encouragement. He has been a wealth of knowledge, helping me through the darkest hours along the way.

I would like to acknowledge my safe-harboring co-advisor, Prof. Prasad Tadepalli. He accepted to be my co-advisor, enabling me to complete my research program as I wished. I would like to extend my gratitude to the rest of my thesis committee: Prof. Stefan Lee, Prof. Xiaoli Fern, and my Graduate Council Representative Prof. Leonard Coop, for their suggestion, questions and advice throughout my Ph.D. exams. Thank you all for serving on my committee and guiding my research.

Life would have been hard without the support from others. I would like to thank my labmates for all the fun and great collaborations we have had. They are Renjie Zheng, Juneki Hong, Göksu Öztürk Miraç, Atsuhito Kita, Ning Dai, Liang Zhang, He Zhang, and Sizhen Li. I am also grateful to my co-workers during my internships. They are Mingbo Ma, Xintong Li, Kaibo Liu, He Bai, Hui Zhang, Tian Yuan, Enlei Gong, Xiaoran Fan and Zeyu Chen.

The journey from a kid in a small town in southwest China to a Ph.D. in the United States was a long way, and I couldn't make it without the support of my family. I would like to thank my parents and my fiancée for their unfailing love and encouragement.

Parts of the research in this thesis were supported in part by National Science Foundation (NSF) Grants IIS-1817231 and IIS-2009071.

TABLE OF CONTENTS

TABLE OF CONTENTS (Continued)

Page

TABLE OF CONTENTS (Continued)

LIST OF FIGURES

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF FIGURES (Continued)

LIST OF TABLES

LIST OF TABLES (Continued)

LIST OF ALGORITHMS

Chapter 1: Introduction

Simultaneous translation incrementally translates source-language speech into speech or text in target-language, and is widely useful in many cross-lingual communication scenarios such as international travels and multinational conferences. Recently, text-to-text simultaneous machine translation has witnessed great progress thanks to fixed-latency policies (such as wait-k) [62] and adaptive policies [39, 4]. A more nature way is to translate directly from the source speech to maximize the use of input information. However, these methods cannot be directly applied to models that take speech as the input.

1.1 Simultaneous Speech translation

Neural machine translation (NMT) has received much attention in recent years. Sequence-to-Sequence (seq2seq) models based on Recurrent Neural Network (RNN) [8] and Transformer [88] achieves significant performance in this task. We briefly review full-sentence machine translation and the wait-k policy in simultaneous translation.

Full-Sentence NMT uses a Seq2seq framework (Fig. 1.1) where the encoder processes the source sentence $\mathbf{x} = (x_1, x_2, ..., x_m)$ into a sequence of hidden states. A decoder sequentially generates a target sentence $\mathbf{y} = (y_1, y_2, ..., y_n)$ conditioned on those hidden states and previous predictions:

$$\hat{\mathbf{y}} = \mathop{\mathbf{argmax}}_{\mathbf{y}} p_{\text{full}}(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}_{\text{full}}) \tag{1.1}$$

$$p_{\text{full}}(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}) = \prod_{t=1}^{|\mathbf{y}|} p(y_t \mid \mathbf{x}, \mathbf{y}_{<t}; \boldsymbol{\theta}) \tag{1.2}$$

The model is trained as follows:

$$\boldsymbol{\theta}_{\text{full}} = \mathop{\mathbf{argmax}}_{\boldsymbol{\theta}} \prod_{(\mathbf{x}, \mathbf{y}^*) \in D} p_{\text{full}}(\mathbf{y}^* \mid \mathbf{x}; \boldsymbol{\theta}) \tag{1.3}$$

Simultaneous Translation translates concurrently with the (growing) source sentence, so Ma

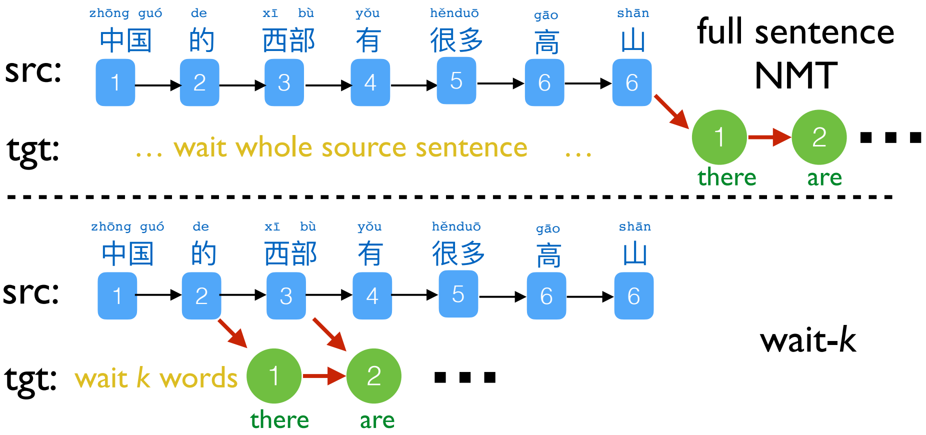


Figure 1.1: Full-sentence vs. simultaneous (wait-k) MT.

et al. [62] propose the wait-k policy (Fig. 1.1) following a simple, fixed schedule that commits one target word on receiving each new source word, after an initial wait of k source words. In this example, the model uses a wait-2 policy, it starts translating *"there"* after receiving first two source words "中国 的". Then when it receives the next source word "西部", it generates the new translated word *"are"*. Formally, the prediction of $\mathbf{y}$ for a trained wait-k model is

$$p_{\text{wait-}k}(\mathbf{y} \,|\, \mathbf{x}; \boldsymbol{\theta}) = \prod_{t=1}^{|\mathbf{y}|} p(y_t \,|\, \mathbf{x}_{<t+k}, \mathbf{y}_{<t}; \boldsymbol{\theta}) \tag{1.4}$$

where the wait-k model is trained as follows

$$\boldsymbol{\theta}^{\text{wait-}k} = \mathop{\mathbf{argmax}}_{\boldsymbol{\theta}} \prod_{(\mathbf{x},\mathbf{y}^*)\in D} p_{\text{wait-}k}(\mathbf{y}^* \mid \mathbf{x}; \boldsymbol{\theta}). \tag{1.5}$$

In general, we expect the model can translate the give input into the target language in a monotonic fashion.

For speech processing like automatic speech recognition (ASR) and speech translation (ST). They differ from the above formulae in that the input becomes processed speech features $\mathbf{s} = (s_1, ..., s_{|\mathbf{s}|})$.

A typical instance of speech translation parallel data can be described as a triplet. 1. A continuous speech waveform $\mathbf{s}$, and it can be described as spectrogram or mel-spectrogram of the source speech, like $\mathbf{s} = (s_1, ..., s_{|\mathbf{s}|})$. 2. The text transcript $\mathbf{x}$ in the source language. 3. A corresponding text translation $\mathbf{y}$ in the target language.

End-to-End Speech Translation (E2E-ST) aims to directly translate the source speech $\mathbf{s}$ into the text translation in the target language.

$$\hat{\mathbf{y}} = \mathop{\mathbf{argmax}}_{\mathbf{y}} p_{\text{full}}(\mathbf{y} \mid \mathbf{s}; \boldsymbol{\theta}_{\text{full}}) \tag{1.6}$$

$$p_{\text{full}}(\mathbf{y} \mid \mathbf{s}; \boldsymbol{\theta}) = \prod_{t=1}^{|\mathbf{y}|} p(y_t \mid \mathbf{s}, \mathbf{y}_{<t}; \boldsymbol{\theta}) \tag{1.7}$$

It is also known as direct ST. The training remains the same as the text-based NMT model. Besides that, to facilitate closing the speech-text modality discrepancy and improve the performance, a multi-task learning framework, which jointly train ASR and ST with shared parameters.

1.2 Challenges in Simultaneous Speech Translation

The conventional approach to this problem is a cascaded one [6, 96, 108], involving a pipeline of three steps. First, the streaming automatic speech recognition (ASR) module transcribes the input speech on the fly [68, 90], and then a simultaneous text-to-text translation module translates the partial transcription into target-language text [70, 28, 62, 104, 102, 105, 5]. Finally, an incremental TTS system is used to generate corresponding audio wavs.

However, the cascaded approach inevitably suffers from three limitations: (a) **error propagation**, where the errors generated in each step will be passed on to the next step, creating an accumulation of errors. For example, streaming ASR's mistakes confuse the translation module (which are trained on clean text), and this problem worsens with noisy environments and accented speech; and (b) **extra latency**, where Each module needs to wait for the module in the previous step to finish processing. (c) **information loss**, where the prosody, durations, and other information in the source speech can help improve the quality of translated text and speech, but they are discarded in the speech recognition step. (d) There exist many non-orthographic language that cannot be transcribed into text. To overcome the above issues, some recent efforts [77, 65, 64] attempt to directly translate the source speech into target text simultaneously by adapting text-based wait-k strategy [62]. However, unlike simultaneous translation whose input is already segmented into words or subwords, in speech translation, the key challenge is to figure out the number of valid tokens within a given source speech segment in order to apply the wait-k

policy.

Besides that, the parallel corpora for simultaneous translation are limited, and even less for speech input. It is challenge to obtain high-quality translations. Therefore, both simultaneous translation and speech translation suffer from the problem of data scarcity.

1.3 Our Proposed Methods

In this dissertation, three topics that related to simultaneous speech translation are introduced. First, we will introduce a simple and effective technique to generate pseudo-references with fewer reordering based on parallel corpora that designed for full-sentence translation [23]. Training simultaneous translation models with these generated pseudo references can reduce anticipations during training and result in fewer hallucinations in decoding and lower latency (Chapter 2). It can effectively enhance the monotonicity of the translation model without the use of additional data. Next, we will introduce a ASR-assisted direct simultaneous speech translation framework [21] (Chapter 3). It bridges the gap between text-based translation and ST, we can easily adopt the wait-k methods which were designed for text-based input to address speech input. Then, to alleviate the problem of data scarcity in the parallel speech translation corpora, we also propose using multi-modal pretraining method to learn a unified acoustic and text representations [109, 22] (Chapter 4). Besides that, we extend the multi-modal pretraining method to speech synthesis. The proposed model can generally conduct voice cloning between different languages (Chapter 5).

Chapter 2: Improving Simultaneous Translation with Pseudo-References

2.1 Motivation

Recently, all state-of-the-art simultaneous translation models are trained on conventional parallel text which involve many unnecessary long-distance reorderings [13, 16]; see Fig. 2.1 for an example. The simultaneous translation models trained using these parallel sentences will learn to either make bold hallucinations (for fixed-latency policies) or introduce long delays (for adaptive ones). Alternatively, one may want to use transcribed corpora from professional simultaneous interpretation [66, 10, 69]. These data are more monotonic in word-order, but they are all very small in size due to the high cost of data collection (e.g., the NAIST one [69] has only $387k$ target words). More importantly, simultaneous interpreters tend to summarize and inevitably make many mistakes [81, 96, 108] due to the high cognitive load and intense time pressure during interpretation [19].

How can we combine the merits of both types of data, and obtain a large-scale, more monotonic parallel corpora for simultaneous translation? We propose a simple and effective technique to generate pseudo-references with fewer reorderings; see the "Pseudo-Refs" in Fig. 2.1. While previous work [42] addresses this problem via language-specific hand-written rules, our technique can be easily adopted to any language pairs without using extra data or expert linguistic knowledge. Training with these generated pseudo references can reduce anticipations during training and result in fewer hallucinations in decoding and lower latency.

2.2 Methods

Since the wait-k models are trained on conventional full-sentence bitexts, their performance is hurt by unnecessary long-distance reorderings between the source and target sentences. For example, the training sentence pair in Fig. 1.1, a wait-2 model learns to output y_1=*"there"* after observing $x_1 x_2$="中国 的" (*china 's*) which seems to induce a good anticipation ("中国 的..." ↔ "*There* ..."), but it could be a wrong hallucination in many other contexts (e.g., "中国 的 街道 很 挤" ↔ "*Chinese streets are crowded*", not "*There* ..."). Even for adaptive policies

Source Input	*zhōngguó de* *xībù* *yǒu* *hǔnduō* *gāo* *shān* 中国 的 西部 有 很多 高 山 *china ’s* ***west*** ***have*** ***many*** ***big*** ***mountain***
Gold-Ref	**there** **are** **many** **big** **mountains** in **western** china
Pseudo-Refs	*(wait-1)* china ’s **west** **has** **many** **big** **mountains** *(...wait-2...)* the chinese **west** **has** **many** **big** **mountains** *(...wait-3...)* **western** china **has** **many** **big** **mountains** *(...wait-4...)* there are **many** **big** ...

Figure 2.1: Example of unnecessary reorderings in the bitext which can force the model to anticipate aggressively, along with the ideal pseudo-references with different wait-k policies. Larger k improves fluency but sacrifices latency (pseudo-refs with $k \geq 4$ are identical to the original reference).

[39, 4, 103], the model only learns a higher latency policy (wait till x_4=“有”) by training on the example in Fig. 1.1. As a result, training-time wait-k models tend to do wild hallucinations [62].

To solve this problem, we propose to generate pseudo-references which are *non-anticipatory* under a specific simultaneous translation policy by the method introduced in Section 2.2.1. Meanwhile, we also propose to use BLEU score to filter the generated pseudo-references to guarantee that they are *semantic preserving* in Section 2.2.2.

2.2.1 Generating Pseudo-References with Test-time Wait-k

To generate *non-anticipatory* pseudo-references under a wait-k policy, we propose to use the full-sentence NMT model $\boldsymbol{\theta}_{\text{full}}$ (Eq. 1.3) which is *not* trained to anticipate, but decode with a wait-k policy. This combination is called *test-time wait-k* [62], which is unlikely to hallucinate since the full source content is always available during training. Although here the full-sentence model $\boldsymbol{\theta}_{\text{full}}$ only has access to the partially available source words $\boldsymbol{x}_{<t+k}$, it can still enforce fluency because $\hat{y}_t$ relies on the decoded target-side prefix $\hat{\boldsymbol{y}}_{<t}$ (Eq. 1.4). Formally, the generation of pseudo-references is:

$$\tilde{\mathbf{y}}^* = \operatorname*{\mathbf{argmax}}_{\mathbf{y}} p_{\text{wait-}k}(\mathbf{y} \mid \mathbf{x}; \boldsymbol{\theta}_{\text{full}})$$

Fig. 2.1 shows the pseudo-references with different wait-k policies ($k = 1..4$). Note that

$k = 1$ or 2 results in non-idiomatic translations, and larger k leads to more fluent pseudo-references, which converge to the original reference with $k \geq 4$. The reason is that in each wait-k policy, each target word $\hat{y}_t$ only rely on observed source words ($\mathbf{x}_{<t+k}$).

To further improve the quality of the pseudo-references generated by test-time wait-k, we propose to select better pseudo-references by using beam search. Beam search usually improves translation quality but its application to simultaneous translation is non-trivial, where output words are committed on the fly [107]. However, for pseudo-reference generation, unlike simultaneous translation decoding, we can simply adopt conventional off-line beam search algorithm since the source sentence is completely known. A larger beam size will generally give better results, but make anticipations more likely to be retained if they are correct and reasonable. To trade-off the expectations of quality and monotonicity, we choose beam size $b = 5$ in this work.

2.2.2 Translation Quality of Pseudo-References

We can use sentence-level BLEU score to filter out low quality pseudo-references. Fig. 2.2 shows the sentence level BLEU distributions of the pseudo-references generated with different wait-k policies. As k increases, the translation qualities are better since more source prefixes can be observed during decoding. The obvious peak at the BLEU=100 on Zh→En denotes those pseudo-references which are identical to the original ones. Those original references are probably already non-hallucinatory or correspond to very short source sentences (e.g. shorter than k). The figure shows that even for wait-1 policy, around 40% pseudo-references can achieve BLEU score above 60.

2.3 Anticipation & Hallucination Metrics

2.3.1 Anticipation Rate of (Pseudo-)References

During the training of a simultaneous translation model, an anticipation happens when a target word is generated before the corresponding source word is encoded. To identify the anticipations, we need the word alignment between the parallel sentences.

A word alignment a between a source sentence $\mathbf{x}$ and a target sentence $\mathbf{y}$ is a set of source-target word index pairs (s, t) where the s^{th} source word x_s aligns with the t^{th} target word y_t. In

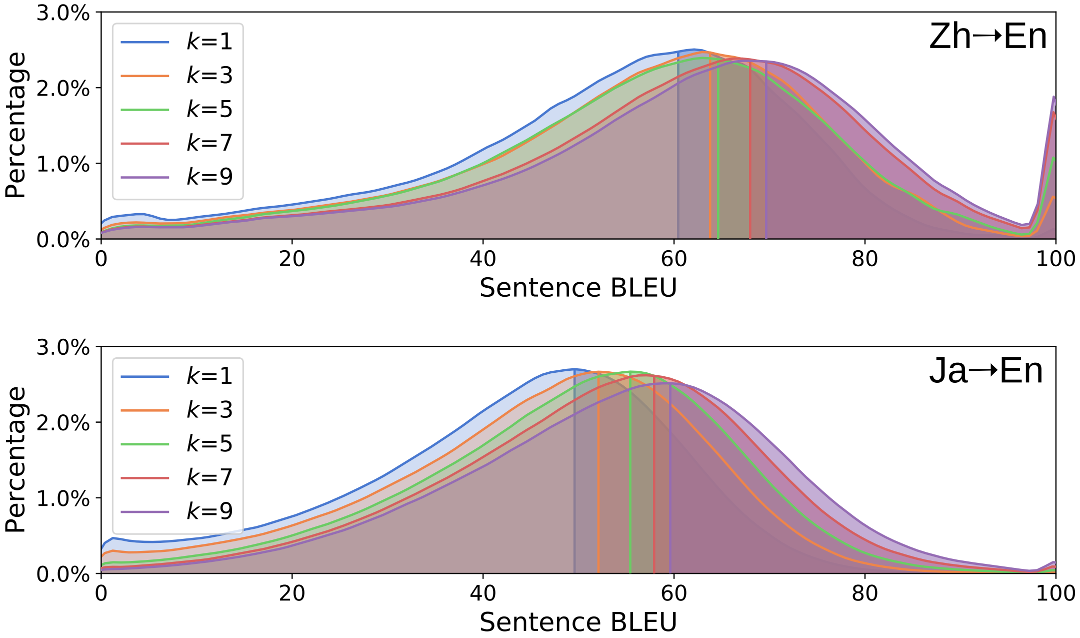


Figure 2.2: Sentence-level BLEU distributions of Pseudo-Refs using wait-k policies for Zh→En and Ja→En, respectively. The parts to the right of the vertical lines indicate the top 40% references in terms of BLEU in each distribution.

the example in Fig. 2.3, the word alignment is:

$$a = \{(1,8), (3,7), (4,1), (4,2), (5,3), (6,4), (7,5)\}$$

Based on the word alignment a, we propose a new metric called "k-anticipation" to detect the anticipations under wait-k policy. Formally, a target word y_t is k-anticipated ($A_k(t,a) = 1$) if it aligns to at least one source word $\mathbf{x}_s$ where $s \geq t + k$:

$$A_k(t,a) = \mathbb{1}[\{(s,t) \in a \,|\, s \geq t + k\} \neq \varnothing]$$

We further define the k-anticipation rate (AR_k) of an $(\mathbf{x}, \mathbf{y}, a)$ triple under wait-k policy to be:

$$AR_k(\mathbf{x}, \mathbf{y}, a) = \frac{1}{|\mathbf{y}|} \sum_{t=1}^{|\mathbf{y}|} A_k(t,a)$$

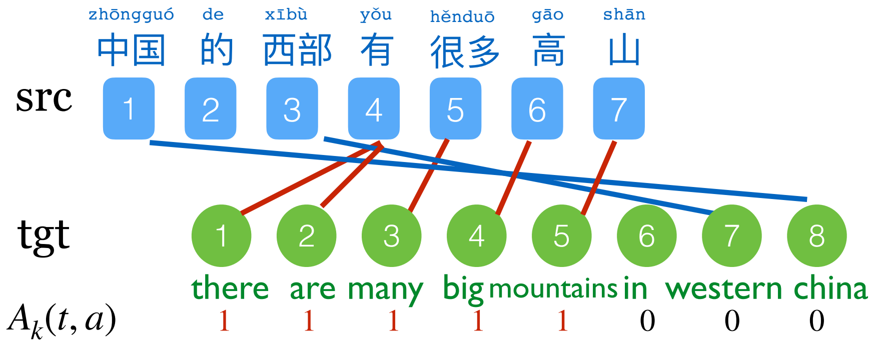


Figure 2.3: An example word alignment and the wait-1 policy. The red and blue lines indicate the 1-anticipated and non-anticipated alignments, resp. Here AR_1 = 5/8.

2.3.2 Hallucination Rate of Hypotheses

The goal of reducing the anticipation rate during the training of a simultaneous translation model is to avoid hallucination at testing time. Similar to the anticipation metric introduced in the previous section, we define another metric to quantify the number of hallucinations in decoding. A target word $\hat{y}_t$ is a *hallucination* if it can not be aligned to any source word. Formally, based on word alignment a, whether target word $\hat{y}_t$ is a hallucination is

$$H(t, a) = \mathbb{1}[\{(s, t) \in a\} = \varnothing]$$

We further define hallucination rate HR as

$$HR(\mathbf{x}, \hat{\mathbf{y}}, a) = \frac{1}{|\hat{\mathbf{y}}|} \sum_{t=1}^{|\hat{\mathbf{y}}|} H(t, a)$$

To avoid non-faithful contextual alignments, we use IBM Model 1 [17] for HR.

