Consecutive Transcription and Translation for Speech-to-text Translation

Submission ID: 9845

Abstract

Speech-to-text translation (ST), which directly translates the source language speech to the target language text, has attracted intensive attention recently. However, the combination of speech recognition and machine translation in a single model poses a heavy burden on the direct cross-modal cross-lingual mapping. To reduce the learning difficulty, we propose COnSecutive Transcription and Translation (**COSTT**), an integral framework for speech-to-text translation. Our method is verified on three mainstream datasets, including Augmented LibriSpeech English-French dataset, TED English-German dataset, and TED English-Chinese dataset. Experiments show that our proposed **COSTT** outperforms the previous state-of-the-art methods. Our code and models will be released.

1 Introduction

Speech translation (ST) aims at translating from source language speech into the target language text. Traditionally, it is realized by cascading an automatic speech recognition (ASR) and a machine translation (MT) (Sperber et al. 2017, 2019b; Zhang et al. 2019; Beck, Cohn, and Haffari 2019; Cheng et al. 2019). Recently, end-to-end ST has attracted much attention due to its appealing properties, such as lower latency, smaller model size, and less error accumulation (Liu et al. 2019a, 2018; Weiss et al. 2017; Bérard et al. 2018; Duong et al. 2016; Jia et al. 2019).

Although end-to-end systems are very promising, cascaded systems still dominate practical deployment in industry. The possible reasons are: *a)* Most research work compared cascaded and end-to-end models under identical data situations. However, in practice, the cascaded system can benefit from the accumulating independent speech recognition or machine translation data, while the end-to-end system still suffers from the lack of end-to-end corpora. *b)* Despite the advantage of reducing error accumulation, the end-to-end system has to integrate multiple complex deep learning tasks into a single model to solve the task, which introduces heavy burden for the cross-modal and cross-lingual mapping. Therefore, it is still an open problem whether end-to-end models or cascaded models are generally stronger.

We argue that a desirable ST model should take advantages of both end-to-end and cascaded models and acquire the practically acceptable capabilities as follows: *a)* it should be end-to-end to avoid error accumulation; *b)* it should be flexible enough to leverage large-scale independent ASR or MT data. At present, few existing end-to-end models can meet all these goals. Most studies resort to pre-training or multitask learning to bridge the benefits of cascaded and end-to-end models (Bansal et al. 2018; Sung et al. 2019; Sperber et al. 2019a). A de-facto framework usually initializes the ST model with the encoder trained from ASR data (i.e. source audio and source text pairs) and then fine-tunes on a speech translation dataset to make the cross-lingual translation. However, it is still challenging for these methods to leverage the bilingual MT data, due to the lack of intermediate text translating stage.

Our idea is motivated by two motivating insights from ASR and MT models. *a)* A branch of ASR models has intermediate steps to extract acoustic feature and decode phonemes, before emitting transcription; and *b)* Speech translation can benefit from decoding the source speech transcription in addition to the target translation text. We propose **COSTT**, a unified speech translation framework with consecutive decoding for jointly modeling speech recognition and translation. **COSTT** consists of two phases, an acoustic-semantic modeling phase (AS) and a transcription-translation modeling phase (TT). The AS phase accepts the speech features and generates compressed acoustic representations. For TT phases, we jointly model both the source and target text in a single shared decoder, which directly generates the speech text sequence and the translation sequence at one pass. This architecture is closer to cascaded translation while maintaining the benefits of end-to-end models. The combination of the AS and the first-part output of the TT phase serves as an ASR model; the TT phase alone serves as an MT model; while the whole makes an end-to-end speech translation by ignoring the first-part of TT output. Simple and effective, **COSTT** is powerful enough to cover the advantage of ASR, MT, and ST models simultaneously.

The contributions of this paper are as follows: *1)* We propose **COSTT**, a unified training framework with consecutive decoding which bridges the benefits of both cascaded and end-to-end models. *2)* As a benefit of explicit multiphase modeling, **COSTT** facilitates the use of parallel bilingual text corpus, which is difficult for traditional end-to-end ST models. *3)* **COSTT** achieves state-of-the-art results on three popular benchmark datasets.

