Synchronous Speech Recognition and Speech-to-Text Translation with Interactive Decoding

Yuchen Liu,[1,2*] **Jiajun Zhang,**[1,2] **Hao Xiong,**[4] **Long Zhou,**[1,2]
Zhongjun He,[4] **Hua Wu,**[4] **Haifeng Wang,**[4] **and Chengqing Zong**[1,2,3]
[1] National Laboratory of Pattern Recognition, Institute of Automation, CAS
[2] University of Chinese Academy of Sciences
[3] CAS Center for Excellence in Brain Science and Intelligence Technology
[4] Baidu Inc., No. 10, Shangdi 10th Street, Beijing, China
{yuchen.liu, jjzhang, long.zhou, cqzong}@nlpr.ia.ac.cn
{xionghao05, hezhongjun, wu_hua, wanghaifeng}@baidu.com

Abstract

Speech-to-text translation (ST), which translates source language speech into target language text, has attracted intensive attention in recent years. Compared to the traditional pipeline system, the end-to-end ST model has potential benefits of lower latency, smaller model size, and less error propagation. However, it is notoriously difficult to implement such a model without transcriptions as intermediate. Existing works generally apply multi-task learning to improve translation quality by jointly training end-to-end ST along with automatic speech recognition (ASR). However, different tasks in this method cannot utilize information from each other, which limits the improvement. Other works propose a two-stage model where the second model can use the hidden state from the first one, but its cascade manner greatly affects the efficiency of training and inference process. In this paper, we propose a novel interactive attention mechanism which enables ASR and ST to perform synchronously and interactively in a single model. Specifically, the generation of transcriptions and translations not only relies on its previous outputs but also the outputs predicted in the other task. Experiments on TED speech translation corpora have shown that our proposed model can outperform strong baselines on the quality of speech translation and achieve better speech recognition performances as well.

1 Introduction

Speech-to-text translation (hereinafter referred to as speech translation) aims to translate a speech in source language into a text in target language, which can help people efficiently communicate with each other in different languages. The traditional approach is a pipeline system composed of an automatic speech recognition (ASR) model and a text machine translation (MT) model. In this approach, two models are independently trained and tuned, leading to the problem of time delay, parameter redundancy, and error propagation. In contrast, end-to-end ST model has potential advantages to alleviate these problems. Recent works have emerged rapidly and shown promising performances (**?**; **?**; **?**; **?**; **?**).

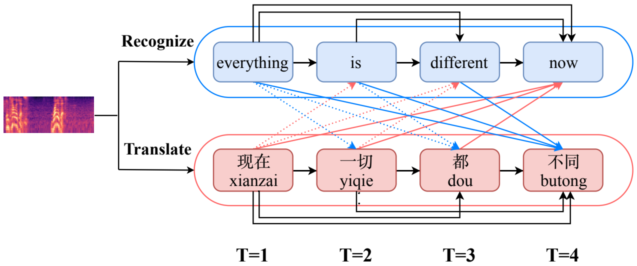

Figure 1: An example of an English speech recognition and the corresponding translation in Chinese, where the outputs in two tasks can interact with each other.

Despite the advantages, it is notoriously difficult to implement an end-to-end ST model which does not use transcriptions as intermediate, and its performance is generally limited. Previous studies resort to pretraining or multi-task learning to improve the translation quality. They either apply a pretrained encoder trained on ASR data (**?**), or jointly train with ASR to obtain a better acoustic model, or with MT to acquire a better language model (**?**; **?**; **?**). However, the basic unit shared between different tasks is module parameters. Different tasks in this method cannot utilize information from each other. To alleviate this flaw, several studies propose analogous two-stage model (**?**; **?**; **?**). In this model, decoder in the first stage performs recognition and generates a hidden state with which the second decoder conducts translation. Although the translation quality can be improved with additional information from the first decoder, the second decoder needs to wait until the complete transcription is recognized, greatly limiting the efficiency of training and inference process. In addition, this model can only make the translation process utilize information from recognition process, but cannot for the other direction.

However, we find that the generation process in ASR and ST can help each other: (1) the generation of speech translation would become easier with additional information from the transcribed words than just from the speech signal, (2) the translated words can also assist the recognition process.

[*] This work is done while Yuchen Liu is doing research intern at Baidu Inc.