2.4 Expriments

2.4.1 Datasets and Models

We conduct the experiments on two language pairs Zh→En and Ja→En. We use NIST corpus (2M pairs) for Zh→En as training set, and NIST 2006 and NIST 2008 as dev and test set, which contains 616 and 691 sentences with 4 English references respectively. We also collected a set of

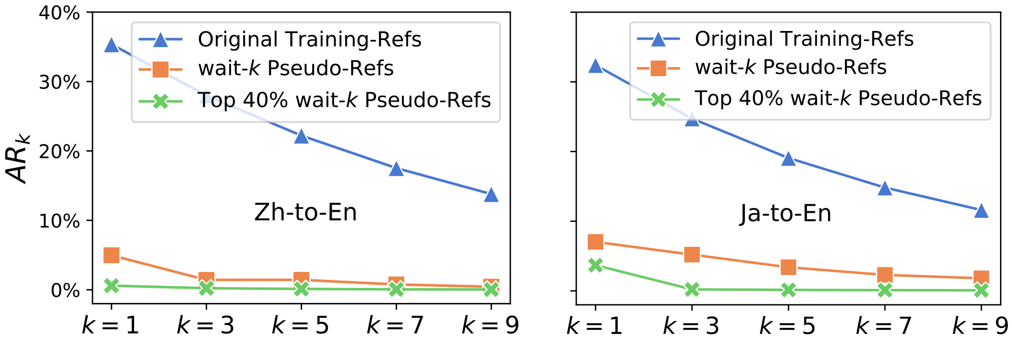


Figure 2.4: k-Anticipation rates (AR_k) of gold training references and Pseudo-Refs with various k. The top 40% Pseudo-Refs are selected in terms of BLEU.

references annotated by human interpreters with sight-interpreting[1] for the test set. For Ja→En translation, we use ASPEC corpus (3M pairs). Following Morishita et al. [67], we only use the first 1.5M parallel sentences and discard the rest noisy data. We use the dev and test datasets in ASPEC with 1,790 and 1,812 pairs. We preprocess the data with Mecab [55] as the word segmentation tool and Unidic [100] as its dictionary. Consecutive Japanese tokens which only contain Hiragana characters are combined to reduce the redundancy.

The full-sentence model is trained on the original training set. We use *fast_align* [35] as the word aligner (Model 2 for anticipation and Model 1 for hallucination) and train it on the training set. All the datasets are tokenized with BPE [78]. We implement wait-k policies on base Transformer [88] following Ma et al. [62] for all experiments

2.4.2 Results

We compare the performance of wait-k models trained on three different settings: (i) original training references only; (ii) original training references with all Pseudo-Refs; (iii) original training references with top 40% Pseudo-Refs in sentence-level BLEU.

Chinese-to-English Table 2.1 shows the results of Zh→En translation. Compared with using original references only, adding Pseudo-Refs substantially improves the translation quality and

[1] Sight interpreting refers to (real-time) oral translation of written text. It is considered as a special variant of simultaneous interpretation but with better translation quality.

(4-reference BLEU)		k=1	k=3	k=5	k=7	k=9	Avg.△
Training- Refs (*)	BLEU ↑	29.7	32.1	34.2	35.6	37.6	
	HR% ↓	*8.4*	*7.8*	*6.4*	*6.0*	*5.8*	
*+100% Pseudo-Refs	BLEU ↑	31.8	32.6	35.9	37.9	**39.4**	+1.7 (5.0%)
	HR% ↓	***5.5***	*7.4*	*5.4*	*5.2*	***4.6***	−1.3 (18.9%)
*+Top 40% Pseudo-Refs	BLEU ↑	**32.3**	**34.3**	**36.4**	**38.4**	38.8	+2.2 (6.5%)
	HR% ↓	*5.9*	***5.8***	***5.3***	***5.1***	*5.3*	−1.4 (20.3%)

Table 2.1: BLEU scores and hallucination rates (*HR*) of Zh→En wait-k models on the test set against the original 4 references. (Full-sentence BLEU: 39.9).

(single-reference BLEU)	k=1	k=3	k=5	k=7	k=9	Avg.△
Training-Refs (*)	10.9	12.1	13.0	13.7	13.8	
*+Top 40% Pseudo-Refs	**12.6**	**14.2**	**13.9**	**14.2**	**14.1**	+1.1 (7.5%)

Table 2.2: BLEU scores of Zh→En wait-k models on the test set, taking human sight interpretation as reference.

reduces hallucination rate. The filtered 40% Pseudo-Refs achieve the best results except $k = 9$. Fig. 2.4 shows that the generated Pseudo-Refs can significant reduce the k-anticipation rate compared with the original training references, especially for smaller k. As shown in Table 2.2, if taking the human sight-interpreting result as a single reference, the improvement is more salient than evaluated on the standard 4 references (+7.5% vs. +6.5%), which confirms that our method tend to translate in a "*syntactic linearity*" fashion like human sight and simultaneous interpreters [63].

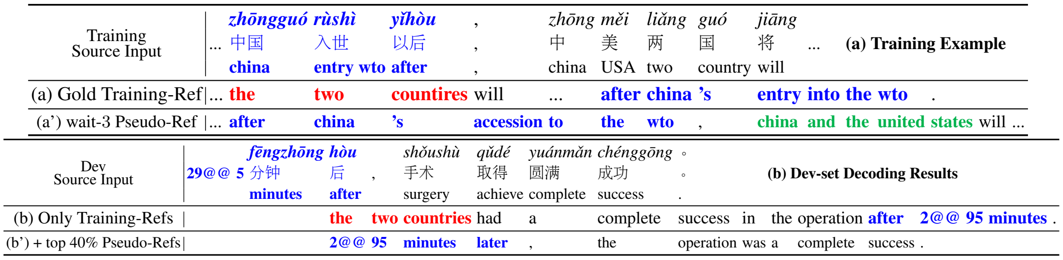
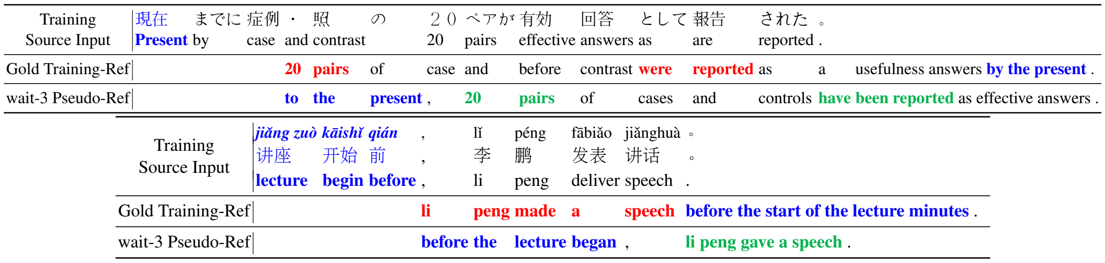

Training Source Input
zhōngguó rùshì yǐhòu , zhōng měi liǎng guó jiāng
... 中国 入世 以后 , 中 美 两 国 将 ...
china entry wto after , china USA two country will
(a) Training Example
(a) Gold Training-Ref | ... the two countires will ... after china 's entry into the wto .
(a') wait-3 Pseudo-Ref | ... after china 's accession to the wto , china and the united states will ...
Dev Source Input
fēngzhōng hòu shǒushù qǔdé yuánmǎn chénggōng 。
29@@ 5 分钟 后 , 手术 取得 圆满 成功 。
minutes after surgery achieve complete success .
(b) Dev-set Decoding Results
(b) Only Training-Refs | the two countries had a complete success in the operation after 2@@ 95 minutes .
(b') + top 40% Pseudo-Refs | 2@@ 95 minutes later , the operation was a complete success .

Figure 2.5: In the training example in (a), the gold reference anticipates "the two countries", which encourages the wait-k model trained on it to make irrelevant hallucination after any temporal phrase; see the decoding example in (b). Training with the pseudo-reference in (a') fixes this problem, resulting in the correct translation in (b').

(single-reference BLEU)		k=3	k=5	k=7	k=9	Avg.△
Training-Refs (*)	BLEU ↑	16.6	19.0	20.8	21.7	
	HR% ↓	*10.8*	*7.3*	*6.5*	*6.2*	
*+100% Pseudo-Refs	BLEU ↑	17.7	18.9	20.8	22.2	+0.3 (1.5%)
	HR% ↓	***6.5***	***6.2***	***5.6***	*5.3*	−1.4 (18.2%)
*+Top 40% Pseudo-Refs	BLEU ↑	**17.9**	**19.2**	**21.5**	**22.5**	+0.6 (3.1%)
	HR% ↓	*8.3*	*7.6*	*6.0*	***5.2***	−0.7 (9.1%)

Table 2.3: BLEU scores and *HR* of Ja→En wait-k models on the test set. (Full-sentence: 28.4).

Fig. 3.7 shows an example of how the wait-k model is improved by generated Pseudo-Refs. In this example, the original training references actively delay the translation of adverbial clause (time). It makes the model learn to anticipate the subject before its appearance. It is common in the original set. Fig. 2.6 shows two other examples of generated pseudo references on Ja→En and Zh→En, respectively. The generated pseudo-references are obviously more ideal than the original references. We also show several examples of solving other avoidable anticipations in Figs. 2.7–2.10. Besides changing the structure of training references, the full-sentence model also has the ability to generate Pseudo-Ref that avoids anticipation by adding several prepositions while preserving the meaning. There is an illustrated example in Fig. 2.9.



Training Source Input	現在 までに 症例 ・ 照 の ２０ ペアが 有効 回答 として 報告 された 。 **Present** by case and contrast 20 pairs effective answers as are reported .
Gold Training-Ref	**20 pairs** of case and before contrast **were reported** as a usefulness answers **by the present** .
wait-3 Pseudo-Ref	**to the present** , **20 pairs** of cases and controls **have been reported** as effective answers .

Training Source Input	***jiǎng zuò kāishǐ qián*** , lǐ péng fābiǎo jiǎnghuà 。 讲座 开始 前 , 李 鹏 发表 讲话 。 **lecture begin before** , li peng deliver speech .
Gold Training-Ref	**li peng made a speech before the start of the lecture minutes** .
wait-3 Pseudo-Ref	**before the lecture began** , **li peng gave a speech** .

Figure 2.6: Two examples dealing with adverbial clause delay. The adverbial clauses are at the end of the training references. This introduces anticipation during training and hallucination during decoding.

Japanese-to-English Table 2.3 shows the results of Ja→En translation task. Japanese-to-English simultaneous translation is a more difficult task due to long distance reorderings (SOV-to-SVO); many Japanese sentences are difficult to be translated into English monotonically. Besides that, the test set has only one single reference and does not cover many possible expressions. Results show that filtered Pseudo-Refs still improve the translation quality (Tab. 2.3), and reduces anticipation (Fig. 2.4) and hallucination (Tab. 2.3).

Training Source Input	*wǔjiǎodàlóu* 五角大楼 pentagon	*méiyǒu* 没有 not	*xuānbù* 宣布 announce	*xīn* 新 new	*de* 的 's	***fāshè*** 发射 **launch date**	***rìqí*** 日期	。 。
Gold Training-Ref				no	new	**launch date**		was announced by the pentagon .
wait-3 Pseudo-Ref				the	pentagon has		not	announced a new **launch date** .

Figure 2.7: The training reference uses passive voice while the source sentence uses active voice. This kind of problem often appears in sentences with “there be" (e.g. Fig. 2.8). The generated Pseudo-Ref can avoid anticipation by keeping the active voice as the source sentence.

Training Source Input	*liǎng* 两 two	*guó* 国 country	*jīngmào* 经贸 economic trade	*hézuò* 合作 corperation	*cúnzài* 存在 exist	*zhe* 着	***hěn*** 很 **very**	***dà*** 大 **big**	***de*** 的 **'s**	***qiánlì*** 潜力 **potential**	。 。 .
Gold Training-Ref				there	is	**very**	**great**	**potential**	for	economic and trade cooperation between the two countries .	
wait-3 Pseudo-Ref				the	economic and		trade	cooperation between the			two countries has **great potential** .

Figure 2.8: A similar example in which the pseudo-reference avoids the anticipation brought by the “there be” phrase in the gold reference.

Training Source Input	*dàn* 但 but	*xiéyì* 协议 agreement	*hái* 还 also	*xūyào* 需要 need	*dédào* 得到 get	*sūdān* 苏丹 sudan	*nèigé* 内阁 cabinet	*de* 的 's	***pīzhǔn*** 批准 **approval**	。 .		
Gold Training-Ref				but	the	agreement still		needs	**approval**	by the	sud@@ anese cabinet	.
wait-3 Pseudo-Ref				but	the	agreement still		needs to		be **approved** by	the	sud@@ anese cabinet .

Figure 2.9: An example about improving the reference by adding preposition. The generated Pseudo-Ref avoids anticipation by adding a preposition “to”. Besides changing the structure of training references, the full-sentence model also has the ability to generate Pseudo-Ref that avoids anticipation by adding several prepositions while preserving the meaning.

2.5 Related work

In the pre-neural statistical MT era, there exist several efforts using source-side reordering as a preprocessing step for full-sentence translation [27, 37, 97]. Unlike this work, they rewrite the source sentences. But in the simultaneous translated scenario, the source input is incrementally revealed and unpredictable. Zheng et al. [106] propose to improve full sentence translation by

Training Source Input	*wǒmen*	*de*	*xīnwén*	*méitǐ*	*nénggòu*	*dédào*	*rénmín*	*de*	*xìnrèn*	,	***gēnběn***	***yuányīn***	*jiù*	*zài*	*zhèlǐ*	.
	我们	的	新闻	媒体	能够	得到	人民	的	信任	，	根本	原因	就	在	这里	。
	we	’s	news	media	can	get	people	’s	trust	,	**fundamental**	**reason**	that	on	this	.
Gold Training-Ref	this	is	the	**fundamental reason**		why	our	news	media can		be	trust	by	the people		.
wait-3 Pseudo-Ref				our	news	media can		obtain the		trust of		the	people ,		the	**fundamental reason** for this .
wait-5 Pseudo-Ref						our	news	media can		win	the	trust	of	the people, and this is the **fundamental reason** .		

Figure 2.10: Comparisons of Pseudo-Refs using different wait-k policies. These examples also show the trade-off between latency and fluency of pseudo-references. Using wait-3 policy can effectively reduce the anticipation, but the translation is not completely correct due to the requirement of delay requirement is too small. Using wait-5 policy can not only avoid anticipation but also obtain high quality Pseudo-Ref.

generating pseudo-references from multiple gold references, while our work does not require the existence of *multiple* gold references and is designed for simultaneous translation.

This work is closely related to the work of He et al. [42], which addresses the same problem but only in the special case of Ja→En translation, and uses handwritten language-specific syntactic transformations rules to rewrite the original reference into a more monotonic one. By comparison, our work is much more general in the following aspects: (a) it is not restricted to any language pairs; (b) it does not require language-specific grammar rules or syntactic processing tools; and (c) it can generate pseudo-references with a specific policy according to the requirement of latency.

2.6 Summary

In this chapter, we introduce a simple but effective method to generate more monotonic pseudo references for simultaneous translation. We leverage the full-sentence trained model to generate translation in test-time wait-k mode. We take generated translation results as our pseudo references, which contain fewer reorderings with the source sentence. These pseudo references cause fewer anticipations in training and can substantially improve simultaneous translation quality by being added to the training data.

Chapter 3: Direct Simultaneous Speech-to-Text Translation with

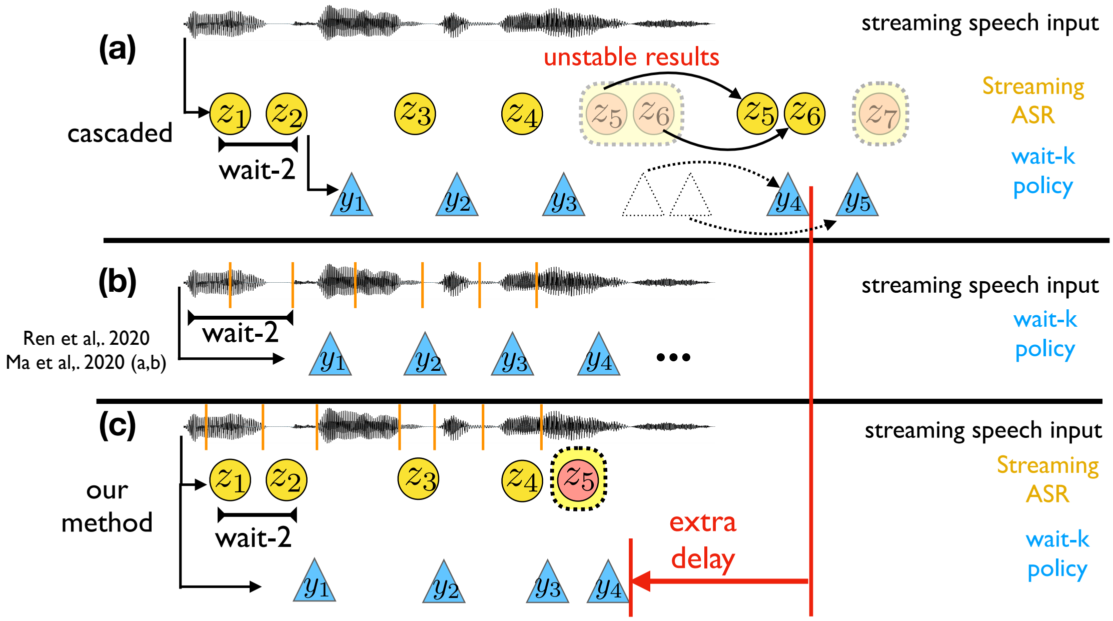


Figure 3.1: Comparison between (a) cascaded pipeline, (b) direct simultaneous ST, and (c) our ASR-assisted simultaneous ST. In (a), streaming ASR keeps revising some tail words for better accuracy, but causing extra delays to MT. Method (b) directly translates source speech without using ASR. Our work (c) uses the intermediate results of the streaming ASR module to guide the decoding policy of (but not feed as input to) the speech translation module. Extra delays between ASR and MT are reduced in direct translation systems (b–c).

As we discussed in Chapter 1, the conventional cascaded method has several limitations:

(a) **error propagation**, (b) **extra latency** (see Fig. 3.1), (c) **information loss** and (d) non-orthographic language.

To overcome the above issues, some recent efforts [77, 65, 64] attempt to directly translate the source speech into target text simultaneously by adapting text-based wait-k strategy [62]. However, unlike simultaneous translation whose input is already segmented into words or subwords, in speech translation, the key challenge is to figure out the number of valid tokens within a given source speech segment in order to apply the wait-k policy. Ma et al. [65, 64] simply assume a fixed number of words within a certain number of speech frames, which does not consider various aspects of speech such as different speech rate, duration, pauses and silences, all of which are common in realistic speech. Ren et al. [77] design an extra Connectionist Temporal Classification (CTC)-based speech segmenter to detect the word boundaries in speech. However, the CTC-based segmenter inherits the same shortcoming of CTC, which only makes local predictions, thus limiting its segmentation accuracy. On the other hand, to alleviate the error propagation, Ren et al. [77] employ several different knowledge distillation techniques to learn the attentions of ASR and MT jointly. These knowledge distillation techniques are complicated to train and it is an indirect solution for the error propagation problem.

3.2 Methods

We instead present a simple but effective solution (see Fig. 3.2) by employing two separate, but synchronized, decoders, one for streaming ASR and the other for End-to-End Speech-to-text Translation (E2E-ST).

Our key idea is to use the intermediate results of streaming ASR to guide the decoding policy of, but not feed as input to, the E2E-ST decoder. We look at the beam of streaming ASR to decide the number of tokens within the given source speech segment. Then it is straightforward for the E2E-ST decoder to apply the wait-k policy and decide whether to commit a target word or to wait for more speech frames. During training time, we jointly train ASR and E2E-ST tasks with a shared speech encoder in a multi-task learning (MTL) fashion to further improve the translation accuracy. We also note that having streaming ASR as an auxiliary output is extremely useful in real application scenarios where the user often wants to see both the transcript and the translation. En-to-De and En-to-Es experiments on the MuST-C dataset demonstrate that our proposed technique achieves substantially better translation quality at similar level of latency.

In text-to-text simultaneous translation, the input stream is already segmented. However,

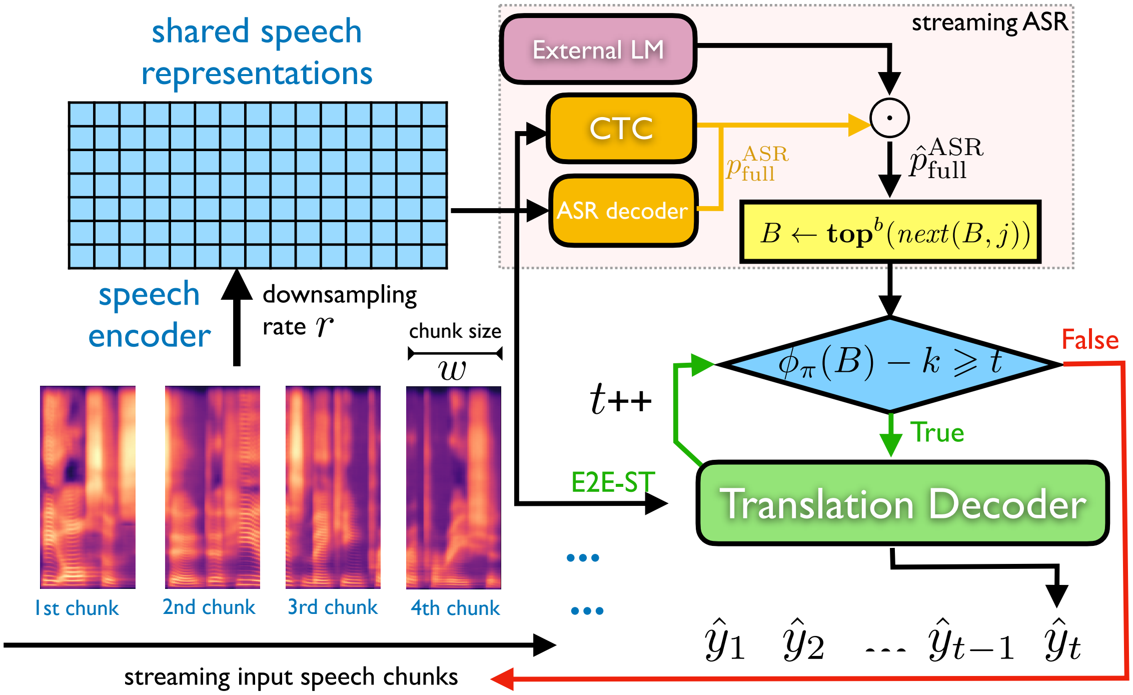

shared speech representations
streaming ASR
External LM
CTC
ASR decoder
$p^{\text{ASR}}_{\text{full}}$
$\hat{p}^{\text{ASR}}_{\text{full}}$
$B \leftarrow \mathbf{top}^b(next(B, j))$
speech encoder
downsampling rate r
chunk size
w
$\phi_\pi(B) - k \geqslant t$
False
t++
True
E2E-ST
Translation Decoder
1st chunk
2nd chunk
3rd chunk
4th chunk
$\hat{y}_1$ $\hat{y}_2$ ... $\hat{y}_{t-1}$ $\hat{y}_t$
streaming input speech chunks

Figure 3.2: Decoding for synchronized streaming ASR and E2E-ST. Speech signals are fed into the encoder chunk by chunk. For each new-coming speech chunk, we look at the current streaming ASR beam (B) to decide the translation policy. See details in Algorithm 1.

when we deal with speech frames as source inputs, it is not easy to determine the number of valid tokens within certain speech segments. Therefore, to better guide the translation policy, it is essential to detect the number of valid tokens accurately within low latency. Different from the sophisticated design of speech segmenter in Ren et al. [77], we propose a simple but effective method by using a synchronized streaming ASR and using its beam to determine the number of words within certain speech segments. Note that we only use streaming ASR for source word counting, but the translation decoder does not condition on any of ASR's output.