2 Related Work

For speech translation, there are two main research paradigms, the cascaded system and the end-to-end model (Sperber and Paulik 2020; Jan et al. 2018, 2019).

For cascaded system, the most concerned point is how to avoid early decisions, relieve error propagation and better integrate the separately trained ASR and MT modules. To relieve the problem of error propagation and tighter couple cascaded systems: *a)* robust translation models (Cheng et al. 2018, 2019) introduce synthetic ASR errors and ASR related features into the source side of MT corpora; *b)* techniques such as domain adaptation (Liu et al. 2003; Fügen 2008), resegmention (Matusov, Mauser, and Ney 2006), punctuation restoration (Fügen 2008), disfluency detection (Fitzgerald, Hall, and Jelinek 2009) and so on, are proposed to provide the translation model with well-formed and domain matched text inputs.

And a paradigm shift towards end-to-end system is emerging to alleviate the drawbacks of cascaded systems. Bérard et al. (2016); Duong et al. (2016) have given the first proof of the potential of end-to-end speech-to-text translation, which has attracted intensive attentions recently (Vila et al. 2018; Salesky et al. 2018; Salesky, Sperber, and Waibel 2019; Di Gangi, Negri, and Turchi 2019; Bahar, Bieschke, and Ney 2019; Di Gangi et al. 2019; Inaguma et al. 2020). Many works have proved that pre-training then transferring (Weiss et al. 2017; Bérard et al. 2018; Bansal et al. 2018; Stoian, Bansal, and Goldwater 2020) and multi-task learning (Vydana et al. 2020) can significantly improve the performance of end-to-end models. The two-pass decoding (Sung et al. 2019) and attention-passing (Anastasopoulos and Chiang 2018; Sperber et al. 2019a) techniques are proposed to handle the relatively deeper relationships and alleviate error propagation in end-to-end models. Many data augmentation techniques (Jia et al. 2019; Pino et al. 2019b; Bahar et al. 2019; Pino et al. 2019a) are proposed to utilize external ASR and MT corpora. Many semi-supervised training (Wang et al. 2019) methods bring great gain to end-to-end models, such as knowledge distillation (Liu et al. 2019a), modality agnostic meta-learning (Indurthi et al. 2019), model adaptation (Di Gangi et al. 2020) and so on. Curriculum learning (Kano, Sakti, and Nakamura 2018; Wang et al. 2020) is proposed to improve performance of ST models. Liu et al. (2019b); Liu, Spanakis, and Niehues (2020) optimize the decoding strategy to achieve low-latency end-to-end speech translation. (Chuang et al. 2020; Salesky and Black 2020; Salesky, Sperber, and Black 2019) explore additional features to enhance end-to-end models. How to efficiently utilize ASR and MT parallel data is a big problem for ST. However, existing methods mostly resort to ordinary pretraining or multitask learning to integrate external ASR resources, which may face the issue of catastrophic forgetting and modal mismatch. And it is still challenging for previous methods to leverage external bilingual MT data efficiently.

3 Proposed COSTT Approach

3.1 Overview

The detailed framework of our method is shown in Figure 1. To be specific, the speech translation model accepts the original audio feature as input and outputs the target text sequence. We divide our method into two phases, including the acoustic-semantic modeling phase (AS) and the transcription-translation modeling phase (TT). Firstly, the AS phase accepts the speech features, outputs the acoustic representation, and encodes the shrunk acoustic representation into semantic representation. In this work, the small-grained unit, phonemes are selected as the acoustic modeling unit. Then, the TT phase accepts the AS's representation and consecutively outputs source transcription and target translation text sequences with a single shared decoder.