As the example shown in Figure 1, the input is a complete speech utterance in English and the outputs in two tasks can interact with each other. When translating Chinese word "yiqie" (the meaning of everything) at step $T = 2$ in the ST task, the already transcribed word "everything" at step $T = 1$ in the transcription can provide the additional context. For ASR task, the translated word "xianzai" at step $T = 1$ can also help to recognize "now" at step $T = 4$. Therefore, if the generation of two tasks can interact with each other, the quality of transcription and translation can both be improved.

To this end, we propose a novel interactive learning model which can perform speech recognition and speech translation synchronously and interactively. Compared with the traditional multi-task learning model which shares part of parameters and treats different tasks separately, tasks in our approach can exchange the information of each other. With an interactive attention sub-layer, translation decoder in our model predicts next word with the transcribed words as auxiliary information, and for recognition decoder vice versa. Therefore, at each step, word prediction in each task not only relies on its previously generated outputs, but also the outputs in the other task. Furthermore, we introduce a wait-k policy where the generation process of speech translation is always k steps later than speech recognition, so that the translation decoder can attend to more transcribed words. We conduct extensive experiments to verify the effectiveness of our proposed approaches on new TED English-to-German/French/Chinese/Japanese speech translation corpora.

Our main contributions are summarized as follows:

- We propose an interactive learning model which can conduct speech recognition and speech translation interactively, enhancing the quality of both tasks.
- Different from traditional multi-task learning model which generates transcriptions or translations separately, our method can simultaneously generate both transcriptions and translations in one model.
- Experiments on four language pairs have demonstrated that our model can outperform strong baselines, including the pipeline system, the pretrained end-to-end ST model, the traditional multi-task learning model, and the two-stage model.

2 Related Work

Speech Translation Speech translation has traditionally been approached through a pipeline system which consists of an ASR model and a text MT model (**?**; **?**; **?**; **?**; **?**). Recent works have shown the feasibility of collapsing the cascade system into an end-to-end model. The first conjecture was proposed by **?** (**?**) who presumed that end-to-end speech translation is possible to implement with the development of memory, computation speed, and representation methods. It is not until 2016 that **?** (**?**) realized the first pure end-to-end model without using any source transcriptions. Considering its notorious difficulty, the performance of end-to-end ST model is generally limited. Several work proposed a variety of approaches to improve the translation quality. Some applied multi-task learning to train speech translation jointly with ASR (**?**; **?**; **?**). Others attempted to pretrain ST model with extra ASR data to promote acoustic model, or with target sentences to improve language model (**?**; **?**). **?** (**?**) proposed to use a text MT model as teacher model to instruct ST model through knowledge distillation.

An intuition is that speech translation can become easier if the model has access to the transcription as intermediate. Therefore, several researchers proposed two-stage models where the first decoder is used to recognize transcriptions and the second decoder conducts translating with the hidden state in the former stage. **?** (**?**) first proposed the basic two-stage model and used pretraining strategy for the individual sub-models. **?** (**?**; **?**) employed a triangle model on low-resource speech translation. **?** (**?**) further applied an attention-passing mechanism which can integrate auxiliary data and improve model robustness. However, the second decoder needs to wait until the complete transcription is recognized, which greatly affects the training and inference efficiency. Besides, it can only utilize transcriptions to improve translation quality but leaves the recognition task alone. As shown in Figure 1, the outputs of recognition and translation are complementary and can benefit each other. Therefore, it is reasonable to improve the quality of both tasks through interactive learning.

Synchronous Inference **?** (**?**) proposed a synchronous bidirectional inference model in which left-to-right and right-to-left inferences perform in parallel. The two decoding directions can help each other, and make full use of the target-side history and future information during translation. **?** (**?**) further applied this inference model on other sequence generation tasks, such as summarization, obtaining significant improvement as well. However, their works are conducted on the same task with outputs in different directions. The most related work with us is from **?** (**?**) who synchronously performed multilingual translation within a beam. In our work, we have two different tasks and aim to implement speech recognition and speech translation in one model synchronously.

3 Background

Considering that Transformer model is now the state-of-art model in MT field (**?**), and also shows a superior performance in ASR filed (**?**; **?**), we adopt Transformer model as the core structure. However, our proposed approach can be applied to any encoder-decoder architectures.