3.2.1 Streaming ASR-Guided Simultaneous ST

As shown in Fig. 3.2, at inference time, the speech signals are fed into the ST encoder by a series of fixed-size chunks $\bar{\mathbf{s}}_{[1:i]} = [\bar{\mathbf{s}}_1, ..., \bar{\mathbf{s}}_i]$, where $w = |\bar{\mathbf{s}}_i|$ can be chosen from 32, 48 and 64 frames

Algorithm 1 Streaming ASR-guided Simultaneous ST

```
Input: speech chunks $\bar{\mathbf{s}}_{[1:T]}$; $k$; $\phi_\pi(B_j)$; streaming decoding models: $p^{\text{ST}}_{\text{full}}$ and $\hat{p}^{\text{ASR}}_{\text{full}}$
Initialize: ASR and ST indices: $j = t = 0$; $B = B_0$
for $i = 1 \sim T$ do                                          ▷ feed speech chunks
    repeat $w/r$ steps                                         ▷ do ASR beam search $w/r$ steps
        $B \leftarrow \mathbf{top}^b(\mathit{next}(B, j))$; $j$++           ▷ ASR beam search
    while $\phi_\pi(B) - k \geqslant t$ do                         ▷ new tokens?
        $\hat{y}_{t+1} \leftarrow p^{\text{ST}}_{\text{wait-}k}(y_{t+1} \mid \bar{\mathbf{s}}_{[1:i+1]}, \hat{\mathbf{y}}_{\leq t}; \boldsymbol{\theta}^{\text{ST}}_{\text{full}})$
        yield $\hat{y}_{t+1}$; $t$++                                ▷ commit translation to user
```

of spectrogram.

As a result of the CNN encoder, there is down sampling rate r (e.g., we use $r = 4$), from spectrogram to encoder hidden states. For example, when we receive a chunk of 32 frames, the encoder will generate 8 more hidden states. In conventional streaming ASR, the number of steps of beam search is the same as the number of hidden states.

We denote B_j to be the beam at time step j, which is an ordered list of size of b, and it expands to the next beam B_{j+1} with the same size:

$$\begin{aligned} B_0 =& [\langle \texttt{<s>},\ \hat{p}^{\text{ASR}}_{\text{full}}(\texttt{<s>} \mid \bar{\mathbf{s}}_0; \boldsymbol{\theta})\rangle] \\ B_j =& \mathbf{top}^b(\mathit{next}(B_{j-1}, j)) \\ \mathit{next}(B, j) =& \{\langle \mathbf{z} \circ z_j,\ p \cdot \hat{p}^{\text{ASR}}_{\text{full}}(z_j \mid \bar{\mathbf{s}}_{\leq \tau(j)}, \mathbf{z}; \boldsymbol{\theta})\rangle \mid \\ & \langle \mathbf{z}, p\rangle \in B, z_j \in V\} \end{aligned}$$

where $\mathbf{top}^b(\cdot)$ returns the top b candidates, and $\mathit{next}(B, j)$ expands the candidates from the previous step to the next step. Each candidate is a pair $\langle \mathbf{z}, p\rangle$, where $\mathbf{z}$ is the current prefix and p is the accumulated probability from joint score between an external language model, CTC and ASR probabilities, $\hat{p}^{\text{ASR}}_{\text{full}}$. We denote the number of observable speech chunks at j step as $\tau(j) = \lceil j * r/w \rceil$. And vice versa, for each new speech chunk, ASR beam search will advance for w/r steps.

Note CTC often commits empty tokens ϵ due to empty speech frames, and the lengths of different hypotheses within beam of streaming ASR are quite different from each other. To take every hypothesis into consideration, we design two policies to decide the number of valid tokens.

- **Longest Common Prefix** (LCP) uses the length of longest shared prefix in the streaming ASR beam as the number of valid tokens within given speech. This is the most conserva-

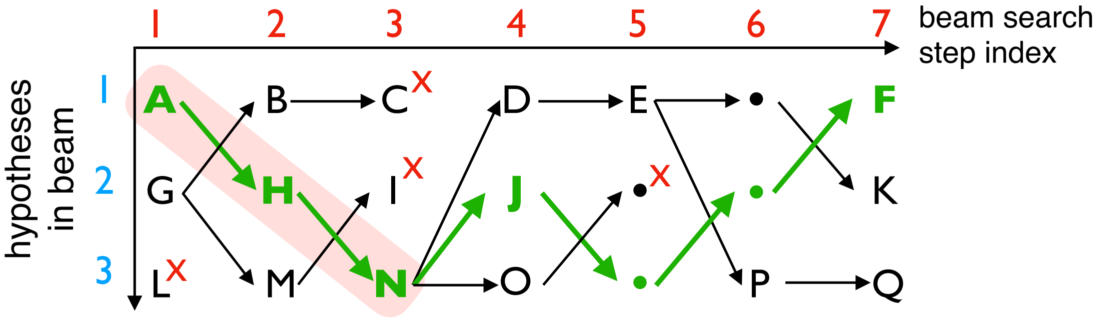


Figure 3.3: An example of streaming ASR beam search with beam size 3. LCP is shaded in red ($\phi_{\text{LCP}}(B_7)=3$); SH is highlighted in bold ($\phi_{\text{SH}}(B_7)=5$). We use • to represent empty outputs in some steps caused by CTC.

tive strategy, which has similar latency to cascaded methods.

- **Shortest Hypothesis** (SH) uses the length of shortest hypothesis in the current streaming ASR beam as the number of valid tokens.

More formally, let $\phi_\pi(B)$ denote the number of valid tokens in the beam B under policy π:

$$\begin{aligned}\phi_{\text{LCP}}(B) &= \max\{i \mid \exists \mathbf{z}', s.t. \forall \langle \mathbf{z}, c\rangle \in B, \mathbf{z}_{\leq i} = \mathbf{z}'\}\\ \phi_{\text{SH}}(B) &= \min\{|\mathbf{z}| \mid \langle \mathbf{z}, c\rangle \in B\}\end{aligned}$$

For example in Fig. 3.3, $\phi_{\text{LCP}}(B_7)=3$, $\phi_{\text{SH}}(B_7)=5$. Also note that $\phi_{\text{LCP}}(B) \leq \phi_{\text{SH}}(B)$ for any beam B, and that both policies are *monotonic*, i.e. $\phi_\pi(B_j) \leq \phi_\pi(B_{j+1})$ for $\pi \in \{\text{LCP}, \text{SH}\}$ and all j.

Note we always feed the entire observable speech segments into ST for translation, and streaming ASR-generated transcript is not used for translation, so LCP might have similar latency with cascaded methods but the translation accuracy is much better because more information on the source side is revealed to the translation decoder.

As shown in Algorithm 1, during simultaneous ST, we monitor the value of $\phi_\pi(B_j)$ while speech chunks are gradually fed into system. When we have $\phi_\pi(B) - k \geqslant t$ where t is the number of translated tokens, the ST decoder will be triggered to generate one new token as follows:

$$\hat{y}_t = \operatorname*{\mathbf{argmax}}_{y_t} p_{\text{wait-}k}(y_t \mid \bar{\mathbf{s}}_{[1:\tau(j)]}, \hat{\mathbf{y}}_{<t}; \hat{\boldsymbol{\theta}}^{\text{ST}}_{\text{full}}) \tag{3.1}$$

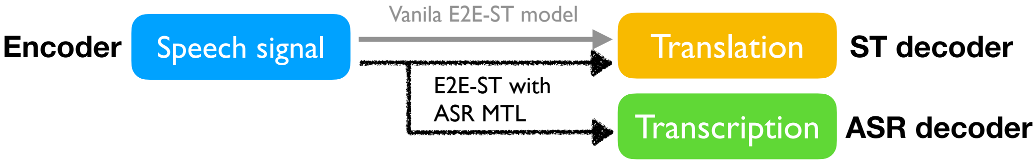


Figure 3.4: We use full-sentence MTL framework to jointly learn ASR and ST with a shared encoder.

3.2.2 Joint Training between ST and ASR

Different from existing simultaneous translation solutions from [77, 65, 64], which make adaptations over vanilla E2E-ST architecture as shown in gray line of Fig. 3.4, we instead use simple MTL architecture which performs joint full-sentence training between ST and ASR:

$$\hat{\boldsymbol{\theta}}^{\text{ST}}_{\text{full}}, \hat{\boldsymbol{\theta}}^{\text{ASR}}_{\text{full}} = \operatorname*{\mathbf{argmax}}_{\boldsymbol{\theta}^{\text{ST}}_{\text{full}}, \boldsymbol{\theta}^{\text{ASR}}_{\text{full}}} \prod_{(\mathbf{s},\mathbf{y}^*,\mathbf{z}^*)\in D} p^{\text{ST}}_{\text{full}}(\mathbf{y}^* \mid \mathbf{s}; \boldsymbol{\theta}^{\text{ST}}_{\text{full}}) \cdot p^{\text{ASR}}_{\text{full}}(\mathbf{z}^* \mid \mathbf{s}; \boldsymbol{\theta}^{\text{ASR}}_{\text{full}})$$

For ASR training, we use hybrid CTC/Attention framework [94]. Note that we train ASR and ST MTL with full-sentence fashion for simplicity and training efficiency, and only perform wait-k decoding policy at inference time. Also, $\boldsymbol{\theta}^{\text{ST}}_{\text{full}}$ and $\boldsymbol{\theta}^{\text{ASR}}_{\text{full}}$ share the same speech encoder.

3.3 Experiments

3.3.1 Datasets and Models

We conduct experiments on English-to-German (En→De) and English-Spanish (En→Es) translation on MuST-C [31]. We employ Transformer [88] as the basic architecture and LSTM [43] for LM. For streaming ASR decoding we use a beam size of 5. Translation decoding is greedy due to incremental commitment.

Raw audios are processed with Kaldi [74] to extract 80-dimensional log-Mel filterbanks stacked with 3-dimensional pitch features using a 10ms step size and a 25ms window size. Text is processed by SentencePiece [54] with a joint vocabulary size of 8K. We take Transformer [88] as our base architecture, which follows 2 layers of 2D convolution of size 3 with stride size of 2. The Transformer model has 12 encoder layers and 6 decoder layers. Each layer has 4 attention head with a size of 256. Our streaming ASR decoding method follows Moritz et al. [68]. We employ 10 frames look ahead for all experiments. For LM, we use 2 layers stacked LSTM [43] with 1024-dimensional hidden states, and set the embedding size as 1024. LM are trained on English transcript from the corresponding language pair in MuST-C corpus. For the cascaded model, we train ASR and MT models on Must-C dataset respectively, and they have the same Transformer architecture of our ST model. Our experiments are run on 8 1080Ti GPUs. And the we report the case-sensitive detokenized BLEU.

3.3.2 Translation quality against latency

In order to clearly compare with related works, we evaluate the latency with AL defined in Ma et al. [65] and AP defined in Ren et al. [77]. As shown in Fig. 3.5, for En→De, results are on the dev set to be consistent with Ma et al. [65]. Compared with baseline models, our method achieves much better translation quality with similar latency. To validate the effectiveness of our method, we compare our method with Ren et al. [77] on En→Es translation. Their method does not evaluate the plausibility of the detected tokens, so it has a more aggressive decoding policy which results in lower latency. However, our method can still achieve better results with slightly lower latency. Besides that, our model is trained in full-sentence mode, and only decodes with wait-k at inference time, which is very efficient to train. Our test-time wait-k could achieve similar quality with their genuine wait-k (i.e., retrained) models which are very slow to train. When we compare with their test-time wait-k, our model significantly outperforms theirs.

We further evaluate our method on the test set of En→De and En→Es translation. As shown in Fig. 3.6, compared with the cascaded model, our model has notable successes in latency and translation quality. To verify the online usability of our model, we also show computational-aware latency. Because our chunk window is 480ms, and the latency caused by the computation is smaller than this window size, which means that we can finish decoding the previous speech chunk when the next speech chunk needs to be processed, so our model can be effectively used online.

Fig. 3.7 demonstrates that our method can effectively avoid the error propagation and obtain better latency compared to the cascaded model.

Model	En→De			En→Es		
	$w = 32$	$w = 48$	$w = 64$	$w = 32$	$w = 48$	$w = 64$
LCP	17.31	17.54	17.95	21.94	21.92	22.36
– *LM*	14.60	15.66	15.91	18.54	19.15	19.95
– *LM* & *AD*	13.76	14.82	15.26	17.42	18.06	19.32
SH	16.04	15.82	15.87	20.45	20.18	19.84
– *LM*	13.76	14.01	13.84	17.31	17.21	17.78
– *LM* & *AD*	10.44	11.25	11.65	13.61	14.27	14.62

Table 3.1: BLEU score of wait-1 decoding with different chunk sizes and ASR scoring functions. *AD* denotes ASR Decoder. *LM* denotes Language Model.

3.3.3 Effect of chunk size and joint decision

Table 4.6 shows that the results are relatively stable with various chunk sizes. It can be flexible to balance the response frequency and computational ability. We explore the effectiveness of ASR joint scoring, and observe that the translation quality drops a lot without LM. Without LM and AD, our token recognition approach is similar to the speech segmentation in Ren et al. [77], which implies that their model is hard to segment the source speech accurately, leading to unreliable translation decisions for ST.

3.4 Summary

We proposed a simple but effective ASR-assisted simultaneous E2E-ST framework. The streaming ASR module can accurately detect the number of tokens within the given speech and guide (but not give direct input to) the wait-k policy for simultaneous translation. Our method improves ST accuracy with similar latency.

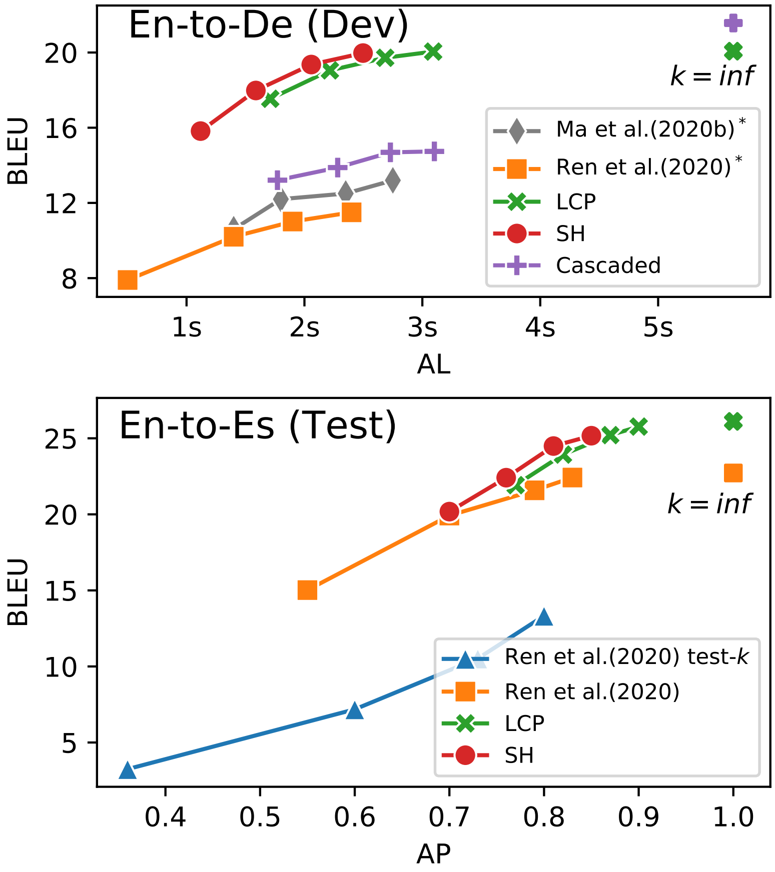


Figure 3.5: Translation quality v.s. latency. The dots on each curve represents different wait-k policy with k=1,3,5,7 from left to right respectively. Baseline* results are from Ma et al. [65]. k=inf is full-sentence decoding for ASR and translation. test-k denotes testing time wait-k. We use a chunk size of 48.

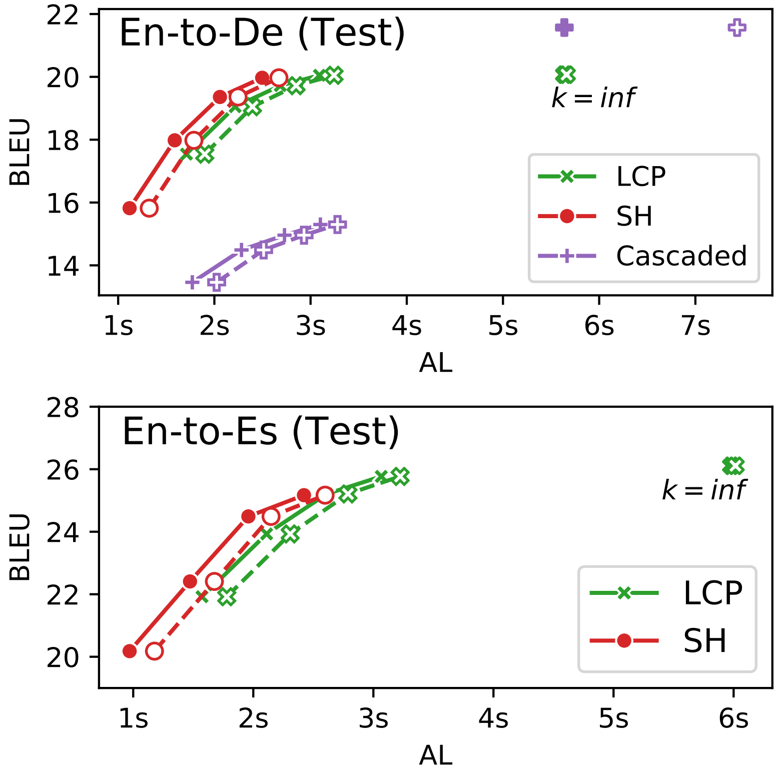


Figure 3.6: Translation quality against latency. Each curve represents decoding with wait-k policy, k=1,3,5,7 from left to right. The dashed lines and hollow markers indicate the latency considering the computational time. The chunk size is 48.

chunk index	1	2	3	4	5	6	end
Gold transcript	can I be	**honest**	*SIL*	I don 't **love**	**that question**	*SIL*	
Gold translation	Darf ich **ehrlich** sein ?			Ich **mag diese Frage** nicht .			
Streaming ASR	can I			be **on this** I don 't	**love that question**		
simul-MT wait-3				Kann ich **da** sein ? "	Ich **liebe**		**diese Frage** nicht .
SH wait-3				Kann ich **ehrlich** sein ? Ich **liebe**	**diese Frage**		nicht .
LCP wait-3				Kann ich **ehrlich** sein ? Ich			**liebe diese Frage** nicht .

Figure 3.7: An example from the dev set of En→De translation. In the cascaded approach (streaming ASR + simul-MT wait-3), the ASR error (*"on this"* for *"honest"*) is propagated to the MT module, causing the wrong translation (*"da"*). Our methods give accurate translations ("ehrlich") with better latency (esp. for the SH policy, the output of *"diese Frage"* is synchronous with hearing *"that question"*). *"SIL"* denotes silence in speech.

Chapter 4: Improving Speech Translation with Multimodal Pretraining

Direct speech translation shows it superiority in avoiding error propagation and reducing latency, but it still suffers from the problem of data scarcity. Although we can use the triple data ({speech $\mathbf{s}$, transcript $\mathbf{x}$ in the source language, translation $\mathbf{y}$ in the target language}) for multi-task training, such data is also scarce, and not all languages have orthography systems.

4.1 Motivation

To improve the translation accuracy of E2E-ST models, researchers either initialize the encoder of ST with a pretrained ASR encoder [11, 9, 91] to get better representations of the speech signal, or perform Multi-Task Learning (MTL) with ASR to bring more training and supervision signals to the shared encoder [3, 2, 84, 60] (see Fig. 3.1). These methods improve the translation quality by providing more training signals to the encoder to learn better phonetic information and hidden representation correspondence [85].

However, both above solutions assume the existence of substantial speech transcriptions of the source language. Unfortunately, this assumption is problematic. On the one hand, for certain low-resource languages, especially endangered ones [14, 15], the source speech transcriptions are expensive to collect. Moreover, according to the report from Ethnologue [36][1], there are more than 3000 languages that have no written form or no standard orthography, making phonetic transcription impossible [34]. On the other hand, the amount of speech audios with transcriptions are limited (as they are expensive to collect), and there exist far more audios without any annotations. It will be much more straightforward and cheaper to leverage these raw audios to train a robust encoder directly.

In recent years, self-supervised learning (SSL) [72, 30, 86] has attracted much attention in the NLP community due to its strong performance to many downstream tasks. However, because of the difference in modality, the feature representation of speech is very different compared to the discrete distributed feature representation of text data, so these methods cannot be directly applied to speech.

[1] https://www.ethnologue.com/

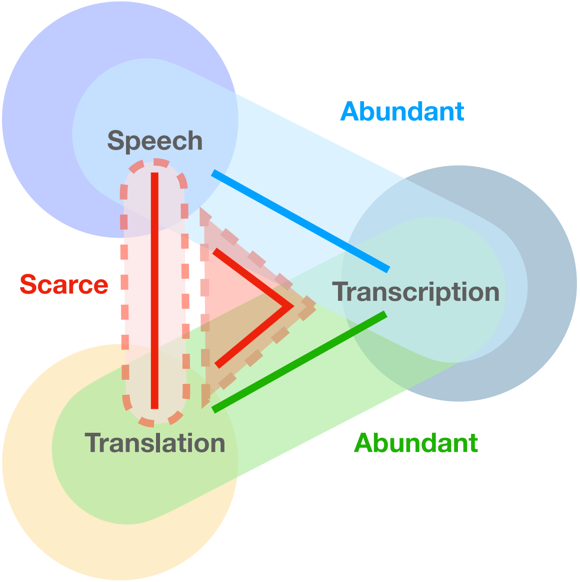
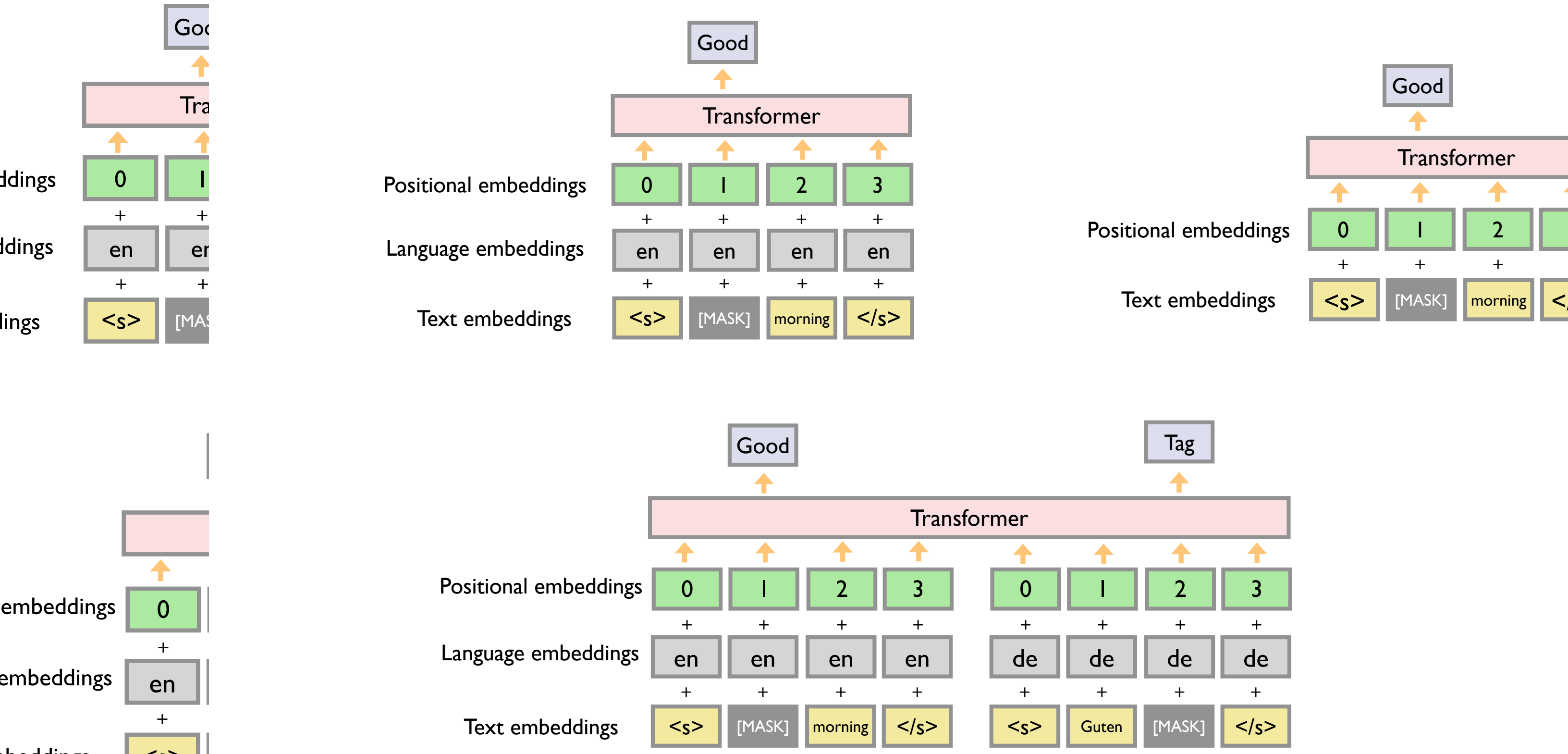


Figure 4.1: The quality of end-to-end speech translation models has been limited by the scarcity of speech translation datasets. However, there is an abundance of datasets for speech, text, speech recognition, and machine translation data that can be leveraged.

4.2 Related Works

4.2.1 Masked Language Modeling

Radford et al. [75], Howard and Ruder [44] and Devlin et al. [30] investigate language modeling for pretraining Transformer encoders. Unlike Radford et al. [75] using unidirectional language models for pretraining, Devlin et al. [30] proposes BERT which enables deep bidirectional representation pretraining by a masked language modeling (MLM) objective inspired by the Cloze task [87] which randomly masks some of the tokens from the input, with an objective to recover the masked word based only on its context. Their approaches lead to drastic improvements on several natural language understanding tasks including text classification [89],and question answering [76].



(b) Translation Language Model (TLM) for crosslingual text.

Figure 4.2: Previous work for text monomodal representation learning.

4.2.2 Translation Language Modeling

Lample and Conneau [56] extend MLM to cross-lingual pretraining by proposing two methods: one unsupervised that only relies on monolingual data, and one supervised that leverages parallel data with a new cross-lingual language model objective which is called Translation Language Model (TLM). As shown in Fig. 4.2(b), TLM encodes both source and target sentences from a parallel data after masking several tokens with `[MASK]`, and then learn to recover the masked tokens. Experiments show that TLM achieves state-of-the-art results on cross-lingual classification, unsupervised and supervised machine translation.

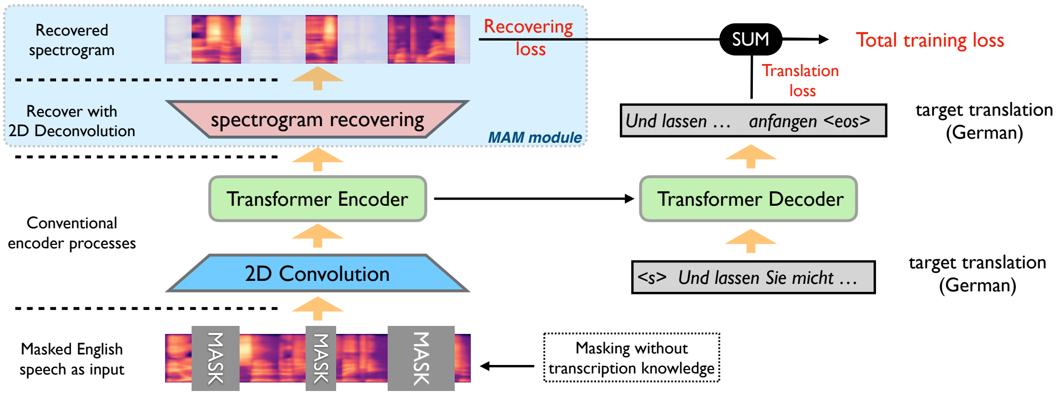

Recovered spectrogram
Recovering loss
SUM
Total training loss
Translation loss
Recover with 2D Deconvolution
spectrogram recovering
MAM module
Und lassen … anfangen <eos>
target translation (German)
Transformer Encoder
Transformer Decoder
Conventional encoder processes
2D Convolution
<s> Und lassen Sie micht …
target translation (German)
Masked English speech as input
MASK
MASK
MASK
Masking without transcription knowledge

Figure 4.3: MAM (in blue box) can be treated as one extra module besides standard Transformer encoder-decoder and convolution layers for processing speech signals.

4.3 Masked Acoustic Model

To relieve from the dependency on source language transcript, we present a straightforward yet effective solution, Masked Acoustic Modeling (MAM), to utilize the speech data in a self-supervised fashion without using any source language transcript, unlike other speech pretraining models [25, 92]. Aside from the regular training of E2E-ST (without ASR as MTL or pretraining), MAM masks certain portions of the speech input randomly and aims to recover the masked speech signals with their context on the encoder side. MAM not merely provides an alternative solution to improving E2E-ST, but also is a general technique that can be used as a pretraining module on arbitrary acoustic signals, e.g., multilingual speech, music, animal sounds.

As shown in Fig. 4.3, MAM can be used as part of training objective for ST task. Formally, we define a random replacement function over the original speech input $\mathbf{x}$:

$$\hat{\mathbf{x}} \sim \text{Mask}_{\text{span}}(\mathbf{x}, \lambda), \tag{4.1}$$

where $\text{Mask}(\cdot)_{\text{span}}$ is similar with SpanBERT [50], we first sample a serial of span widths and apply those spans randomly to different positions of the input signal $\mathbf{x}$ with a probability of λ (30% in our experiments). And then we replace those frames with with the same random initialized vector, $\epsilon \in \mathbb{R}^{d_x}$. We do not allow overlap in this case. Note that we use the same vector ϵ to represent all the corrupted frames (see one example in Fig.4.9(b)). Then we obtain a corrupted input $\hat{\mathbf{x}}$ and its corresponding latent representation $\hat{\boldsymbol{h}}$.

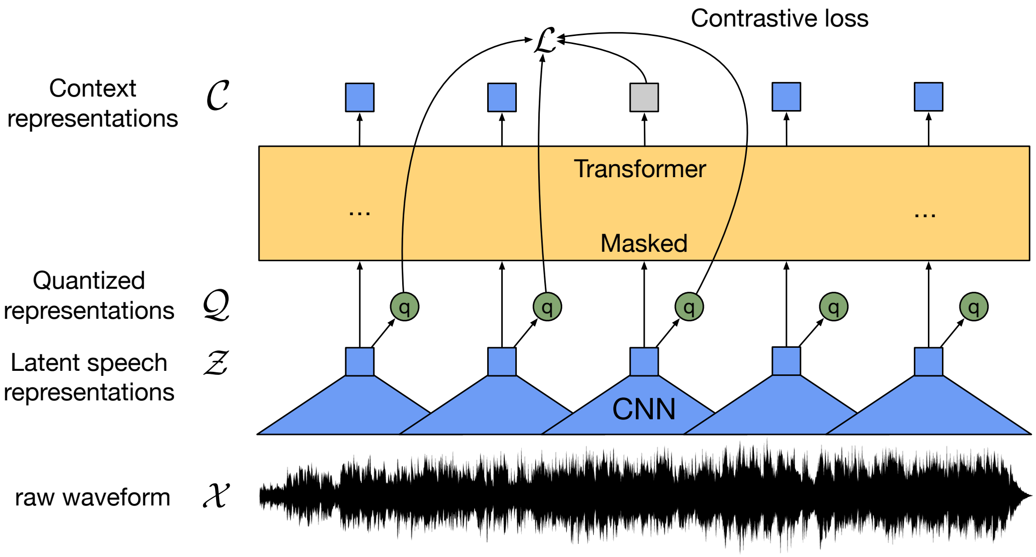


Figure 4.4: Wav2Vec 2.0.

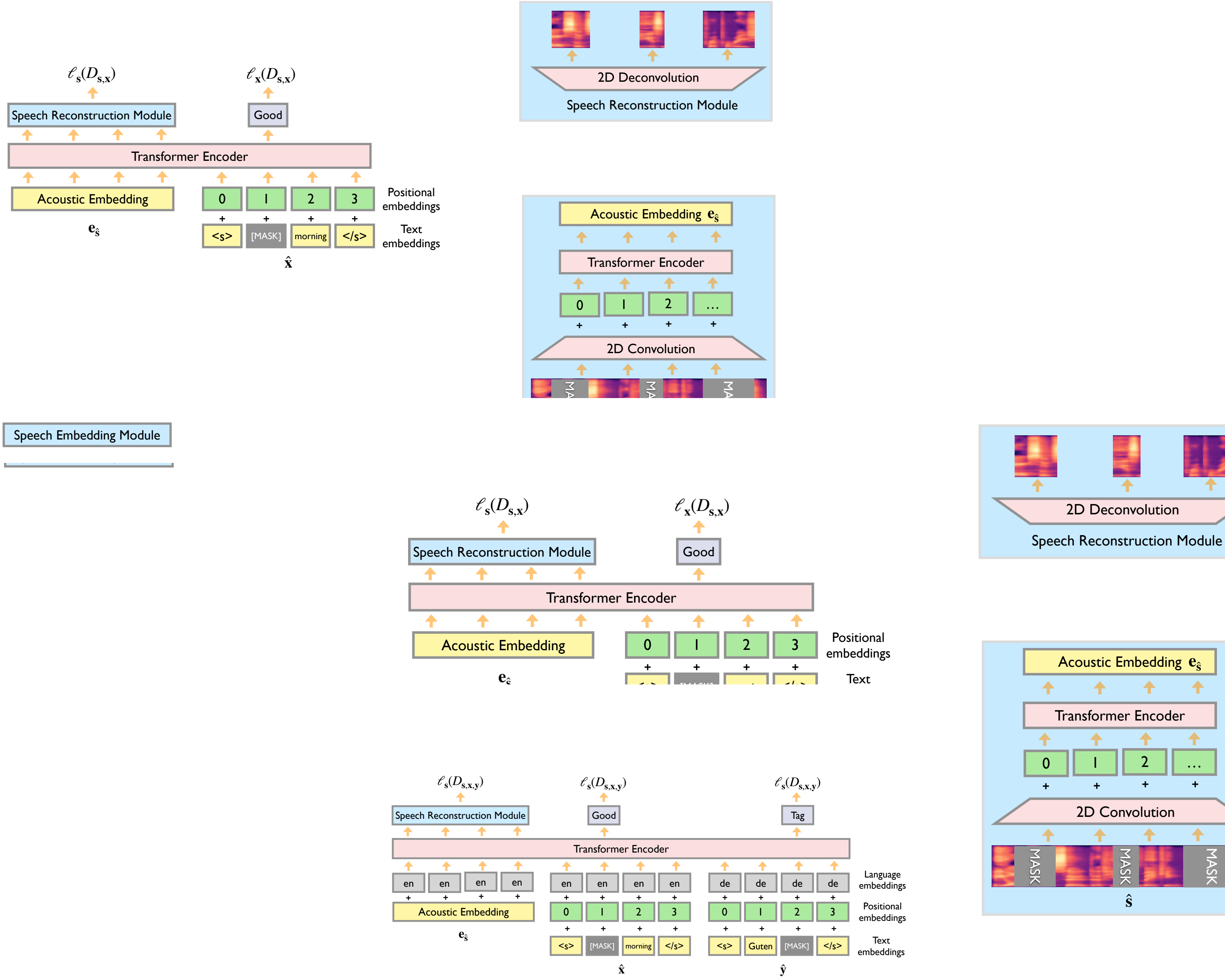
For MAM module, we have the following training objective to reconstruct the original speech signal with the surrounding context information with self-supervised fashion:

$$\ell_{\text{Rec}}(D_{\mathbf{s}}) = \textstyle\sum_{\mathbf{s} in D_{\mathbf{s}}} ||\mathbf{s} - \phi(f(\hat{\mathbf{s}}))||_2^2 \tag{4.2}$$

where ϕ is a reconstruction function which tries to recover the original signal from the hidden representation $f(\hat{\mathbf{s}})$ with corrupted inputs. For simplicity, we use regular 2D deconvolution as ϕ, and mean squared error for measuring the difference between original input and recovered signal.

The reconstruction module can be used in jointly training with ST to boost the performance of translation. And it can also be directly used in self-supervised training on arbitrary raw speech. We will show the experimental results in the following sections.

Different from BERT-style pretraining, MAM tries to recover the missing semantic information (e.g., words, subword units) and learns the capabilities to restore the missing speech characteristics and generate the original speech.



(d) Translation FAT-MLM

Figure 4.5: Fused Acoustic and Text-Masked Language Model (FAT-MLM).

4.4 Fused Acoustic and Text Masked Language Model

Parallel to MAM, Baevski et al. [7] proposes the wav2vec 2.0 pretraining model, which masks the speech input in the latent space and pretrains the model via a contrastive task defined over a quantization of the latent representations. It was shown in Fig [**?**]. Besides that, some other works in speech self-supervised learning [58, 24, 45] also successfully improved many speech related tasks, such as speech recognition.

However all these existing methods can only handle one modality, either text or speech, while joint acoustic and text representation is desired for many end-to-end spoken language processing tasks, such as spoken question answering [26] and end-to-end speech-to-text translation [61]. For example, end-to-end speech translation (ST) is desired due to its advantages over the pipeline paradigm, such as low latency, alleviation of error propagation, and fewer parameters [95, 12, 49, 83, 108, 21]. However, its translation quality is limited by the scarcity of large-scale parallel speech translation data while there exists sufficient data for speech recognition and text machine translation (Fig. 4.1). It would be helpful if source speech and bilingual text can be encoded into a unified representation via abundant speech recognition and text machine translation data. Liu et al. [61] show that jointly training a multi-modal ST encoder can largely improve the translation quality. However, their proposed representation learning method is constrained to the sequence-to-sequence framework and there is no experiment showing whether their proposed method can benefit from extra speech recognition and machine translation data.

Inspired by recent cross-lingual language model pretraining work [56] which shows the potential to unify the representations of different languages into one encoder, we propose a Fused Acoustic and Text Masked Language Model (FAT-MLM). This model jointly learns a unified representation for both acoustic and text input. In this way, we extend the masked language model's input from only acoustic or text data to multimodal corpora containing both acoustic and text data, such as speech recognition and speech translation for the first time (Fig. 4.1).

We further extend this Fused Acoustic and Text encoder to a sequence-to-sequence framework and present an end-to-end Speech Translation model (FAT-ST). This enables the model to be trained from both speech and text machine translation data into one single encoder-decoder model. Meanwhile, this model can also learn from speech recognition data using an extra FAT-MLM loss. This resolves the limitation of existing single encoder and decoder speech translation models, which can only learn from scarce parallel speech translation data, but neglects much larger scale speech recognition and text machine translation data (Fig. 4.1).

Although existing pretraining models show a strong representation learning ability and significantly improve upon many down-streaming tasks, they all can only learn the representation for either text or speech. However, a unified speech and text multi-modal representation is useful for many end-to-end spoken language processing tasks.

To address this problem, we propose the Fused Acoustic and Text Masked Language Model (FAT-MLM), a multimodal pretraining model which encodes acoustic, text into a unified representation. The idea is similar with Lample and Conneau [56] who propose to learn a unified representation of different languages. They first propose a method relying on the shared subword vocabulary to align different languages' representation. However this is unapplicable in our case because of the modality difference. Thus we propose a method similar to their second approach TLM which uses parallel speech recognition data. In the following sections, we first introduce the monolingual FAT-MLM and then show how to extend it to translation scenario.

4.4.1 Monolingual FAT-MLM

The monolingual FAT-MLM takes speech and transcription tuples as input, denotes as $D_{\mathbf{s},\mathbf{x}} = \{(\mathbf{s},\mathbf{x})\}$, where $\mathbf{s} = (s_1, ..., s_{|s|})$ is a sequence of acoustic features $s_i \in \mathbb{R}^{d_s}$ which can be the spectrogram or mel-spectrogram of the speech audio, and each s_i represents the frame-level speech feature, and $\mathbf{x} = (x_1, ..., x_{|\mathbf{x}|})$ is the sequence of corresponding transcription.

As shown in Fig. 4.5(b), similar with MAM, we first randomly mask several spans of $\mathbf{s}$ by a random masking function over the input $\mathbf{s}$:

$$\hat{\mathbf{s}} \sim \text{Mask}_{\text{span}}(\mathbf{s}, \lambda) \tag{4.3}$$

where $\text{Mask}_{\text{span}}(\cdot)$ replaces several random spans of $\mathbf{s}$ by probability of λ (30% in our work) with a random initialized vector $\epsilon_{\mathbf{s}} \in \mathbb{R}^{d_{\mathbf{s}}}$. Then we encode $\hat{\mathbf{s}}$ with Convolutions and a Transformer encoder for acoustic embeddings $\boldsymbol{e}_{\hat{\mathbf{s}}}$. Similarly, we randomly mask tokens in $\mathbf{x}$ by a random masking function over the input $\mathbf{s}, \mathbf{x}$:

$$\hat{\mathbf{x}} \sim \text{Mask}_{\text{token}}(\mathbf{x}, \lambda) \tag{4.4}$$

where $\text{Mask}_{\text{token}}(\cdot)$ replaces several tokens of $\mathbf{x}$ by probability of λ with a random initialized vector $\epsilon_{\text{token}} \in \mathbb{R}^{d_{\mathbf{x}}}$. Then we concatenate acoustic embeddings and source text embeddings $[\hat{\boldsymbol{e}}_{\mathbf{s}}; \hat{\mathbf{x}}]$, and obtain the latent representation $f([e_{\hat{\mathbf{s}}}; \hat{\mathbf{x}}])$ using another Transformer encoder, denoted as f.

Same with Lample and Conneau [56], we reset the positional embeddings for different types of input.