The Transformer follows the typical encoder-decoder architecture. The encoder first maps the input sequence $I = (i_1, i_2, \cdots, i_n)$ into a sequence of continuous representations $Z = (z_1, z_2, \cdots, z_n)$, from which the decoder generates the output sequence $O = (o_1, o_2, \cdots, o_m)$ one word at a time. In Transformer, the encoder includes N layers and each layer is composed of two sub-layers: the self-attention sub-layer and the feed-forward sub-layer. The decoder also consists of N layers and each layer has three sub-layers. The first one is the masked self-attention sub-layer, which adds masks to prevent present positions from attending to the future positions during training. The second is the encoder-

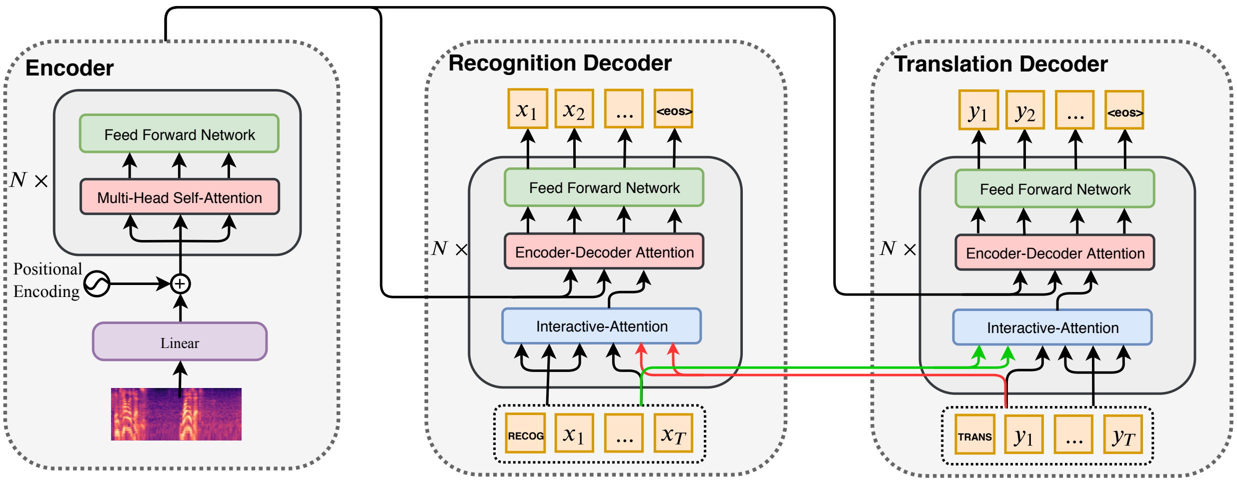

Figure 2: The model architecture of our method which is adopted from Transformer. The left part is the encoder which takes speech features as input and is shared by speech recognition model and speech translation model. The middle and right parts are two decoders where the middle is recognition decoder and the right is the translation decoder. There is an interactive attention sub-layer between two decoders which makes the decoders can utilize information from each other.

decoder attention sub-layer, followed by the feed-forward sub-layer. Residual connection and layer normalization are employed around each sub-layer in the encoder and decoder.

The calculation process of three attention sub-layers can be formalized into the same formula as,

$$\text{Attention}(\mathbf{Q}, \mathbf{K}, \mathbf{V}) = \text{softmax}(\frac{\mathbf{Q}\mathbf{K}^T}{\sqrt{d_k}})\mathbf{V} \tag{1}$$

where $\mathbf{Q}$, $\mathbf{K}$ and $\mathbf{V}$ denotes the query, key, and value respectively. d_k is the dimension of the key. The feed-forward sub-layer is then applied to yield the output of a whole layer. And softmax function is employed to predict the final output.

It is worth noting that for the self-attention sub-layer, the query, key, and value are the hidden representation from the same layer. For encoder-decoder sub-layer, the query is the hidden representation from the masked self-attention sub-layer in the decoder, the key and value are from the top layer in the encoder.

4 Our Approach

In this section, we propose a novel framework to implement interactive learning for speech recognition and speech translation during training and inference, which is shown in Figure 2. Before we introduce this framework in detail, we first introduce how Transformer model is applied to the ASR, MT, and ST tasks.