The training objective of monolingual FAT-MLM includes a speech reconstruction loss $\ell_{\mathbf{s}}(D_{\mathbf{s},\mathbf{x}})$ and a text reconstruction loss $\ell_{\mathbf{x}}(D_{\mathbf{s},\mathbf{x}})$. For speech input $\mathbf{s}$, we have the following training objective to reconstruct the original speech signal with the surrounding context information[2]:

$$\ell_{\mathbf{s}}(D_{\mathbf{s},\mathbf{x}}) = \textstyle\sum_{(\mathbf{s},\mathbf{x})\in D_{\mathbf{s},\mathbf{x}}} ||\mathbf{s} - g(f([e_{\hat{\mathbf{s}}};\hat{\mathbf{x}}]))||_2^2 \tag{4.5}$$

where g is a reconstruction function (we use 2D deconvolution in this work) which tries to recover the original signal from encoded representation $f([e_{\hat{\mathbf{s}}};\hat{\mathbf{x}}])$. We use mean squared error for measuring the difference between s and the reconstructed spectrogram. For transcription input $\mathbf{x}$, following Devlin et al. [30] we use cross entropy loss , denoted as

$$\ell_{\mathbf{x}}(D_{\mathbf{s},\mathbf{x}}) = -\textstyle\sum_{(\mathbf{s},\mathbf{x})\in D_{\mathbf{s},\mathbf{x}}} \log p(\mathbf{x} \mid [\boldsymbol{e}_{\hat{\mathbf{s}}};\hat{\mathbf{x}}]) \tag{4.6}$$

to reconstruct the masked token. The final loss for monolingual FAT-MLM is:

$$\ell_{\text{FAT-MLM}}(D_{\mathbf{s},\mathbf{x}}) = \ell_{\mathbf{s}}(D_{\mathbf{s},\mathbf{x}}) + \ell_{\mathbf{x}}(D_{\mathbf{s},\mathbf{x}}) \tag{4.7}$$

4.4.2 Translation FAT-MLM

To support multimodal crosslingual tasks such as speech translation, We propose Translation FAT-MLM which extends Monolingual FAT-MLM by using additional target language translation of the source language transcription as input. Formally Translation FAT-MLM takes $D_{\mathbf{s},\mathbf{x},\mathbf{y}} = \{(\mathbf{s},\mathbf{x},\mathbf{y})\}$ as input, where $\mathbf{y} = [y_1, ..., y_{|y|}]$ denotes the sequence of target language translation. This kind of triplet input is very common in speech translation corpus.

As shown in Fig. 4.5(d), we incorporate source language embedding e_{src} and target language embedding e_{tgt} for different languages to show the language difference. Similar to Monolingual FAT-MLM, Translation FAT-MLM randomly masks the translation input $\hat{\mathbf{y}} \sim \text{Mask}_{\text{token}}(\mathbf{y}, \lambda)$ and concatenate it with another two embeddings:

$$\boldsymbol{h}_{\mathbf{s},\mathbf{x},\mathbf{y}} = [\boldsymbol{e}_{\hat{\mathbf{s}}} + \boldsymbol{e}_{\text{src}}; \hat{\mathbf{x}} + \boldsymbol{e}_{\text{src}}; \hat{\mathbf{y}} + \boldsymbol{e}_{\text{tgt}}]$$

[2] Similar with previous work using masked language model objective, this loss only takes the masked input into consideration.

Then we reconstruct masked input from concatenated embeddings $\boldsymbol{h}_{\mathbf{s},\mathbf{x},\mathbf{y}}$ via a Transformer encoder. The reconstruction loss for different masked input is:

$$\begin{aligned}
\ell_{\mathbf{s}}(D_{\mathbf{s},\mathbf{x},\mathbf{y}}) &= \textstyle\sum_{(\mathbf{s},\mathbf{x},\mathbf{y})\in D_{\mathbf{s},\mathbf{x},\mathbf{y}}} ||\mathbf{s} - g(f(\boldsymbol{h}_{\mathbf{s},\mathbf{x},\mathbf{y}})||_2^2 \\
\ell_{\mathbf{x}}(D_{\mathbf{s},\mathbf{x},\mathbf{y}}) &= -\textstyle\sum_{(\mathbf{s},\mathbf{x},\mathbf{y})\in D_{\mathbf{s},\mathbf{x},\mathbf{y}}} \log p(\mathbf{x} \mid \boldsymbol{h}_{\mathbf{s},\mathbf{x},\mathbf{y}}) \\
\ell_{\mathbf{y}}(D_{\mathbf{s},\mathbf{x},\mathbf{y}}) &= -\textstyle\sum_{(\mathbf{s},\mathbf{x},\mathbf{y})\in D_{\mathbf{s},\mathbf{x},\mathbf{y}}} \log p(\mathbf{y} \mid \boldsymbol{h}_{\mathbf{s},\mathbf{x},\mathbf{y}})
\end{aligned}$$

We sum these loss functions for the final loss function of Translation FAT-MLM:

$$\ell_{\text{FAT-MLM}}(D_{\mathbf{s},\mathbf{x},\mathbf{y}}) = \ell_{\mathbf{s}}(D_{\mathbf{s},\mathbf{x},\mathbf{y}}) + \ell_{\mathbf{x}}(D_{\mathbf{s},\mathbf{x},\mathbf{y}}) + \ell_{\mathbf{y}}(D_{\mathbf{s},\mathbf{x},\mathbf{y}})$$

To fully utilize the corpora for different tasks, FAT-MLM can take any combination of speech, transcription, translation triplets $D_{2^{\{\mathbf{s},\mathbf{x},\mathbf{y}\}}}$ as input.[3] Specifically, these combinations include speech only data $\{\mathbf{s}\}$, monolingual text data, $\{\mathbf{x}\}$ or $\{\mathbf{y}\}$, speech and transcription tuple $\{(\mathbf{s}, \mathbf{x})\}$ for speech recognition, transcription and translation tuple $\{(\mathbf{x}, \mathbf{y})\}$ for machine translation, speech and translation tuple $\{(\mathbf{s}, \mathbf{y})\}$ for direct speech translation and speech transcription translation triplets $\{(\mathbf{s}, \mathbf{x}, \mathbf{y})\}$. For different combinations of input, FAT-MLM encodes the full concatenation of their embeddings and recover the masked portion. The loss function is:

$$\ell_{\text{FAT-MLM}}(D_{2^{\{\mathbf{s},\mathbf{x},\mathbf{y}\}}}) = \ell_{\mathbf{s}}(D_{\mathbf{s}\star}) + \ell_{\mathbf{x}}(D_{\mathbf{x}\star}) + \ell_{\mathbf{y}}(D_{\mathbf{y}\star}) \tag{4.8}$$

where $D_{\mathbf{s}\star}$, $D_{\mathbf{x}\star}$, $D_{\mathbf{y}\star}$ means any input including speech, source language text and target language text respectively. Note that in this framework, we can denote MLM as $\ell_{\mathbf{x}}(D_{\mathbf{x}})$, TLM as $\ell_{\mathbf{x},\mathbf{y}}(D_{\mathbf{x},\mathbf{y}})$, MAM as $\ell_{\mathbf{s}}(\mathbf{s})$.

4.5 Improving Downstream Tasks

In this section, we present how to adapt FAT-MLM to speech translation and enable speech translation models to learn from speech recognition and text machine translation.

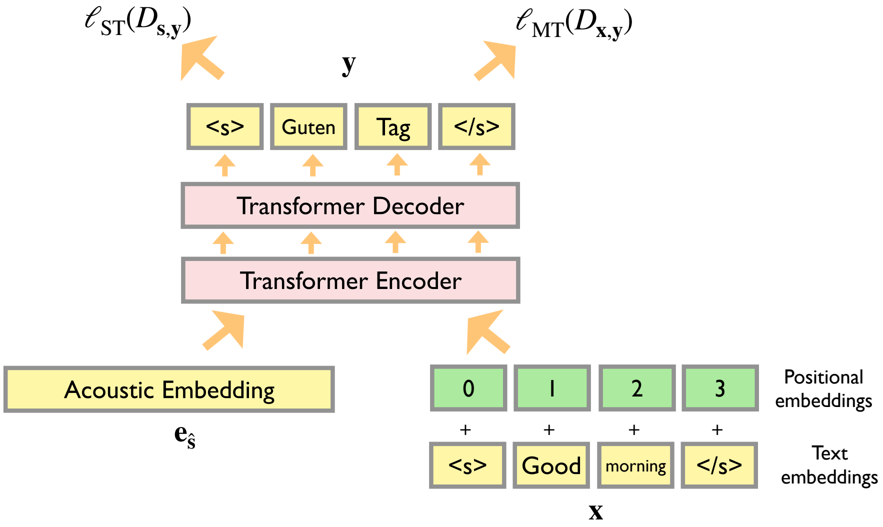


Figure 4.6: Fused Acoustic and Text-Speech Translation (FAT-ST).

4.5.1 From Text Translation to Speech Translation

Regardless of the particular design of different seq-to-seq models, the text machine translation encoder always takes the input sequence $\mathbf{x} = (x_1, ..., x_n)$ where each $x_i \in \mathbb{R}^{d_x}$ is a word embedding of d_x dimensions, and produces a new sequence of hidden states $\boldsymbol{h} = f(\mathbf{x}) = (h_1, ..., h_n)$. On the other hand, a decoder predicts the next output word y_t given the source sequence (actually its representation $\boldsymbol{h}$) and previously generated words, denoted $\mathbf{y}_{<t} = (y_1, ..., y_{t-1})$. The decoder stops when it emits `<eos>`, and the final hypothesis $\mathbf{y} = (y_1, ..., \texttt{<eos>})$ has probability

$$p(\mathbf{y} \mid \mathbf{x})_{\mathrm{MT}} = \prod_{t=1}^{|\mathbf{y}|} p(y_t \mid \mathbf{x},\, \mathbf{y}_{<t}) \tag{4.9}$$

At training time, we maximize the conditional probability of each ground-truth target sentence $\mathbf{y}^\star$ given input $\mathbf{x}$ over the whole training data $D_{\mathbf{x},\mathbf{y}}$, or equivalently minimizing the following loss:

$$\ell_{\mathrm{MT}}(D_{\mathbf{x},\mathbf{y}}) = -\sum_{(\mathbf{x},\mathbf{y})\in D_{\mathbf{x},\mathbf{y}}} \log p(\mathbf{y} \mid \mathbf{x}) \tag{4.10}$$

Different from text machine translation, speech translation takes speech features $\mathbf{s} = (s_1, ..., s_{|\mathbf{s}|})$ as input. Same as the speech input portion of FAT-MLM, these speech features are converted

[3] $2^{\{\mathbf{s},\mathbf{x},\mathbf{y}\}}$ is the power set of $\{\mathbf{s}, \mathbf{x}, \mathbf{y}\}$ triplets.

from the speech signals (e.g. spectrogram). Formally, the decoding and training of speech translation models can be defined as follows:

$$p(\mathbf{y} \mid \mathbf{s})_{\text{ST}} = \textstyle\prod_{t=1}^{|\mathbf{y}|} p(y_t \mid \mathbf{s},\, \mathbf{y}_{<t}) \tag{4.11}$$

$$\ell_{\text{ST}}(D_{\mathbf{s},\mathbf{y}}) = -\textstyle\sum_{(\mathbf{s},\mathbf{y})\in D_{\mathbf{s},\mathbf{y}}} \log p(\mathbf{y} \mid \mathbf{s}) \tag{4.12}$$

4.5.2 FAT-ST

To boost the performance of end-to-end speech translation, we propose to enable speech translation to encode both acoustic and text features as input by simply adapting the architecture of monolingual FAT-MLM to a Fused Acoustic and Text Speech Translation model (FAT-ST).

As shown in Fig. 4.6, FAT-ST's encoder shares identical architecture with monolingual FAT-MLM. In this way, we can simply encode either acoustic or text features by this encoder and the FAT-ST model can be optimized by speech translation loss ℓ_{ST}, machine translation loss ℓ_{MT} and FAT-MLM loss $\ell_{\text{FAT-MLM}}$. For a speech translation dataset $D_{\mathbf{s},\mathbf{x},\mathbf{y}}$, we decouple the triplets into three part $D_{\mathbf{s},\mathbf{y}}$ for ℓ_{ST}, $D_{\mathbf{s},\mathbf{x}}$ for $\ell_{\text{FAT-MLM}}$ and $D_{\mathbf{x},\mathbf{y}}$ for ℓ_{MT}. The loss function of FAT-ST is:

$$\begin{aligned}\ell_{\text{FAT-ST}}(D_{\mathbf{s},\mathbf{y}} \cup D_{\mathbf{s},\mathbf{x}} \cup D_{\mathbf{x},\mathbf{y}}) = {} & \ell_{\text{ST}}(D_{\mathbf{s},\mathbf{y}}) + \ell_{\text{MT}}(D_{\mathbf{x},\mathbf{y}}) \\ & + \ell_{\text{FAT-MLM}}(D_{\mathbf{s},\mathbf{x}})\end{aligned}$$

Please note that the speech recognition and machine translation data can either be included in speech translation data or additional datasets. Meanwhile, in practice, we find that CTC loss [38] is useful to improve the translation quality so that we include it in all the experiments.

4.5.3 Finetuning FAT-ST from Translation FAT-MLM

Similar to Lample and Conneau [56] we can further improve FAT-ST by finetuning from FAT-MLM. Since the FAT-ST decoder predicts text only, we initialize it from the acoustic and text shared Transformer encoder. Although Transformer decoder is unidirectional which is different from bidirectional FAT-MLM, it can still benefit from FAT-MLM in our experiments, This is also observed by Lample and Conneau [56] and Devlin et al. [30].

4.6 Experiments

We conducted speech translation experiments in 3 directions: English to German (En→De), English to Spanish (En→Es), and English to Dutch (En→Nl) to show the translation quality of baselines and our proposed methods.

4.6.1 Dataset

(a) Bilingual Dataset

Type	Name	En → De		En → Es		En → Nl	
		Hours	#Sent	Hours	#Sent	Hours	#Sent
$D_{\mathbf{s},\mathbf{x},\mathbf{y}}$	Must-C ST	408	226K	504	262K	442	245K
$D_{\mathbf{x},\mathbf{y}}$	Europarl MT	-	1.9M	-	2.0M	-	2.0M

(b) Monolingual Dataset

Type	Name	En		De	Es	Nl
		Hours	#Sent	#Sent	#Sent	#Sent
$D_{\mathbf{s},\mathbf{x}}$	Librispeech ASR	960	281K	-	-	-
$D_{\mathbf{s}}$	Libri-light Speech	3,748	579K	-	-	-
$D_{\mathbf{x}}/D_{\mathbf{y}}$	Europarl / Wiki Text	-	2.3M	2.1M	2.0M	2.3M

Table 4.1: Statistics of all datasets used in our experiments. Note that we use Europarl for En, De, Es monolingual text and Wiki Text for Nl because there is no monolingual Nl portion in Europarl. #Sent means the number of sentences.

We use 5 corpora with different modalities and languages: speech translation data $D_{\mathbf{s},\mathbf{x},\mathbf{y}}$ Must-C [31], speech recognition data $D_{\mathbf{s},\mathbf{x}}$ Librispeech [71], machine translation and monolingual text data $D_{\mathbf{x},\mathbf{y}}, D_{\mathbf{x}}, D_{\mathbf{y}}$ Europarl V7 [53], speech only data $D_{\mathbf{s}}$ Libri-Light (medium version) [51] and monolingual text data Wiki Text (only for Nl). The statistical results of the dataset are shown in Table. 4.1. We evaluate our models on Must-C dev and test set. Note that Must-C is collected based on spontaneous speeches (TED) which are very different from other audiobook speech dataset used in our experiments. Spontaneous speeches are much harder for speech translation than audiobook dataset such as Libri-trans [52]. That is one of the reasons why the translation accuracy of end-to-end speech translation is much worse than cascaded systems on Must-C than other speech translation corpus.

4.6.2 Training Detail

Pretrain Method	Models	En→De	En→Es	En→Nl	Avg.	Model Size
No Pretraining	ST	19.64	23.68	23.01	22.11	31.25M
	ST + ASR	21.70	26.83	25.44	24.66 (+2.55)	44.82M
	ST + ASR & MT	21.58	26.37	26.17	24.71 (+2.60)	56.81M
	ST + MAM	20.78	25.34	24.46	23.53 (+1.42)	33.15M
	ST + MAM + ASR	22.41	26.89	26.49	25.26 (+3.15)	46.72M
	Liu et al. [61]	22.55	-	-	-	-
	Le et al. [57]	23.63	28.12	27.55	26.43 (+4.32)	51.20M
	Cascade§	23.65	28.68	27.91	26.75 (+4.64)	83.79M
	FAT-ST (base).	22.70	27.86	27.03	25.86 (+3.75)	39.34M
ASR & MT	ST	21.95	26.83	26.03	24.94 (+2.83)	31.25M
	ST + ASR & MT	22.05	26.95	26.15	25.05 (+2.94)	56.81M
MAM	FAT-ST (base)	22.29	27.21	26.26	25.25 (+3.14)	39.34M
FAT-MLM	FAT-ST (base)	**23.68**	28.61	**27.84**	26.71 (+4.60)	39.34M
	FAT-ST (big)	23.64	**29.00**	27.64	**26.76** (+4.65)	58.25M

Table 4.2: BLEU comparisons on Must-C test set between our proposed methods and other baselines over 3 translation directions using MuST-C ($D_{\mathbf{s},\mathbf{x},\mathbf{y}}$) only (including pretraining methods). § are reported in Inaguma et al. [46].

Raw audio files are processed by Kaldi [74] to extract 80-dimensional log-Mel filterbanks stacked with 3-dimensional pitch features using a window size of 25 ms and step size of 10 ms. We train sentencepiece [54] models with a joint vocabulary size of 8K for text in each dataset. Training samples that have more than 3000 frames have been ignored for GPU efficiency. Our basic Transformer-based E2E-ST framework has similar settings with ESPnet-ST[46]. the speech input is first down-sampled the speech input with 2 layers of 2D convolution of size 3 with stride size of 2. Then there is a standard 12-layers Transformer with feed-forward layer of 2048 hidden size to bridge the source and target side. We only use 4 attention heads on each side of the transformer and each of them has a dimensionality of 256. We also show the results of FAT-ST big model with 4096 hidden size for feed-forward layers of all transformer layer. For speech reconstruction module, we simply linearly project the outputs of the Transformer encoder to another latent space, then upsample the latent representation with 2-layers deconvolution to match the size of the original input signal. We choose 30% for the random

Pretrain Data	Pretrain Method	Train Data	Models	En→De	En→Es	En→Nl	Avg.
$\emptyset$		$D_{\mathbf{s},\mathbf{x},\mathbf{y}}$	ST	19.64	23.68	23.01	22.11
			Cascade[§]	23.65	28.68	27.91	26.75 (+4.64)
$D_{\mathbf{s},\mathbf{x},\mathbf{y}} \cup D_{\mathbf{s},\mathbf{x}} \cup D_{\mathbf{x},\mathbf{y}}$	ASR & MT		ST	22.20	27.16	26.15	25.17 (+3.06)
			ST + ASR & MT	22.73	27.99	27.12	25.95 (+3.84)
	FAT-MLM		FAT-ST (base)	23.98	28.95	28.08	27.00 (+4.89)
			FAT-ST (big)	24.34	29.41	28.86	27.54 (+5.43)
$D_{\mathbf{s},\mathbf{x},\mathbf{y}} \cup D_{\mathbf{s},\mathbf{x}} \cup D_{\mathbf{x},\mathbf{y}}$ $\cup D_{\mathbf{s}} \cup D_{\mathbf{x}} \cup D_{\mathbf{y}}$	FAT-MLM		FAT-ST (base)	24.02	29.25	28.28	27.18 (+5.07)
			FAT-ST (big)	24.58	30.10	29.36	28.01 (+5.90)
$D_{\mathbf{s},\mathbf{x},\mathbf{y}} \cup D_{\mathbf{s},\mathbf{x}} \cup D_{\mathbf{x},\mathbf{y}}$ $\cup D_{\mathbf{s}} \cup D_{\mathbf{x}} \cup D_{\mathbf{y}}$	FAT-MLM	$D_{\mathbf{s},\mathbf{x},\mathbf{y}}$ $D_{\mathbf{s},\mathbf{x}} \cup D_{\mathbf{x},\mathbf{y}}$	FAT-ST (base)	23.91	29.01	28.18	27.03 (+4.92)
			FAT-ST (big)	**25.47**	**30.75**	**30.08**	**28.77 (+6.66)**
$\emptyset$		$D_{\mathbf{s},\mathbf{x},\mathbf{y}}$ + $D'_{\mathbf{s},\mathbf{y}}$	Pino et al. [73]	25.2	-	-	-

Table 4.3: BLEU comparisons on Must-C test set between our proposed methods using additional data. $D_{\mathbf{s},\mathbf{x}}$: Librispeech, $D_{\mathbf{x},\mathbf{y}}$: Europarl MT, $D_{\mathbf{s}}$: libri-light, $D_{\mathbf{x}}, D_{\mathbf{y}}$: monolingual data from Europarl or Wiki Text. [§] are reported in Inaguma et al. [46]. Pino et al. [73] use extra $D'_{\mathbf{s},\mathbf{y}}$ which includes Librispeech ($D_{\mathbf{s},\mathbf{x}}$) and 35,217 hour version of Libri-light speech data (almost $10\times$ of our $D_{\mathbf{s}}$) paired with their corresponding pseudo-translations generated by ASR and MT models. Their model size is 435.0M.

masking ratio λ across all the experiments including pretraining. During inference, we do not perform any masking over the speech input. We average the last 5 checkpoints for testing. For decoding, we use a beam search with beam-size 5 and length penalty 0.6 for German, 0.0 for Spanish and 0.3 for Dutch.

4.6.3 Translation Quality Comparisons

We showcase the translation accuracy of FAT-ST comparing against to the baselines in Table 4.2 and Table 4.3:

- **ST**: this is the vanilla speech translation system which does not use transcriptions.

- **ST + ASR MTL**: ST model with an additional ASR decoder and is trained with ASR multi-task learning using the transcriptions.

- **ST + ASR & MT MTL**: ST model with an additional ASR decoder and a MT encoder. It is trained with ASR and MT multi-task learning.

- **ST + MAM**: ST trained with additional MAM loss [20] which can be formalized as $\ell_{\mathbf{s}}(D_{\mathbf{s}})$

Model	# Parameters
MAM	23.69 M
FAT-MLM (base)	25.76 M
FAT-MLM (big)	38.36 M

Table 4.4: Models sizes of different models.

(See Fig. 4.3).

- **ST + MAM + ASR MTL**: ST trained with MAM loss and ASR multi-task learning.
- **Liu et al. [61]**: An end-to-end ST system with a multimodal encoder.
- **Le et al. [57]**: The state-of-the-art end-to-end ST model with an extra ASR decoder.
- **Cascade**: cascaded model which first transcribes the speech into transcription then passes the results to a machines translation system.
- **ST + ASR & MT pretraining**: the encoder of ST is initialized by a pretrained ASR encoder and decoder initialized by a pretrained MT decoder
- **Pino et al. [73]**: They propose to leverage additional speech data by generating pseudo-translations using a cascaded or an end-to-end speech translation model.

4.6.3.1 Model Size of Pretraining Models

Table 4.4 shows the number of parameters of different pretraining models. We can see that our FAT-MLM base model is a little bit larger than the MAM pretraining model, and the FAT-MLM big model is much larger than the base model.

4.6.3.2 Training with $D_{\mathbf{s},\mathbf{x},\mathbf{y}}$

In Table 4.2, with no pretraining, we can see that our proposed FAT-ST base model achieves the best results except Le et al. [57] and the cascaded model. However, our base model has much less parameters than both of them. Models with ASR or MT MTL and Liu et al. [61] all use the transcription data in Must-C dataset but show worse performance, thus our model can use transcription data more efficiently. Similar to other open source ST implementation results on Must-C [4], our implementation of ST + ASR & MT MTL is worse than ST + ASR.

We also compare the performance of models pretrained from different pretraining models. With pretrained on Must-C, FAT-ST (base) is improved by 0.85 BLEU by being finetuned from FAT-MLM, while it's performance drops by finetuning from MAM. Meanwhile, our proposed methods achieve much better performance compared with ASR & MT pretraining baselines. We also note that our FAT-ST base model for the first time achieves similar performances compared with Cascade baselines in these three translation directions of Must-C, while comparing with the cascaded model, our our base model is much smaller in size and faster in inference (see Fig. 4.7).

4.6.3.3 Pretraining with Additional Data

Table 4.3 shows that FAT-MLM can further improve FAT-ST by simply adding speech recognition data $D_{\mathbf{s},\mathbf{x}}$ (Librispeech) text machine translation data $D_{\mathbf{x},\mathbf{y}}$ (Europarl) and even speech only data $D_{\mathbf{s}}$ (Libri-light) and monolingual text data $D_{\mathbf{x}} \cup D_{\mathbf{y}}$. This shows good representation learning ability of our proposed FAT-MLM models. We can see that using larger data, the performance of our big model is increased much faster than the base model. That's because the number of parameters of the base model is too limited to learn from such big data.

4.6.3.4 Finetuning with Additional Data

The last part of Table 4.2 show that FAT-ST can be improved by learning from extra speech recognition and machine translation data. This is promising because speech translation data is very limited compared with much more abundant speech recognition and machine translation data. Different from Pino et al. [73] who propose to leverage additional speech data by generating pseudo-translations, our method doesn't use any pseudo-labels. Our best model outperforms their result on En→De by using much $7\times$ smaller model size and almost $10\times$ smaller speech data.

[4] ESPnet: https://github.com/espnet/espnet

Train Data	Pretrain Data	Models	→De	→Es	→Nl
$D_{\mathbf{s},\mathbf{x},\mathbf{y}}$	No pretraining	MT§	27.63	32.61	32.08
		FAT-ST (base)	24.41	30.81	29.18
	$D_{\mathbf{s},\mathbf{x},\mathbf{y}}$	FAT-ST (base)	27.24	31.98	31.27
		FAT-ST (big)	26.92	32.29	31.48
	$D_{\mathbf{s},\mathbf{x},\mathbf{y}}$ $\cup D_{\mathbf{s},\mathbf{x}} \cup D_{\mathbf{x},\mathbf{y}}$	FAT-ST (base)	27.43	32.38	32.44
		FAT-ST (big)	27.60	32.95	32.37
	$D_{\mathbf{s},\mathbf{x},\mathbf{y}} \cup D_{\mathbf{s},\mathbf{x}} \cup D_{\mathbf{x},\mathbf{y}}$ $\cup D_{\mathbf{s}} \cup D_{\mathbf{x}} \cup D_{\mathbf{y}}$	FAT-ST (base)	27.63	32.75	32.52
		FAT-ST (big)	28.13	33.39	32.72
$D_{\mathbf{s},\mathbf{x},\mathbf{y}}$ $\cup D_{\mathbf{s},\mathbf{x}} \cup D_{\mathbf{x},\mathbf{y}}$	$D_{\mathbf{s},\mathbf{x},\mathbf{y}} \cup D_{\mathbf{s},\mathbf{x}} \cup D_{\mathbf{x},\mathbf{y}}$ $\cup D_{\mathbf{s}} \cup D_{\mathbf{x}} \cup D_{\mathbf{y}}$	FAT-ST (base)	27.89	32.96	32.43
		FAT-ST (big)	28.80	34.28	34.22

Table 4.5: Comparisons of the auxiliary MT task between MT baselines and our proposed methods. § are reported in Inaguma et al. [46].

Model	En→De
FAT-ST with FAT-MLM (base)	23.68
- FAT-MLM decoder init.	23.20
- FAT-MLM encoder init.	22.70
- CTC loss	22.30
- Hierarchical Transformer	22.07
- FAT-MLM loss	20.64
- MT loss	19.64

Table 4.6: Ablation study. Here, hierarchical transformer means the model only shares the 6 layers of the transformer encoder for acoustic feature input and text feature input.

4.6.3.5 Performance of Auxiliary MT Task

Table 4.5 shows the translation quality of auxiliary MT task of FAT-ST. Although our models trained with Must-C are worse than the MT baseline, by using FAT-MLM trained with more data, our proposed methods can easily outperform the MT baseline. Note that these models' parameters are tuned to optimize speech translation task and MT is just an auxiliary task.

Speech transcription	those are their expectations of who you are not yours
Target reference	那 是 他们 所期望的 你的 样子 而不是 你自己的 期望 *that is they expected your appearance not yourself expectation*
Cascade-ASR	those are **there** expectations **to do** you are not yours
Cascade-Translation	那些 都是 希望 做到的， 你 不是 你的 。 *those are expect achievement you not yours*
FAT-ST	这些 是 他们 对 你的 期 望， 而不是 你的 期望 。 *these are they to your expectation not your expectation*

Table 4.7: English-to-Chinese speech translation example. The cascaded system is our implementation using the TED training data. The errors of cascaded model is highlighted in red.

Models	En→Zh
KD [59]	19.55
LUT [32]	20.84
COSTT [33]	21.12
Cascade [32]	21.36
ST*	22.07
FAT-ST	23.73
FAT-MLM + FAT-ST	25.49

Table 4.8: BLEU comparisons on English-to-Chinese speech translation. * is our implementation. Cascaded model is implemented by Dong et al. [32].

4.6.3.6 Ablation Study

Table 4.6 shows an ablation study of our proposed method. we can see that all the components contribute to the final performance.

4.6.3.7 English→Chinese Speech Translation

We also compare several models in TED English→Chinese speech translation task [59] with 524 hours speech in training set, 1.5 hours validation set (dev2010) and 2.5 hours test set (tst2015). We follow our previous experiments to preprocess the data. Same with previous work, we evaluate the performance with character-level BLEU. Table 4.8 shows that our proposed model can

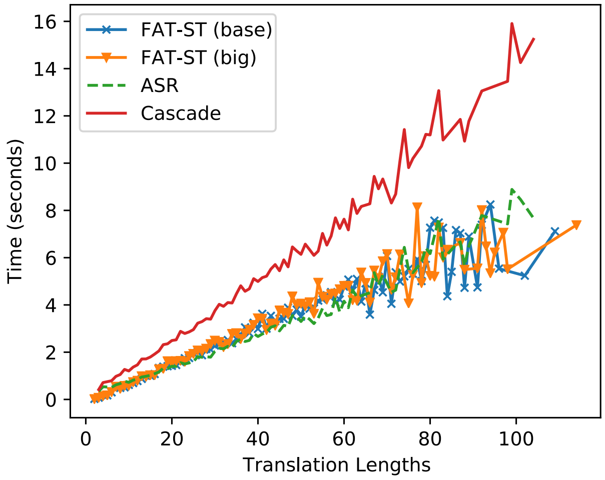


Figure 4.7: Decoding time comparison between Cascaded model (including its ASR) and FAT-ST.

largely outperform other baselines. Table 4.7 shows one example in this dataset. The translation of the cascaded model is wrong because of the errors in the its ASR (their→their, of who→ to do), while our FAT-ST produces the right translation.

4.6.3.8 Decoding Speed

Fig. 4.7 shows the decoding speed comparison between the Cascade model and our proposed FAT-ST. Our proposed FAT-ST model is almost 2× faster than the Cascade system which needs to wait for the speech recognition module to finish before starting to translate. The decoding time of FAT-ST (big) is almost the same as FAT-ST (base) because we only increase the feedforward network in Transformers.

4.7 Analysis on Pretrained models

To demonstrate the effectiveness of our proposed method, we designed several visualization analysis.

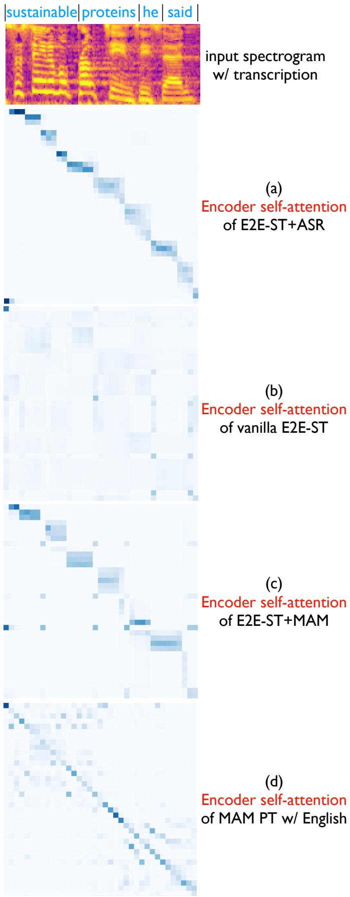


Figure 4.8: One head of the last layer self-attention comparison between different models. ASR MTL and MAM help the encoder learns similar self-attentions.

4.7.1 Attention Visualization

Compared with other tasks, e.g., MT or ASR, which also employ Seq2Seq framework for E2E training, E2E-ST is a more difficult and challenging task in many ways. Firstly, data modalities are different on the source and target sides. For ST, the encoder deals with speech signals and tries to learn word presentations on the decoder side, while MT has text format on both sides. Secondly, due to the nature of the high sampling rate of speech signals, speech inputs are generally multiple (e.g. 4 to 7) times longer than the target sequence, which increases the difficulties of learning the correspondence between source and target. Thirdly, compared with the monotonicity natural of the alignment of ASR, ST usually needs to learn the global reordering between speech signal and translation, and this raises the difficulties to another level. Especially in ST, since source and target are in different languages, it is very challenging to obtain the corresponding phoneme or syllable segments given the training signal from a different language.

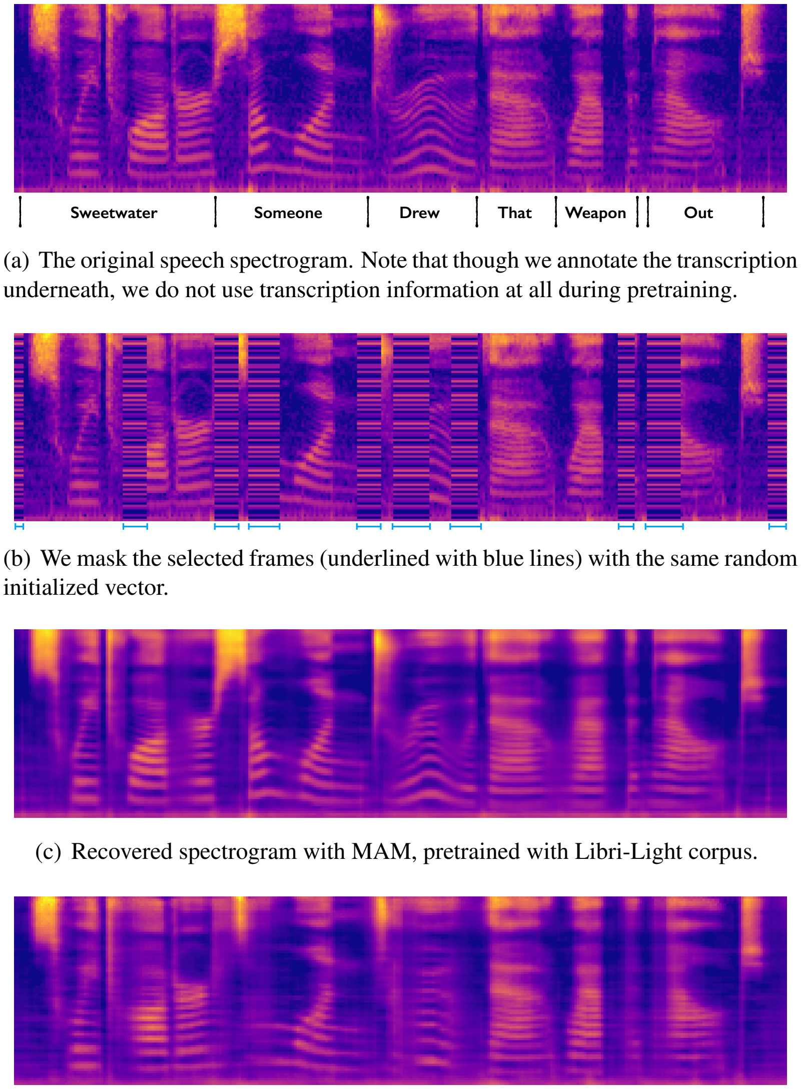
Fig. 4.8 tries to explain and analyze the difference between E2E-ST (a) and E2E-ST with ASR MTL (b). We extract the most top layer from the encoder for comparison. We notice that E2E-ST (a) tends to get more meaningful self-attention on the encoder with the training signal from ASR. With help from ASR, the source input spectrogram is chunked into segments that contain phoneme-level information. During training, the monotonicity natural of the ASR alignment functions as a forced alignment to group a set of adjacent frames to represent certain phonemes or syllables from source speech. With a larger scale of segmented spectrograms, the target side decoder only needs to perform reordering on those segments instead of frames. Our observations also align with the analysis from Stoian et al. [85].

We also visualize the self-attention on encoder for E2E-ST with MAM (without pretraining) in (c) of Fig. 4.8. We find that MAM has the similar ability with ASR to segment the source speech into chunks. As it is shown in (d) of Fig. 4.8, when we only perform pretraining on the English speech (Libri-Light dataset), without E2E-ST training, self-attentions that are generated by pretrained MAM are mostly monotonic on source side. Recovering local frames usually needs the information from surrounding context, especially for the speaker and environment-related characteristic. But we still observe that self-attention sometimes focuses on longer distance frames as well. This type of attention is very similar with low to mid layer self-attention of ASR. When there is a down streaming task (e.g., ASR or ST) is used for fine tuning, the top layer's self-attention will get chunked attention which is similar to (a) and (c).

To conclude, we observe that MAM functions very similar to ASR on the encoder side. Hence, MAM is a reliable framework that can be used as an alternative solution when there is no transcription available. Especially, with the help of a large scale acoustic dataset, which does not have transcription annotation, MAM provides the E2E-ST a much better encoder initialization.

4.7.2 Reconstruction Evaluation

To demonstrate what MAM has learned from pretraining step, we first showcase the reconstruction ability of MAM by visualizing the differences of spectrograms between the original and recovered inputs. This experiment was conducted on two corpora, Libri-Light and the Free



(a) The original speech spectrogram. Note that though we annotate the transcription underneath, we do not use transcription information at all during pretraining.

(b) We mask the selected frames (underlined with blue lines) with the same random initialized vector.

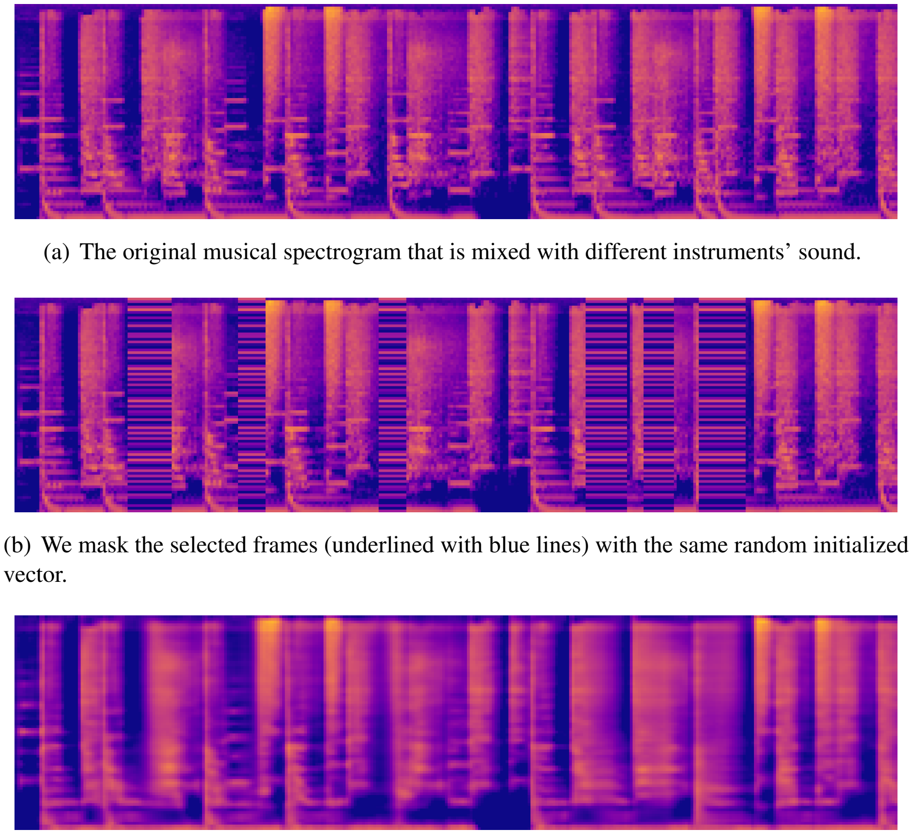
(c) Recovered spectrogram with MAM, pretrained with Libri-Light corpus.

(d) MAM that pretrains with FMA music corpus still have the ability to reconstruct corrupted speech signal.

Figure 4.9: One speech example to showcase the reconstruction ability of pretrained MAM. We notice that MAM reconstructs the corrupted audio signal in both pretraining with ordinary speech and music dataset.

Music Archive (FMA) [29] dataset. We use the “fma-medium” setting [5] which contains about

[5] https://github.com/mdeff/fma



(a) The original musical spectrogram that is mixed with different instruments' sound.

(b) We mask the selected frames (underlined with blue lines) with the same random initialized vector.

(c) Recovered spectrogram with MAM, pretrained with Libri-Light corpus.

Figure 4.10: One speech example to showcase the reconstruction ability of pretrained MAM. pretrained MAM with Libri-Light corpus (only human speech data) can not reconstruct the original music spectrogram accurately since there are many different musical instruments' sound that is unseen in speech data.

25,000 tracks of 30 seconds music within 16 unbalanced genres. The total music length is about 208 hours. We use FMA dataset for reconstruction visualization since FMA only contains music data and the characteristic of the music signal is very different from pure human speech. Note that our reconstructed spectrograms are a little blur compared with the original input since there are some downsampling steps in the E2E-ST baseline framework.

To verify the pretrained results of MAM, we demonstrate the reconstruction ability of MAM by visualizing the results in Fig. 4.9. We show the original spectrogram of input speech in

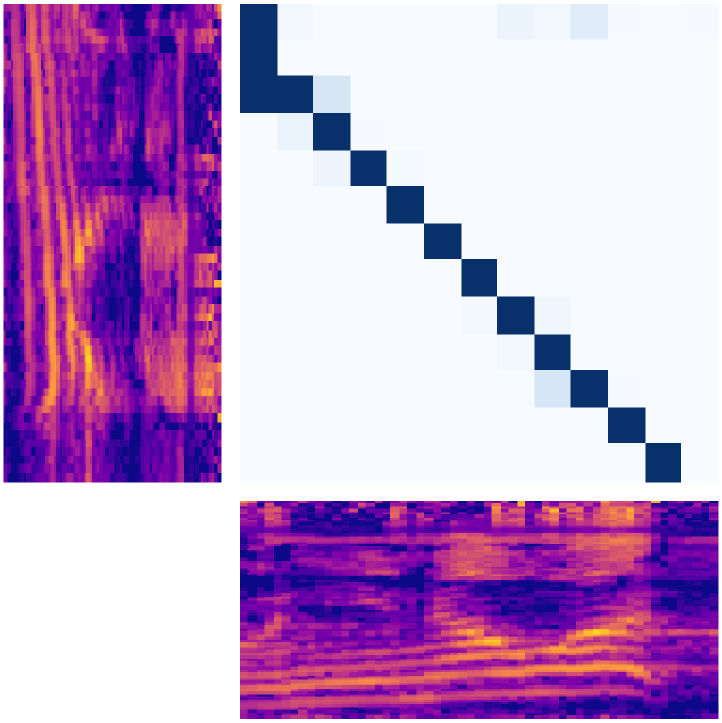


Figure 4.11: One speech self-attention head's output at the first transformer layer in acoustic embedding module and its corresponding spectrogram. This is a Translation FAT-MLM model trained with Must-C En→De dataset.