4.1 ASR, MT, and ST Task

Speech recognition, text machine translation, and speech translation tasks can all adopt Transformer model, while different tasks have different input sequences I and output sequences O. Specifically,

- For ASR task, the input sequence $I = S = [s_1, \ldots, s_T]$ is a sequence of speech features, where T is the frame number of speech sequence. Specifically, the speech feature is first converted from raw speech signal by applying log-Mel filterbanks with mean and variance normalization. Frame stack and downsampling are used to reduce the input length similar with **?** (**?**), resulting in a sequence with dimension of $d_{\text{filterbank}} \times \text{num}_{\text{stack}}$. The output sequence $O = X = [x_1, \ldots, x_N]$ is the corresponding transcription, where N is the source sentence length.
- For MT task, the input sequence $I = X = [x_1, \ldots, x_N]$ is the transcription in source language and the output sequence $O = Y = [y_1, \ldots, y_M]$ is the corresponding translation in target language, where M is the target sentence length.
- For end-to-end ST task, the input sequence $I = S = [s_1, \ldots, s_T]$ is the same with ASR task and the output sequence $O = Y = [y_1, \ldots, y_M]$ is the corresponding translation in target language.

In addition to the end-to-end model, ST task can also be implemented in a pipeline approach, where the speech utterance is first transcribed by an ASR model and then passed to a MT model. Another method is the multi-task learning model where the ASR model and ST model are combined with a shared encoder and trained jointly.

4.2 Interactive Learning Model

In the traditional multi-task learning, different tasks are trained independently with shared parameters. However, as discussed in Section 1, the output in one task is complementary with that in the other which can assist the prediction. Therefore, it is reasonable to improve the performances of both tasks by interactively exchanging information from each other. Besides, the traditional multi-task learning can only perform one task during inference, while sometimes the transcription and translation are required at the same time. To solve these problems, we propose an interactive learning model where two tasks can not only interactively learn from each other but also generate predictions synchronously.

The main model structure is shown in Figure 2. First, the speech signal is processed into the acoustic feature sequence and projected by a linear transformation layer, whose dimension is converted to the hidden size d_{model}. Then, the encoder embeds the sequence into a high level acoustic representation. Two decoders are applied for different tasks in which one performs speech recognition and the other conducts speech translation.

To make two decoders interactively learn from each other, we replace the self-attention sub-layer in the standard Transformer decoder with our proposed interactive attention sub-layer. As shown in Figure 3, the interactive attention sub-layer is composed of a self-attention sub-layer and a cross-attention sub-layer. The former uses the hidden representation from task 1 as the query $\mathbf{Q_1}$, key $\mathbf{K_1}$ and value $\mathbf{V_1}$ to learn higher representation $\mathbf{H_{self}}$. While the latter uses the hidden representation from task 1 as the query $\mathbf{Q_1}$, and the hidden representation from task 2 as key $\mathbf{K_2}$ and value $\mathbf{V_2}$ to integrate the representation $\mathbf{H_{cross}}$ of the other task. All the hidden representations are extracted from the same layer. It can be calculated as:

$$\mathbf{H_{self}} = \text{Attention}(\mathbf{Q_1}, \mathbf{K_1}, \mathbf{V_1}) \tag{2}$$

$$\mathbf{H_{cross}} = \text{Attention}(\mathbf{Q_1}, \mathbf{K_2}, \mathbf{V_2}) \tag{3}$$

Then the output of self-attention sub-layer and that of cross-attention sub-layer can be integrated by a fusion function to obtain the final representation:

$$\mathbf{H_{final}} = \text{Fusion}(\mathbf{H_{self}}, \mathbf{H_{cross}}) \tag{4}$$

We use a linear interpolation as fusion function, which can be calculated as:

$$\mathbf{H_{final}} = \mathbf{H_{self}} + \lambda * \mathbf{H_{cross}} \tag{5}$$

where λ is a hyper-parameter to control how much information of the other task should be taken into consideration. Then both decoders can obtain the combined representation which contains information from the outputs in two tasks.

We apply interactive attention sub-layer to replace the self-attention sub-layer in the standard Transformer decoder, and it also utilizes the residual connections (**?**) around each sub-layer, followed by layer normalization (**?**). Other modules remain the same as standard Transformer model.