Fig. 4.9(a). Then we corrupted the original spectrogram by replacing the selected mask frames with ϵ, which is a random initialized vector, to form $\hat{\mathbf{x}}$ (see Fig. 4.9(b)). In Fig. 4.9(c), we show that our proposed MAM is able to recover the missing segments of input speech by pretraining over the Libri-Light dataset. More interestingly, since MAM does not need any transcription to perform pretraining, we also pretrain MAM with FMA dataset. Surprisingly, as shown in Fig. 4.9(d), MAM performs very similar reconstruction ability compared with the one that is pretrained with speech dataset considering the corrupted audio is only about pure speech. This might be because some music tracks include human singing voices and MAM learns human speech characteristics from those samples though human singing voice can be quite different from speech.

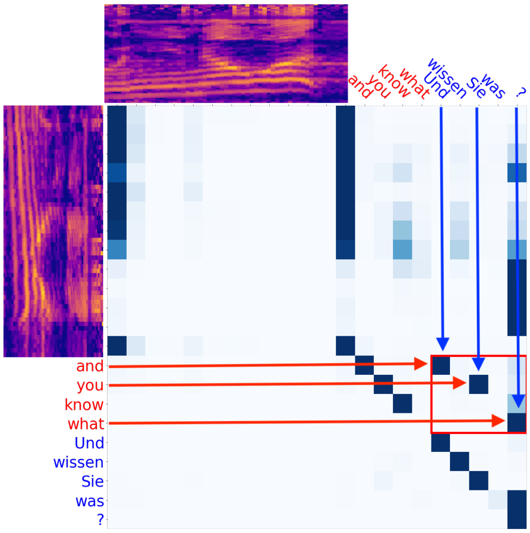
In the other way around, we also try to use Libri-Light pretrained MAM to recover the corrupted music in Fig. 4.10. MAM that pretrained with human speech data does not show good reconstruction in Fig. 4.10(c) since there are many different musical instruments' sounds that are unseen in speech data.

4.7.3 Cross-Modal Cross-Lingual Alignment

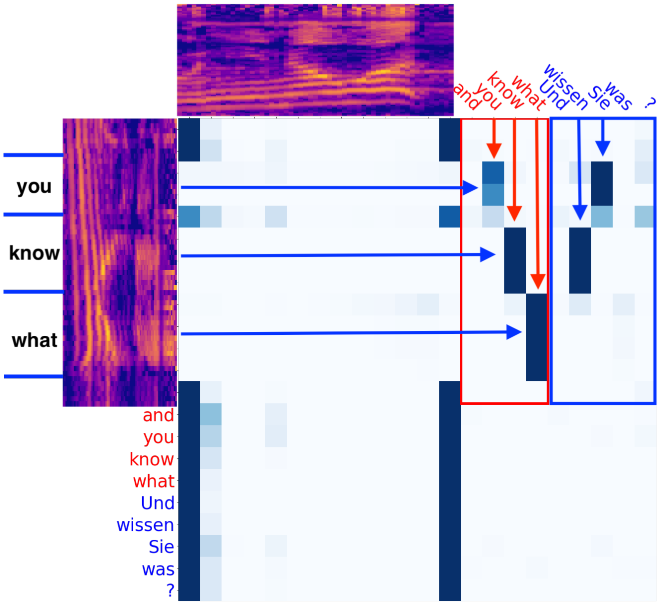
To demonstrate FAT-MLM's ability to unify the representation of different modality and language, we show the self-attention layers of a translation FAT-MLM in Fig. 4.11 and 4.12. The clear monotonic attention in Fig. 4.11 shows that our proposed method can learn good representation for speech [20]. Fig. 4.12(a) shows the self-attention on concatenated input in two different modalities. that FAT-MLM can learn a good crosslingual alignment between two languages, such as "and" to "Und" and "you" to "Sie". Fig. 4.12(b) shows that FAT-MLM is able to learn a clear monotonic speech-to-text crossmodal attention like many speech recognition models.

4.8 Summary

In this chapter, we first present a novel acoustic modeling framework Masked Acoustic Model (MAM). MAM not only can be used as an extra component during training time, but also can be used as a separate pretraining framework with arbitrary acoustic signal. Then, we further extend MAM to Fused Acoustic and Text Masked Language Model (FAT-MLM) which learns a unified representation for text and speech from any data that combines speech and text. We extend this framework to a sequence-to-sequence speech translation model which enables learning from speech recognition and text-based machine translation data at the first time. Our results show significant improvement on three translation directions of the Must-C dataset and outperform the cascaded baseline.



(a) This self-attention head shows bilingual alignment between “`and`‘ and “`Und`”, “`you`‘ and “`Sie`”, “`what`” and “`?`” in transcription and translation respectively.



(b) Left side spectrogram shows gold speech-transcription alignment. This self-attention head shows monotonic crossmodal attention in red box. Meanwhile, the speech-to-translation attention (in blue box) clearly show the alignment between “`you`‘ and “`Sie`”, “`know`” and “`wissen`” in speech and translation respectively. Note that in this speech, the pronounciation of “`and`” is very weak.

Figure 4.12: Two self-attention heads’ output at the first layer of acoustic and text shared transformerfrom a Translation FAT-MLM model trained with Must-C En→De dataset, annotated with corresponding spectrogram, transcription (red) and translation (blue).

Chapter 5: Cloning the Voice for Translation with Multimodal Pretrained Model

5.1 Motivation

Synthesizing generated translation text into speech in the target language is beneficial for breaking down communication barriers in case the users are not comfortable read it. However, the current conventional Text-to-Speech (TTS) system, which usually synthesizes speech based on a given text, does not have the ability to vary based on another given source speech.

In Chapter 4, we introduce a speech representation learning framework FAT. However, it focuses on *speech understanding* tasks which take speech as input, but for the inverse direction, *speech synthesis*, which synthesis speech as output, the potential of representation learning is yet to be realized. MAM and FAT-MLM show that reconstructing masked spectrogram with continuous units can improve speech-to-text translation. The quality of their proposed speech reconstruction is far from the requirement of speech synthesis tasks. (see Fig. 4.9)

To address this problem, we extend FAT to a new framework, Alignment-Aware Acoustic-Text Pretraining (A^{3}T), where we introduce cross-modal alignment embeddings which make the model easier to learn the alignment between the acoustic and phoneme input during multi-modal pretraining, and significantly improve the quality of the reconstructed acoustic signals. Different from the segment embeddings used in Segatron and Moreover, we borrow several useful ideas from recent text-to-speech (TTS) literature, including Conformer [40, 41] and Post-Net [79], to further improve the quality of our reconstructed spectrograms.

Without any finetuning, the proposed model can be adopted as a speech-editing system, a task that modifies an existing speech, by reconstructing the desired acoustic signals given original contextual speech and modified text. Furthermore, with training on corpora in different languages, the model can be adopted as a cross-lingual multi-speaker TTS system with our proposed prompt-based decoding method, to clone the voice of the given speech and generate the speech in another language.

5.2 Alignment-Aware Acoustic-Text Pretraining

Although existing speech pretraining models show a strong representation learning ability and significantly improve upon many down-stream tasks in *speech understanding*, all these efforts can not support *speech synthesis* tasks. To address this problem, we propose the Alignment-Aware Acoustic-Text Pretraining (A^{3}T) which learns to generate high-quality spectrogram given speech context and text.

5.2.1 A^{3}T

A^{3}T takes speech and transcription tuples as input, denotes as $D_{\mathbf{s},\mathbf{x}} = \{\langle \mathbf{s}, \mathbf{x} \rangle^{(n)}\}_{n=1}^{|D|}$, where $\mathbf{s} = (s_1, ..., s_{|s|})$ is a sequence of acoustic features $s_i \in \mathbb{R}^{d_\text{s}}$ which can be the spectrogram or mel-spectrogram of the speech audio, and each s_i represents the frame-level speech feature, and $\mathbf{x} = (x_1, ..., x_{|\mathbf{x}|})$ is the sequence of corresponding transcription.

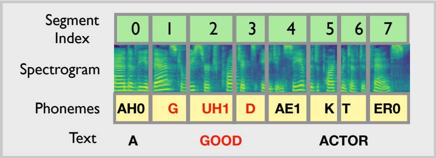
As shown in Fig. 5.1, we first randomly mask several spans of $\mathbf{s}$ by a random masking function over the input $\mathbf{s}$: $\hat{\mathbf{s}} \sim \text{Mask}_\text{span}(\mathbf{s}, \lambda)$, where $\text{Mask}_\text{span}(\cdot)$ replaces several random spans of $\mathbf{s}$ by the probability of λ with the same number of a random initialized masking vector $\epsilon_\mathbf{s} \in \mathbb{R}^{d_\text{s}}$. Then we encode $\hat{\mathbf{s}}$ with a acoustic encoder for acoustic embeddings $\boldsymbol{e}_{\hat{\mathbf{s}}}$. In this work, we use a nonlinear feed-forward layer as the acoustic encoder.

5.2.2 Cross-modal Alignment Embedding

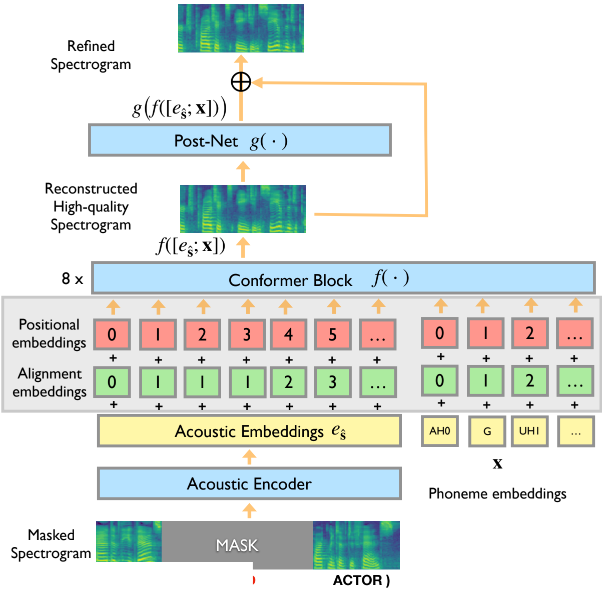
To strengthen the interaction between the speech and text input, we introduce cross-modal alignment embedding as one input of encoder, where we sum the ith acoustic embedding $\boldsymbol{e}_{s_i}$ or text embedding $\mathbf{x}_i$ with its positional embedding $\boldsymbol{e}_{\text{pos}_i}$ and alignment embedding $\boldsymbol{e}_{\text{aln}_i}$ all together: $\boldsymbol{e}_{s_i} + \boldsymbol{e}_{\text{pos}_i} + \boldsymbol{e}_{\text{aln}_i}$, where previous work have proved the embedding sum operation is simple and effective [30]. After that, the phoneme embedding and its acoustic embeddings will share the same alignment embedding. We use a forced aligner to pre-process the dataset to get the alignment information, which is shown in Fig. 5.1(a).

5.2.3 Conformer

Given the recent success of Convolution-augmented Transformer (Conformer) on various speech tasks [40, 41], we adopt Conformer as the backbone of our encoder and decoder. Compared



(a) Forced Alignment Preprocessing.

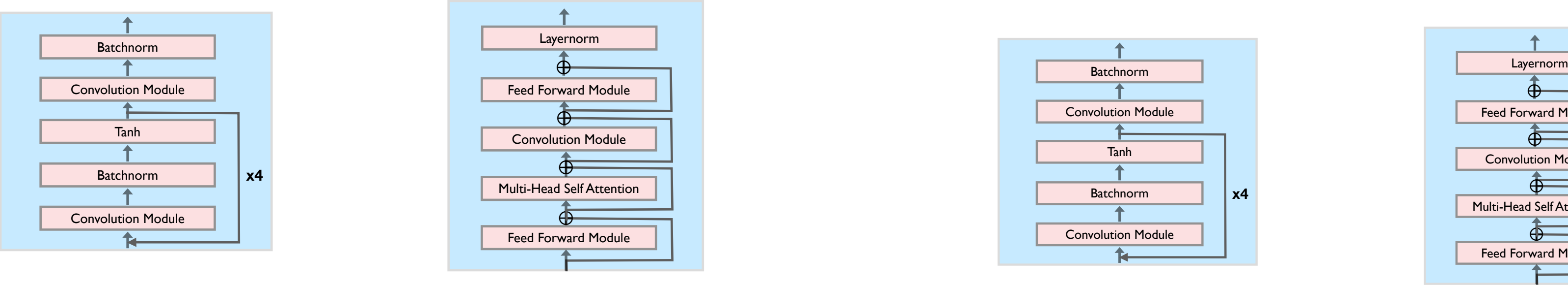


$\ell_{\mathbf{s}}(D_{\mathbf{s},\mathbf{x}})$

$g\big(f(e_{\hat{s}},\mathbf{x})\big)+f(e_{\hat{s}},\mathbf{x})$

embeddings a

ed by forced



(c) Conformer Block.

(d) Post-Net.

Figure 5.1: Alignment-Aware Acoustic-Text Pretraining (A^{3}T).

with Transformer, Conformer introduces a convolution module and an additional feedforward module, which is shown in Fig. 5.1(c). In our experiments, we find Conformer is better than Transformer for acoustic-text pretraining.

5.2.4 Post-Net and Loss Function

We follow Tacotron 2 [79] to use Post-Net to refine the generated spectrogram. The predicted spectorgram is passed through a 5-layer convolution Post-Net to be refined as shown in Fig. 5.1(d).

The training objective of multi-modal A^{3}T includes a speech reconstruction loss $\ell_{\mathbf{s}}(D_{\mathbf{s},\mathbf{x}})$ which takes a spectrogram $\mathbf{s}$ and a text sequence $\mathbf{x}$ as input. We have the following training objective to reconstruct the original speech signal with the surrounding context information:[1]

$$
\begin{aligned}
\ell_{\mathbf{s}}(D_{\mathbf{s},\mathbf{x}}) = \sum_{\langle \mathbf{s},\mathbf{x}\rangle \in D_{\mathbf{s},\mathbf{x}}} & \| \underbrace{f([e_{\hat{\mathbf{s}}};\mathbf{x}]) + g\big(f([e_{\hat{\mathbf{s}}};\mathbf{x}])\big)}_{\text{refined spectrogram}} - \mathbf{s}\|_1 \\
& + \| \underbrace{f([e_{\hat{\mathbf{s}}};\mathbf{x}])}_{\text{reconstructed spectrogram}} - \mathbf{s}\|_1
\end{aligned}
\tag{5.1}
$$

where g is a Post-Net which tries to recover a better original signal from encoded representation $f([e_{\hat{\mathbf{s}}};\hat{\mathbf{x}}])$. We use mean absolute error (MAE) for measuring the difference between s and the reconstructed spectrogram.

Similar with MAM and FAT, we can also apply text-level masking and reconstruction to boost the learning for cross-modal representation.

We use the cross-entropy loss for text reconstruction:

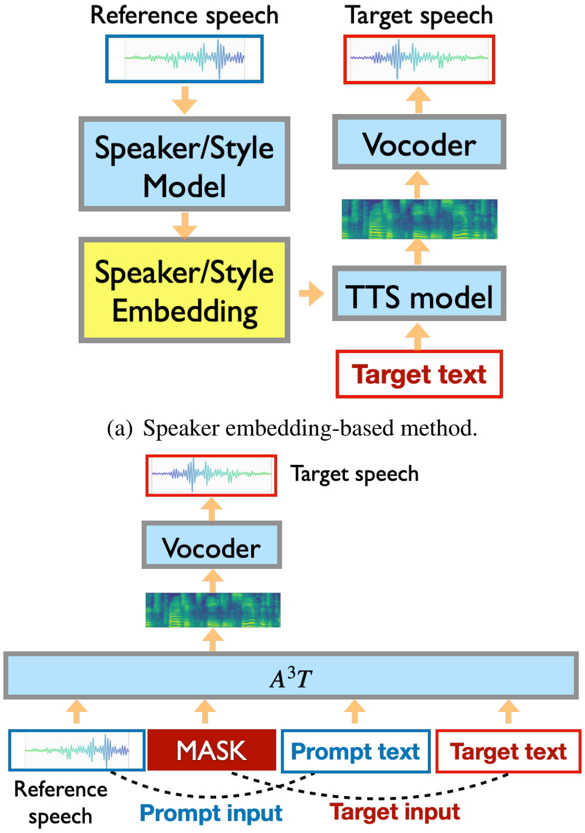
$$
\ell_{\mathrm{x}}(D_{\mathrm{s,x}}) = -\sum_{\langle \mathrm{s,x}\rangle \in D_{\mathrm{s,x}}} logp(\mathrm{x}|[e_{\hat{\mathrm{s}}};\mathrm{x}])
\tag{5.2}
$$

The final loss is the combination of the speech and text:

$$
\ell(D_{\mathrm{s,x}}) = \ell_{\mathrm{s}}(D_{\mathrm{s,x}}) + \ell_{\mathrm{x}}(D_{\mathrm{s,x}})
\tag{5.3}
$$

In this chapter, we use $\ell_{\mathrm{x}}(D_{\mathrm{s,x}})$ for cross lingual setting only.

[1] Similar with previous work using masked language model objective, this loss only takes the masked input into consideration.



(a) Speaker embedding-based method.

(b) Prompt-based decoding.

Figure 5.2: Illustrations for one-shot TTS. The prompt speech and text are wrapped with blue rectangles, and the target speech and text are wrapped with red.

5.2.5 Prompt-based decoding

Voice cloning is similar with the task of multi-speaker speaker TTS. Existing popular unseen speaker TTS models [48] are trained with seen speaker embeddings and generalizes to unseen speaker embeddings during the inference. However, such speaker embeddings are extracted from an external speaker verification model which is trained with tens of thousands of speakers.

In this work, we find our model can achieve comparable naturalness to models with speaker embeddings for unseen speaker TTS task; What's more, our generations are more similar to the unseen speaker's reference speech. The illustrations of how to synthesis speech for unseen speakers with our A^{3}T model are shown in Fig. 5.2, which is named prompt-based A^{3}T.

The key idea is to concatenate the prompt and the target together into a new utterance input, where the target speech is consist of n `[MASK]` and n is predicted by a duration predictor. By inputting the concatenated speech and text, A^{3}T model will predict the spectrogram of these masked frames. The role of the reference text and speech in our model is similar to prompts in language model [18], and hence we call it prompt-based decoding/generation. And we train our model on a union corpus with different languages. The prompt decoding is able to do cross-lingual multi-speaker TTS with given texts in two different languages.

5.3 Experiments

In this section, we introduce our experiments for spectrogram reconstruction pretraining task multi-speaker TTS and cross-lingual multi-speaker TTS. The results are evaluated with the MOS scores.

5.3.1 Datasets

For monolingual multi-speaker TTS, we conduct our model on LJSpeech [47] and VCTK [98]. And for cross-lingual voice cloning, we use VCTK [98] in English and AISHELL-3 [80] in Mandarin.

5.3.2 Configuration Detail

Raw audio files are processed with 50 ms frame size and 12.5 ms frame hop with the Hann window function to extract 80-dimensional log-Mel filterbanks. We use 24K sampling rate for VCTK and 22K for LJSpeech. The forced alignment and G2P are both carried out by HTK [101] to convert English words to phones and align phones with audio segments. For speech-editing systems and prompt-based TTS, we use the publicly available duration predictor from FastSpeech 2 implemented in ESPnet [46]. We use Parallel-WaveGAN [99] vocoder for all the systems.

All A^{3}T models pretrained in our experiments share the same architecture: 4 layers Conformer encoder, 4 layers Conformer decoder, and 5 layers Conv1d Post-Net, with 2 heads multi-head attention in 384-dim. The convolution kernel sizes of the encoder and decoder are 7 and 31, respectively. The shape of alignment embeddings is (500, 384), where we assume the number of phones will not exceed 500 for a single input. The shape of input phone embeddings is (73, 384), and we use a ReLU [1] nonlinear layer to transform 80-dim log-Mel filterbanks features to 384-dim.

For monolingual multi-speaker TTS, we train the model on LJSpeech and VCTK, and test it on VCTK. For cross-lingual multi-speaker TTS, we train our model on VCTK and AISHELL-3, and test it with 20 sample utterances.

5.3.3 Results

Model	Seen	Unseen
FastSpeech 2	3.33 ± 0.10	3.78 ± 0.10
+GST [93]	3.42 ± 0.10	3.81 ± 0.11
A^3T	3.61 ± 0.09	3.90 ± 0.10
Groundtruth	3.94 ± 0.08	4.09 ± 0.10

Table 5.1: The MOS evaluation (↑) for speaker similarity on multi-speaker TTS on VCTK with 95% confidence intervals. The FastSpeech2 model is equipped with X-vectors [82].

Model	Seen	Unseen
FastSpeech 2	3.34 ± 0.11	3.85 ± 0.11
+GST [93]	3.27 ± 0.11	3.72 ± 0.11
A^3T	3.63 ± 0.10	3.94 ± 0.11
Groundtruth	4.04 ± 0.08	4.05 ± 0.10

Table 5.2: The MOS evaluation (↑) for speech quality on multi-speaker TTS on VCTK with 95% confidence intervals. The FastSpeech2 model is equipped with X-vectors [82].

5.3.3.1 Monolingual Multi-speaker TTS

The quality of the generations and the speaker similarity between the generation and the reference are evaluated, and the results are shown in Tab. 5.1 and Tab. 5.2. From this table, we can see that the style embedding GST [93] improves the similarity scores but harms the quality scores, while our A^{3}T model is the most favorable system in both the speaker similarity and the speech quality. Strikingly, we observe that the average score of the Unseen cases is higher than the Seen, which is counterintuitive. However, when looking into the MOS of the Groundtruth, the gap is still there and we believe this is due to the difference between these two test case sets.

5.3.3.2 Cross-lingual Multi-speaker TTS

To evaluate the ability of cross-lingual synthesizing, we sampled 20 unseen speakers with 20 utterances. Each audio sample is listened by 10 subjects whose first language is Chinese and are well-educated in English. The subjects are asked to evaluate the quality and similarity of synthesized audio. The results are shown in Tab. 5.3 and Tab. 5.4. From these tables, we can see that even without speaker embedding (X-Vector) [82], Our model outperforms other baseline systems in terms of speaker similarity and speech quality.

Model	Unseen
Tacotron 2 + X-vectors + GST	3.33 ± 0.16
FastSpeech 2 + X-vectors + GST	3.49 ± 0.14
our work	3.58 ± 0.14

Table 5.3: The MOS for speech quality on cross-lingual multi-speaker TTS with 95% confidence intervals.

Model	Unseen
Tacotron 2 + X-vectors + GST	3.30 ± 0.17
FastSpeech 2 + X-vectors + GST	3.45 ± 0.16
our work	3.53 ± 0.11

Table 5.4: The MOS for speaker similarity on cross-lingual multi-speaker TTS with 95% confidence intervals.

5.4 Summary

In this chapter, we propose Alignment-Aware Acoustic-Text Pretraining (A^{3}T) which can reconstruct masked acoustic signals with high quality. We show that our proposed A^{3}T model has the improves multi-speaker speech synthesis in both monolingual and cross-lingual settings with our proposed prompt-based decoding.

Chapter 6: Conclusions

In this dissertation, we reviewed several works on simultaneous translation and speech translation. We proposed several works to conduct direct simultaneous speech translation and improve the translation quality with multi-modal pretrained model.

First, we introduce a simple but effective method to generate more monotonic pseudo references for simultaneous translation, which mitigate the reordering issues in the training data (Chapter 2). Pseudo references generated from the full-sentence trained model in test-time wait-k mode have fewer reorderings. And incorporating these pseudo references in generic wait-k training can reduce anticipations in training and avoid hallucination in testing. We also proposed two new metrics, anticipation rate and hallucination rate, to evaluate the bitext and trained model, respectively.

Then we investigate the method for direct simultaneous speech translation (Chapter 3). We apply a synchronized streaming ASR model to guide the decoding policy of the ST model. The direct translation model can successfully avoid error propagation problem compared with the cascade method. And our proposed method is more reliable than the pre-decision method and the trigger-based method on speech signals.

Next, to address the data scarcity problem for speech translation, we introduce a self-supervised frameworks Masked Acoustic Model (MAM) (Chapter 4). MAM can take the use of raw speech data to improve the ability of speech encoder. We also extend MAM to Fused Acoustic and Text Masked Language Model (FAT-MLM) which can be trained on any data combines speech and text that is benefit to speech translation, e.g., speech recognition data and machine translation data. Out experiment shows that this framework can learn a unified representation for text and speech. The proposed methods can greatly improve the translation quality. We also extend FAT to Alignment-Aware Acoustic-Text Pretraining (A^{3}T) by including alignment information (Chapter 5), and make it able to use for TTS downstream tasks, like voice cloning.

Bibliography

[1] Abien Fred Agarap. Deep learning using rectified linear units (relu). *arXiv preprint arXiv:1803.08375*, 2018.

[2] Antonios Anastasopoulos and David Chiang. Tied multitask learning for neural speech translation. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, 2018.

[3] Antonios Anastasopoulos, David Chiang, and Long Duong. An unsupervised probability model for speech-to-translation alignment of low-resource languages. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, 2016.

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

[5] Naveen Arivazhagan, Colin Cherry, Wolfgang Macherey, Chung-Cheng Chiu, Semih Yavuz, Ruoming Pang, Wei Li, and Colin Raffel. Monotonic infinite lookback attention for simultaneous machine translation. *Meeting of the Association for Computational Linguistics*, 2019.

[6] Naveen Arivazhagan, Colin Cherry, Isabelle Te, Wolfgang Macherey, Pallavi Baljekar, and George Foster. Re-translation strategies for long form, simultaneous, spoken language translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 7919–7923. IEEE, 2020.

[7] Alexei Baevski, Henry Zhou, Abdelrahman Mohamed, and Michael Auli. wav2vec 2.0: A framework for self-supervised learning of speech representations. *NeurIPS 2020*, 2020.

[8] D. Bahdanau, K. Cho, and Y. Bengio. Neural machine translation by jointly learning to align and translate. *ArXiv e-prints*, September 2014.

[9] Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019.

[10] Claudio Bendazzoli, Annalisa Sandrelli, et al. An approach to corpus-based interpreting studies: developing epic (european parliament interpreting corpus). *Proceedings of the Marie Curie Euroconferences MuTra: Challenges of Multidimensional Translation-Saarbrücken*, pages 2–6, 2005.

[11] Alexandre Berard, L. Besacier, A. Kocabiyikoglu, and Olivier Pietquin. End-to-end automatic speech translation of audiobooks. *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6224–6228, 2018.

[12] Alexandre Bérard, Laurent Besacier, Ali Can Kocabiyikoglu, and Olivier Pietquin. End-to-end automatic speech translation of audiobooks. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6224–6228. IEEE, 2018.

[13] Alexandra Birch, Phil Blunsom, and Miles Osborne. A quantitative analysis of reordering phenomena. In *Proceedings of the Fourth Workshop on Statistical Machine Translation*, pages 197–205. Association for Computational Linguistics, 2009.

[14] Steven Bird. A scalable method for preserving oral literature from small languages. In Gobinda Chowdhury, Chris Koo, and Jane Hunter, editors, *The Role of Digital Libraries in a Time of Global Change*. Springer Berlin Heidelberg, 2010.

[15] Steven Bird, Lauren Gawne, Katie Gelbart, and Isaac McAlister. Collecting bilingual audio in remote indigenous communities. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, 2014.

[16] Fabienne Braune, Anita Gojun, and Alexander Fraser. Long-distance reordering during search for hierarchical phrase-based smt. In *Proceedings of the Annual Conference of the European Association for Machine Translation (EAMT)*, pages 28–30. Citeseer, 2012.

[17] Peter F. Brown, Stephen A. Della Pietra, Vincent J. Della Pietra, and Robert L. Mercer. The mathematics of statistical machine translation: Parameter estimation. *Computational Linguistics*, 19(2):263–311, 1993. URL `https://www.aclweb.org/anthology/J93-2003`.

[18] Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. Language models are few-shot learners. 2020.

[19] Erik Camayd-Freixas. Cognitive theory of simultaneous interpreting and training. In *Proceedings of the 52nd Conference of the American Translators Association*, volume 13, 2011.

[20] Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. Mam: Masked acoustic modeling for end-to-end speech-to-text translation. *arXiv preprint arXiv:2010.11445*, 2020.

[21] Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. Direct simultaneous speech-to-text translation assisted by synchronized streaming asr. In *Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021*, pages 4618–4624, 2021.

[22] Junkun Chen, Mingbo Ma, Renjie Zheng, and Liang Huang. Specrec: An alternative solution for improving end-to-end speech-to-text translation via spectrogram reconstruction. 2021.

[23] Junkun Chen, Renjie Zheng, Atsuhito Kita, Mingbo Ma, and Liang Huang. Improving simultaneous translation by incorporating pseudo-references with fewer reorderings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5857–5864, 2021.

[24] Sanyuan Chen, Chengyi Wang, Zhengyang Chen, Yu Wu, Shujie Liu, Zhuo Chen, Jinyu Li, Naoyuki Kanda, Takuya Yoshioka, Xiong Xiao, et al. Wavlm: Large-scale self-supervised pre-training for full stack speech processing. *IEEE Journal of Selected Topics in Signal Processing*, 16(6):1505–1518, 2022.

[25] Yung-Sung Chuang, Chi-Liang Liu, Hung-Yi Lee, and Lin-shan Lee. SpeechBERT: An Audio-and-text Jointly Learned Language Model for End-to-end Spoken Question Answering. *arXiv e-prints*, 2019.

[26] Yung-Sung Chuang, Chi-Liang Liu, Hung-yi Lee, and Lin-shan Lee. Speechbert: An audio-and-text jointly learned language model for end-to-end spoken question answering. *arXiv preprint arXiv:1910.11559*, 2019.

[27] Michael Collins, Philipp Koehn, and Ivona Kučerová. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd annual meeting on association for computational linguistics*, pages 531–540. Association for Computational Linguistics, 2005.

[28] Fahim Dalvi, Nadir Durrani, Hassan Sajjad, and Stephan Vogel. 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)*, 2018.

[29] Michaël Defferrard, Kirell Benzi, Pierre Vandergheynst, and Xavier Bresson. Fma: A dataset for music analysis. *arXiv preprint arXiv:1612.01840*, 2016.

[30] J. Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL-HLT*, 2019.

[31] Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. MuST-C: a Multilingual Speech Translation Corpus. In *NAACL*, 2019.

[32] Qianqian Dong, Rong Ye, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. " listen, understand and translate": Triple supervision decouples end-to-end speech-to-text translation. *arXiv preprint arXiv:2009.09704*, 2020.

[33] Qianqian Dong, Mingxuan Wang, Hao Zhou, Shuang Xu, Bo Xu, and Lei Li. Consecutive decoding for speech-to-text translation. In *The Thirty-fifth AAAI Conference on Artificial Intelligence, AAAI*, 2021.

[34] Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. An attentional model for speech translation without transcription. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, 2016.

[35] Chris Dyer, Victor Chahuneau, and Noah A. Smith. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 644–648, Atlanta, Georgia, June 2013. Association for Computational Linguistics. URL `https://www.aclweb.org/anthology/N13-1073`.

[36] Ethnologue. Ethnologue (21st edition). URL `https://www.ethnologue.com/enterprise-faq/how-many-languages-world-are-unwritten-0`.

[37] Michel Galley and Christopher D Manning. A simple and effective hierarchical phrase reordering model. In *Proceedings of the 2008 Conference on Empirical Methods in Natural Language Processing*, pages 848–856, 2008.

[38] Alex Graves, Santiago Fernández, Faustino Gomez, and Jürgen Schmidhuber. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *Proceedings of the 23rd international conference on Machine learning*, pages 369–376, 2006.

[39] Jiatao Gu, Graham Neubig, Kyunghyun Cho, and Victor OK Li. 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, 2017.

[40] Anmol Gulati, James Qin, Chung-Cheng Chiu, Niki Parmar, Yu Zhang, Jiahui Yu, Wei Han, Shibo Wang, Zhengdong Zhang, Yonghui Wu, et al. Conformer: Convolution-augmented transformer for speech recognition. *arXiv preprint arXiv:2005.08100*, 2020.

[41] Pengcheng Guo, Florian Boyer, Xuankai Chang, Tomoki Hayashi, Yosuke Higuchi, Hirofumi Inaguma, Naoyuki Kamo, Chenda Li, Daniel Garcia-Romero, Jiatong Shi, et al. Recent developments on espnet toolkit boosted by conformer. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5874–5878. IEEE, 2021.

[42] He He, Alvin Grissom II, John Morgan, Jordan Boyd-Graber, and Hal Daumé III. Syntax-based rewriting for simultaneous machine translation. In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 55–64, 2015.

[43] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[44] Jeremy Howard and Sebastian Ruder. Universal language model fine-tuning for text classification. *arXiv preprint arXiv:1801.06146*, 2018.

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

[46] Hirofumi Inaguma, Shun Kiyono, Kevin Duh, Shigeki Karita, Nelson Enrique Yalta Soplin, Tomoki Hayashi, and Shinji Watanabe. Espnet-st: All-in-one speech translation toolkit. *arXiv preprint arXiv:2004.10234*, 2020.

[47] Keith Ito and Linda Johnson. The LJ Speech Dataset. `https://keithito.com/LJ-Speech-Dataset/`, 2017.

[48] Ye Jia, Yu Zhang, Ron Weiss, Quan Wang, Jonathan Shen, Fei Ren, Patrick Nguyen, Ruoming Pang, Ignacio Lopez Moreno, Yonghui Wu, et al. Transfer learning from speaker verification to multispeaker text-to-speech synthesis. *Advances in neural information processing systems*, 31, 2018.

[49] Ye Jia, Melvin Johnson, Wolfgang Macherey, Ron J Weiss, Yuan Cao, Chung-Cheng Chiu, Naveen Ari, Stella Laurenzo, and Yonghui Wu. 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, 2019.

[50] Mandar Joshi, Danqi Chen, Yinhan Liu, Daniel S Weld, Luke Zettlemoyer, and Omer Levy. Spanbert: Improving pre-training by representing and predicting spans. *Transactions of the Association for Computational Linguistics*, 8:64–77, 2020.

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

[52] Ali Can Kocabiyikoglu, Laurent Besacier, and Olivier Kraif. Augmenting librispeech with french translations: A multimodal corpus for direct speech translation evaluation. In *Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC 2018)*, 2018.

[53] Philipp Koehn. Europarl: A parallel corpus for statistical machine translation. In *MT summit*, volume 5, pages 79–86. Citeseer, 2005.

[54] Taku Kudo and John Richardson. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, Brussels, Belgium, November 2018. Association for Computational Linguistics.

[55] Taku Kudo, Kaoru Yamamoto, and Yuji Matsumoto. Applying conditional random fields to japanese morphological analysis. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 230–237, 2004.

[56] Guillaume Lample and Alexis Conneau. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*, 2019.

[57] Hang Le, Juan Pino, Changhan Wang, Jiatao Gu, Didier Schwab, and Laurent Besacier. Dual-decoder transformer for joint automatic speech recognition and multilingual speech translation. In *Proceedings of the 28th International Conference on Computational Linguistics*, pages 3520–3533, 2020.

[58] Andy T Liu, Shang-Wen Li, and Hung-yi Lee. Tera: Self-supervised learning of transformer encoder representation for speech. *arXiv preprint arXiv:2007.06028*, 2020.

[59] Yuchen Liu, Hao Xiong, Zhongjun He, Jiajun Zhang, Hua Wu, Haifeng Wang, and Chengqing Zong. End-to-end speech translation with knowledge distillation. *arXiv preprint arXiv:1904.08075*, 2019.

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

[61] Yuchen Liu, Junnan Zhu, Jiajun Zhang, and Chengqing Zong. Bridging the modality gap for speech-to-text translation. *arXiv preprint arXiv:2010.14920*, 2020.

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

[63] Xingcheng Ma. Effect of word order asymmetry on cognitive process of english-chinese sight translation by interpreting trainees: Evidence from eye-tracking. 2019.

[64] Xutai Ma, Juan Pino, and Philipp Koehn. SimulMT to SimulST: Adapting simultaneous text translation to end-to-end simultaneous speech translation. In *Proceedings of the 1st Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics and the 10th International Joint Conference on Natural Language Processing*, 2020.

[65] Xutai Ma, Yongqiang Wang, Mohammad Javad Dousti, P. Koehn, and J. Pino. Streaming simultaneous speech translation with augmented memory transformer. *ArXiv*, abs/2011.00033, 2020.

[66] Shigeki Matsubara, Akira Takagi, Nobuo Kawaguchi, and Yasuyoshi Inagaki. Bilingual spoken monologue corpus for simultaneous machine interpretation research. In *LREC*, 2002.

[67] Makoto Morishita, Jun Suzuki, and Masaaki Nagata. NTT neural machine translation systems at WAT 2019. In *Proceedings of the 6th Workshop on Asian Translation*, pages 99–105, Hong Kong, China, November 2019. Association for Computational Linguistics. doi: 10.18653/v1/D19-5211. URL `https://www.aclweb.org/anthology/D19-5211`.

[68] Niko Moritz, Takaaki Hori, and Jonathan Le. 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, 2020.

[69] Graham Neubig, Hiroaki Shimizu, Sakriani Sakti, Satoshi Nakamura, and Tomoki Toda. The naist simultaneous translation corpus. In *Making Way in Corpus-based Interpreting Studies*, pages 205–215. Springer, 2018.

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

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

[72] Matthew E Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. Deep contextualized word representations. *arXiv preprint arXiv:1802.05365*, 2018.

[73] Juan Pino, Qiantong Xu, Xutai Ma, Mohammad Javad Dousti, and Yun Tang. Self-training for end-to-end speech translation. *Proc. Interspeech 2020*, pages 1476–1480, 2020.

[74] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Nagendra Goel, Mirko Hannemann, Yanmin Qian, Petr Schwarz, and Georg Stemmer. The kaldi speech recognition toolkit. In *In IEEE 2011 workshop*, 2011.

[75] Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. Improving language understanding by generative pre-training. 2018.

[76] Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. SQuAD: 100,000+ questions for machine comprehension of text. *arXiv preprint arXiv:1606.05250*, 2016.

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

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

[79] Jonathan Shen, Ruoming Pang, Ron J Weiss, Mike Schuster, Navdeep Jaitly, Zongheng Yang, Zhifeng Chen, Yu Zhang, Yuxuan Wang, Rj Skerrv-Ryan, et al. Natural tts synthesis by conditioning wavenet on mel spectrogram predictions. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4779–4783. IEEE, 2018.

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

[81] Hiroaki Shimizu, Graham Neubig, Sakriani Sakti, Tomoki Toda, and Satoshi Nakamura. Collection of a simultaneous translation corpus for comparative analysis. In *LREC*, pages 670–673. Citeseer, 2014.

[82] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust dnn embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[83] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. Neural lattice-to-sequence models for uncertain inputs. *arXiv preprint arXiv:1704.00559*, 2017.

[84] Matthias Sperber, Graham Neubig, Jan Niehues, and Alex Waibel. Attention-Passing Models for Robust and Data-Efficient End-to-End Speech Translation. *Transactions of the Association for Computational Linguistics (TACL)*, 2019. URL `https://arxiv.org/abs/1904.07209`.

[85] Mihaela Stoian, Sameer Bansal, and Sharon Goldwater. Analyzing asr pretraining for low-resource speech-to-text translation. In *ICASSP*, 2020.

[86] Yu Sun, Shuohuan Wang, Yukun Li, Shikun Feng, Xuyi Chen, Han Zhang, Xin Tian, Danxiang Zhu, Hao Tian, and Hua Wu. Ernie: Enhanced representation through knowledge integration. *arXiv preprint arXiv:1904.09223*, 2019.

[87] Wilson L Taylor. "cloze procedure": A new tool for measuring readability. *Journalism quarterly*, 30(4):415–433, 1953.

[88] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

[89] Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*, 2018.

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

[91] Chengyi Wang, Yu Wu, Shujie Liu, Zhenglu Yang, and Ming Zhou. Bridging the gap between pre-training and fine-tuning for end-to-end speech translation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI 2020, The Thirty-Second Innovative Applications of Artificial Intelligence Conference, IAAI 2020, The Tenth AAAI Symposium on*

Educational Advances in Artificial Intelligence, EAAI 2020, New York, NY, USA, February 7-12, 2020, 2020.

[92] Chengyi Wang, Yu Wu, Shujie Liu, Ming Zhou, and Zhenglu Yang. Curriculum pre-training for end-to-end speech translation. In *ACL*, 2020.

[93] Yuxuan Wang, Daisy Stanton, Yu Zhang, RJ-Skerry Ryan, Eric Battenberg, Joel Shor, Ying Xiao, Ye Jia, Fei Ren, and Rif A Saurous. Style tokens: Unsupervised style modeling, control and transfer in end-to-end speech synthesis. In *International Conference on Machine Learning*, pages 5180–5189. PMLR, 2018.

[94] S. Watanabe, T. Hori, S. Kim, J. R. Hershey, and T. Hayashi. Hybrid ctc/attention architecture for end-to-end speech recognition. 2017.

[95] Ron J Weiss, Jan Chorowski, Navdeep Jaitly, Yonghui Wu, and Zhifeng Chen. Sequence-to-sequence models can directly translate foreign speech. *Proc. Interspeech 2017*, pages 2625–2629, 2017.

[96] Hao Xiong, Ruiqing Zhang, Chuanqiang Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. Dutongchuan: Context-aware translation model for simultaneous interpreting. *arXiv preprint arXiv:1907.12984*, 2019.

[97] Peng Xu, Jaeho Kang, Michael Ringgaard, and Franz Josef Och. Using a dependency parser to improve smt for subject-object-verb languages. In *Proceedings of human language technologies: The 2009 annual conference of the North American chapter of the association for computational linguistics*, pages 245–253, 2009.

[98] Junichi Yamagishi, Christophe Veaux, and Kirsten MacDonald. CSTR VCTK Corpus: English multi-speaker corpus for CSTR voice cloning toolkit (version 0.92), 2019.

[99] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 6199–6203. IEEE, 2020.

[100] DEN Yasuharu, OGISO Toshinobu, OGURA Hideki, YAMADA Atsushi, MINEMATSU Nobuaki, UCHIMOTO Kiyotaka, and KOISO Hanae. The development of an electronic dictionary for morphological analysis and its application to japanese corpus linguistics. *Japanese linguistics*, 22:101–123, oct 2007. doi: info:doi/10.15084/00002185. URL `https://ci.nii.ac.jp/naid/120006595341/en/`.

[101] Steve Young, Gunnar Evermann, Mark Gales, Thomas Hain, Dan Kershaw, Xunying Liu, Gareth Moore, Julian Odell, Dave Ollason, Dan Povey, et al. The htk book. *Cambridge university engineering department*, 3(175):12, 2002.

[102] Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. Simultaneous translation with flexible policy via restricted imitation learning. In *ACL*, 2019.

[103] Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1349–1354, 2019.

[104] Baigong Zheng, Renjie Zheng, Mingbo Ma, and Liang Huang. Simpler and faster learning of adaptive policies for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, 2019.

[105] Baigong Zheng, Kaibo Liu, Renjie Zheng, Mingbo Ma, Hairong Liu, and Liang Huang. Simultaneous translation policies: From fixed to adaptive. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, 2020.

[106] Renjie Zheng, Mingbo Ma, and Liang Huang. Multi-reference training with pseudo-references for neural translation and text generation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3188–3197, 2018.

[107] Renjie Zheng, Mingbo Ma, Baigong Zheng, and Liang Huang. Speculative beam search for simultaneous translation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 1395–1402, 2019.

[108] Renjie Zheng, Mingbo Ma, Baigong Zheng, Kaibo Liu, Jiahong Yuan, Kenneth Church, and Liang Huang. Fluent and low-latency simultaneous speech-to-speech translation with self-adaptive training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 3928–3937, 2020.

[109] Renjie Zheng, Junkun Chen, Mingbo Ma, and Liang Huang. Fused acoustic and text encoding for multimodal bilingual pretraining and speech translation. *Proceedings of the 38th International Conference on Machine Learning*, 2021.