4.3 Training and Inference

Training Since our approach performs ASR task and ST task in one model, two tasks can be optimized at the same time. We additionally append two special labels ($\langle RECOG \rangle$ and $\langle TRANS \rangle$) at the start of transcriptions and translations to indicate whether the generation process is recognition or translation. Given a set of training data $D = \{\mathrm{S}^{(j)}, \mathrm{X}^{(j)}, \mathrm{Y}^{(j)}\}_{j=1}^{|D|}$, where S is the sequence of speech features, X is the sequence of source transcription and Y is the corresponding target translation, the objective function is to maximize the log-likelihood over both the transcription and the translation,

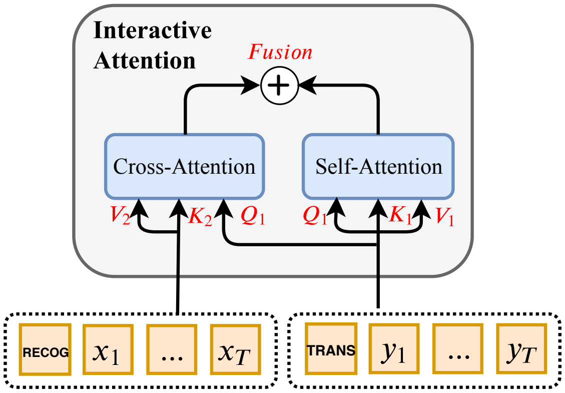

Figure 3: The interactive attention sub-layer consists of a self-attention sub-layer with a cross-attention sub-layer which can capture the information from the other task.

$$L(\theta) = \sum_{j=1}^{|D|}(\log P(\mathrm{X}^{(j)}|\mathrm{S}^{(j)}, \mathrm{Y}^{(j)}) + \log P(\mathrm{Y}^{(j)}|\mathrm{S}^{(j)}, \mathrm{X}^{(j)})) \tag{6}$$

With interactive attention sub-layer, the recognition decoder and translation decoder can utilize the information from both itself and the other. Specifically, at time i, the recognition decoder and translation decoder have generated the first $i-1$ words respectively, then the i-th word in translation can be predicted based on the $i-1$ already generated translation words and the $i-1$ already transcribed words. It is the same for the generation process in speech recognition task. Therefore, the prediction probability of each transcription $P(\mathrm{X}|\mathrm{S}, \mathrm{Y})$ and translation $P(\mathrm{Y}|\mathrm{S}, \mathrm{X})$ can be formalized as,

$$\log P(\mathrm{X}|\mathrm{S}, \mathrm{Y}) = \sum_{i=0}^{N-1} \log p(x_i | x_{<i}, \mathrm{S}, y_{<i}) \tag{7}$$

$$\log P(\mathrm{Y}|\mathrm{S}, \mathrm{X}) = \sum_{i=0}^{M-1} \log p(y_i | y_{<i}, \mathrm{S}, x_{<i}) \tag{8}$$

Inference The inference process is similar with training. We run beam search algorithm for the two tasks. Two beams are applied for different tasks and expand hypotheses respectively. The outputs of two tasks are generated in parallel, with the interactive attention sub-layer to implement information exchanging between two decoders. At each step, the word with highest probability will be selected and added to each hypotheses. The inference process terminates until both tasks reach the end of sentences. In this way, the hypotheses in speech recognition task and speech translation task can be generated synchronously.

4.4 Wait-k Policy

Considering that the speech translation task is more difficult than speech recognition, it would be helpful for the translation process if the translation decoder can get more information at each step. Therefore, we introduce a wait-k policy, in which the translation decoder begins to perform until the first k source words are transcribed by the recognition decoder. That is, the generation process of translation is always k words later than the generation of transcription. For example, if $k = 2$, the first translation word is predicted based on the acoustic representation of encoder with the first two transcription words. Then the second translation word can use the hidden representation of acoustic encoder, the first three transcription words, and the first predicted translation word, etc. **?** (**?**), they applied the wait-k policy in simultaneous translation where the translation decoder is always k words behind the incoming source stream. Different from them, the decoders in our work have access to the complete source speech utterances, and the wait-k policy is only applied to the translation decoder. During training, we append k special label ($\langle DELAY \rangle$) before the start of translation, which indicates that the generation process of translation is k steps later than recognition.

5 Experiments

5.1 Dataset

Prior studies usually conduct experiments on Fisher and Callhome, a corpus of telephone conversations which include English transcriptions and Spanish (Es) translations (**?**). However, the ASR word error rate (WER) of this corpus is fairly high[1], due to the spontaneous speaking style and challenging acoustics. Therefore, we construct a new speech translation corpus collected from TED talks which are a popular data resource in both speech recognition and machine translation fields.

To build this corpus, we first crawl the raw data (including video data, subtitles and timestamps) from the TED website[2]. Audio in each talk is extracted from video and saved in *wav* format. Subtitles in each talk usually have an English manual transcription and more than one translations in different languages. Here, we only collect the subtitles which contains English transcription with translations in German, French, Chinese, and Japanese (briefly, De/Fr/Zh/Ja). Adjacent subtitles and timestamps in English transcriptions are combined according to strong punctuations, such as period and question marks. Then each audio is segmented into small utterances based on the combined timestamps. This process guarantees that each speech utterance contains complete semantic information, which is important for translation. Translations in different languages are also combined based on the timestamps to align with speech utterances[3].

Finally, we obtain 235K/299K/299K/273K triplet data for En-De/Fr/Zh/Ja language pairs respectively, which contain speech utterances, manual transcriptions and translations. Development and test sets are split according to the partition in IWSLT. We use tst2014 as development (*Dev*) set and tst2015 as *test* set. The remaining data are used as training set. This dataset is available on http://www.nlpr.ia.ac.cn/cip/dataset.htm.

5.2 Model Settings

The speech features have 80-dimension log-Mel filterbanks extracted with a step size of 10ms and window size of 25ms, which are extended with mean subtraction and variance normalization. The features are stacked with 3 frames to the left and downsampled to a 30ms frame rate. We remove punctuations, lowercase and tokenize English transcriptions using scripts from Moses[4]. We also lowercase and tokenize the translations in German and French. Chinese sentences are segmented by Jieba[5] and Japanese sentences are segmented by Mecab[6]. For En-De and En-Fr, parallel sentences are encoded using BPE method (**?**) which has a shared vocabulary of 30K tokens. For En-Zh and En-Ja, we encode source transcriptions and target translations, respectively, and the vocabulary size is limited to the most frequent 30K. ASR performance is evaluated with WER computed on lowercased, tokenized manual transcriptions without punctuations. As for text translation and speech translation, we report case-insensitive tokenized BLEU (**?**) for De/Fr language pairs and character-level BLEU for Zh/Ja.

All of the models are implemented based on the model adopted from Transformer. We use the configuration *transformer_base* used by **?** (**?**) which contains 6-layer encoders and 6-layer decoders with 512-dimensional hidden sizes. We train our models with Adam optimizer (**?**) on 2 NVIDIA V100 GPUs. For inference, we perform beam search with a beam size of 4.

5.3 Baselines

We compare the proposed method with the following baseline models:

- Pipeline system: ASR and MT model are independently trained, and then the outputs of ASR model are taken as the inputs to MT model.
- Pretrained ST model: The encoder of end-to-end ST model is first initialized by training on ASR data, and then the model is finetuned on speech translation data.
- Multi-task learning model: ASR model and ST model are jointly trained with the parameters of encoder shared.
- Two-stage model: This model contains two stages where the outputs of the first stage are transcriptions and the second stage are translations. We re-implement the basic model based on Transformer following **?** (**?**). The model in the first stage is also initialized by training on ASR data.

5.4 Results

Table 1 shows the main results of speech recognition and speech translation on En-De/Fr/Zh/Ja TED corpora. The

[1] This corpus contains ASR outputs which are provided by **?** (**?**), with a WER of over 40%.

[2] https://www.ted.com

[3] **?** (**?**) built similar corpora, however their corpora do not consist of En-Zh and En-Ja language pairs, and they used a different segment way.

[4] https://www.statmt.org/moses/

[5] https://github.com/fxsjy/jieba

[6] http://taku910.github.io/mecab

Model	En-De		En-Fr		En-Zh		En-Ja	
	WER(↓)	BLEU(↑)	WER(↓)	BLEU(↑)	WER(↓)	BLEU(↑)	WER(↓)	BLEU(↑)
Text MT	/	22.19	/	30.68	/	25.01	/	22.93
Pipeline	16.19	19.50	14.20	26.62	14.20	21.52	14.21	**20.87**
E2E	16.19	16.07	14.20	27.63	14.20	19.15	14.21	16.59
Multi-task	15.20	18.08	13.04	28.71	13.43	20.60	14.01	18.73
Two-stage	15.18	19.08	13.34	**30.08**	13.55	20.99	14.12	19.32
Interactive	**14.76**	**19.82**$^{*}_{\star}$‡	**12.87**$^{*}_{\star}$‡	29.79$^{*}_{\star}$	**13.38***	**21.68**$_{\star}$‡	**13.91**$^{*}_{\star}$‡	19.60$_{\star}$‡

Table 1: Evaluation of speech recognition and speech translation on TED En-De/Fr/Zh/Ja datasets. E2E denotes to the pretrained end-to-end ST model, and Interactive represents our proposed interactive learning model. $*$, $\star$, and ‡ indicate Interactive learning model is statistically significant ($p < 0.01$) compared with Pipeline, Multi-task, and Two-stage, respectively.

BLEU scores in the first row are the translation results by text MT model when the clean manual transcriptions are given as inputs. This can be seen as the upper bound for speech translation task. We set $\lambda = 0.3$ and $k = 3$ in the interactive learning model.

Similar Languages We first analyze En-De and En-Fr language pairs. From the first two rows, we can see that the translation quality drops dramatically when the output of ASR model is fed as the input to the MT model compared with the clean transcriptions input. It indicates that text MT model is very sensitive to recognition errors, which is one of the main problems in the pipeline system. Pretrained end-to-end ST model outperforms the pipeline system by 0.99 BLEU points on En-Fr language direction, but it does not show superiority on En-De. We argue that end-to-end model may have superiority of less error propagation on more similar language pairs, such as En-Fr or En-Es. This is consistent with **?** (**?**) who conducted experiments on En-Es and found end-to-end ST has better performance than the pipeline system. Compared with the end-to-end model, multi-task learning model can obtain some improvements, which improves 2.01 and 0.98 BLEU scores for En-De and En-Fr, respectively. However, with information exchanging, our proposed interactive learning model significantly outperforms multi-task learning model on the quality of both speech recognition and speech translation. It demonstrates the effectiveness of the interactive attention mechanism. Although our method does not outperform two-stage model on En-Fr speech translation task, it has a better performance on ASR result. The underlying reason is that the goal of two-stage model is to optimize the translation quality with the information of complete transcription while ignoring the recognition, so it can improve the translation quality but leave the recognition alone.

Dissimilar Languages It is even more difficult to implement end-to-end speech translation on dissimilar language pairs, such as En-Zh and En-Ja. Because these kind of models are required to learn not only the alignments between source frames and translation words, but also the word orders in long distances. Therefore, in our experiments, most of the end-to-end models are inferior than pipeline system. However, the proposed interactive learning model can significantly outperform end-to-end ST model, traditional multi-task learning model and two-stage model, approaching to or slightly better than pipeline system.

λ	Dev		Test	
	WER	BLEU	WER	BLEU
0.0	14.87	15.74	13.43	20.60
0.1	**14.47**	15.93	**12.92**	20.88
0.3	14.51	**16.28**	13.24	**21.01**
0.5	15.50	15.66	14.17	20.68
1.0	15.92	15.06	14.52	20.13

Table 2: The performance of speech recognition and speech translation under different hyper-parameters λ on the En-Zh *Dev* set and *Test* set.

Wait-k	Dev		Test	
	WER	BLEU	WER	BLEU
Wait-0	14.51	16.28	13.24	21.01
Wait-1	14.29	16.09	**13.17**	21.30
Wait-3	**14.24**	**16.74**	13.38	**21.68**
Wait-5	14.36	16.55	13.51	21.45

Table 3: The performance of speech recognition and speech translation with different word latency in wait-k policy on the En-Zh *Dev* set and *Test* set.

5.5 Effect of the Hyper-parameters

We investigate how much information from two tasks should be taken into consideration in the interactive attention sublayer. Table 2 reports the WER and BLEU scores under different λ on the En-Zh. If $\lambda = 0.0$, the model degrades to traditional multi-task learning model which does not utilize any information from the other task. As shown in the table, as λ increases, both recognition quality and translation quality can be improved with information interacting. When $\lambda = 0.3$, our interactive learning model achieves the best performance on the speech translation task. However, λ can not be too large, otherwise two tasks may interfere with each other and affect its own performance. Therefore, we use $\lambda = 0.3$ for all experiments.

5.6 Effect of k in Wait-k Policy

We then investigate the effect of word latency in wait-k policy on En-Zh language pairs. As shown in Table 3, the speech translation quality in BLEU scores can be improved with the increase of word latency. It indicates that the speech

Model	Params	Speed	
		Train	Inference
Pipeline	122.4M	/	10.89
E2E	61.2M	4.73	16.17
Multi-task	61.2M	4.41	16.26
Two-stage	92.7M	1.13	7.44
Interactive	61.2M	4.23	11.98

Table 4: Statistics of parameters, training and inference speeds. The number in Train denotes the average number of training steps per second. The number in Inference is the average amount of sentences generated per second.

translation task can become easier if more source information from the same modality is given. However, as k increases, it will affect the performance of speech recognition task. If $k \to \infty$, this model degrades to the analogous two-stage model. Then the speech translation task can obtain the information from complete transcribed sentence, while speech recognition task can not utilize any information from translations. The interactive learning model has the best performance when $k = 3$.

5.7 Parameters and Speeds

The parameter sizes of different models are shown in Table 4. The pipeline system needs a separate ASR model and MT model, so its parameters are doubled. Two-stage model has 1.5 times larger parameters since it has two different decoders in two stages. In multi-task learning model and interactive learning model, we share the parameter between different tasks. Therefore, they have the same number of parameters with end-to-end model. Table 4 also shows the training and inference speed of different models on En-Zh test set. The training speed of interactive learning model is 4.23 steps per second, which is comparable with the end-to-end model but is much faster than two-stage model. During inference, the average decoding speed of interactive learning model is 11.98 utterances per second. Although it is slower than end-to-end model and multi-task learning model, it can generate transcriptions paired with translations in one model synchronously. While two-stage model can also generate transcription and translation in a single model, its implementation which is in a cascade manner is much slower even than pipeline system.

5.8 Case Study

We show the case study in Figure 4. In pipeline system, ASR model first recognizes the speech utterance into “brainstormed on solutions to the best child is facing their city”. Since it wrongly recognizes “the biggest challenges” into “the best child is”, text MT then translates the incorrect recognition phrase, resulting the result is far from the reference. It is more difficult for the end-to-end ST model to generate a correct translation and its output is totally wrong. This model may comprehend the speech of “brainstorm” into “buhrstone” which has a similar pronunciation and it omits the translation of “the biggest”. Although the multi-task learning model has an enhanced acoustic encoder, it repeatedly attends to the speech of “storm” without transcription as guidance and translates it twice. As for two-stage model, it erroneously recognized “the biggest” into “the best” in the first stage based on which the second decoder also gives a wrong translation. Compared to the above approaches, our model generates the right transcription and translation through interactive attention mechanism, which matches the reference best.

Reference	brainstorm on solutions to the biggest challenges facing their city
	集思广益 想 办法 解决 城市 面临 的 最大 难题
Pipeline	对 最好 的 孩子们 (the best child is) 实施 头脑 风暴 他们 要 面对 他们 的 城市
E2E	带 着 石灰岩 (buhrstone) 的 解决方案，带 着 他们 的 城市 面临 挑战
Multi-task	头脑 风暴 风暴 (storm)，解决 城市 面临 的 最大 的 挑战
Two-stage	头脑 风暴 解决 了 城市 面临 的 最好 的 (the best) 挑战
Interactive	头脑 风暴 解决 了 城市 面临 的 最大 挑战

Figure 4: An Example of speech translation generated by different models. Words in blue and green are original words in the manual transcription, corresponding translation reference and correct translations with the similar meaning, while words in red are the wrong translations.

6 Conclusion and Future Work

In this paper, we propose an interactive learning model to conduct speech recognition and speech translation interactively and simultaneously. The generation process of recognition and translation in this model can not only utilize the already generated outputs, but also the outputs generated in the other task. We then present a wait-k policy which can further improve the speech translation quality. Experimental results on different language pairs demonstrate the effectiveness of our model. In the future, we plan to design a streaming encoder and make a step forward in achieving end-to-end simultaneous interpretation.

7 Acknowledgments

The research work described in this paper has been supported by the National Key Research and Development Program of China under Grant No. 2016QY02D0303, the Natural Science Foundation of China under Grant No. U1836221 and 61673380, and Beijing Municipal Science and Technology Project No. Z181100008918017 as well. The research work in this paper has also been supported by Beijing Advanced Innovation Center for Language Resources.

Rem illum minus mollitia pariatur maiores ipsam, doloribus mollitia nihil porro autem minus perferendis repellendus molestias cupiditate expedita fugit, sapiente tempora autem sit perferendis harum inventore, eius rerum molestiae minima aliquam dicta natus officiis est quia earum?Tempora nostrum exercitationem, nulla maxime cum,