Problem Formulation The speech translation corpus usually contains speech-transcription-translation triples. We add phoneme sequences to make up quadruples, denoted as $\mathcal{S} = \{(\mathbf{x}, \mathbf{u}, \mathbf{z}, \mathbf{y})\}$ (More details about the data preparation can be seen in Section 4). Specially, $\mathbf{x} = (x_1, ..., x_{T_x})$ is a sequence of acoustic features. $\mathbf{u} = (u_1, ..., u_{T_u})$, $\mathbf{z} = (z_1, ..., z_{T_z})$, and $\mathbf{y} = (y_1, ..., y_{T_y})$ represents the corresponding phoneme sequence in source language, transcription in source language and the translation in target language respectively. Meanwhile, $\mathcal{A} = \{(\mathbf{z}', \mathbf{y}')\}$ represents the external text translation corpus, which can be utilized for pre-training the decoder. Usually, the amount of end-to-end speech translation corpus is much smaller than that of text translation, i.e. $|\mathcal{S}| \ll |\mathcal{A}|$.

3.2 Acoustic-Semantic Modeling

The acoustic-semantic modeling phase takes the input of low-level audio features $\mathbf{x}$ and outputs a series of vectors $\mathbf{h}_{\text{AS}}$ corresponding to the phoneme sequence $\mathbf{u}$ in the source language. Different from the general sequence-to-sequence models, two modifications are introduced. Firstly, in order to preserve more acoustic information, we introduce the supervision signal of the connectionist temporal classification (CTC) loss function, a scalable, end-to-end approach to monotonic sequence transduction (Graves et al. 2006; Salazar, Kirchhoff, and Huang 2019). Secondly, since the length of audio features is much larger than that of source phoneme ($T_x \gg T_u$), we introduce a shrinking method which can skip the blank-dominated steps to reduce the encoded sequence length.

Self-Attention with CTC General preprocessing includes down-sampling and linear layers. Down-sampling refers to the dimensionality reduction processing of the input audio features in the time and frequency domains. In order to simplify the network, we adopt physical dimensionality reduction, that is, a method of sampling one frame every three frames. The linear layer maps the length of the frequency

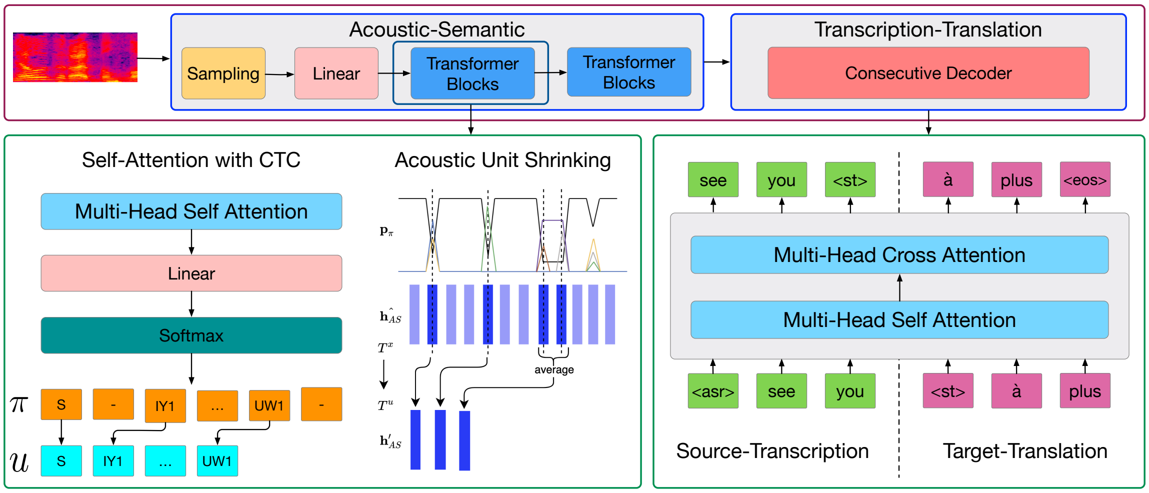

Figure 1: Overview of the proposed **COSTT**. It consists of two phases, an acoustic-semantic modeling phase (AS) and a transcription-translation phase (TT). During AS phase, CTC loss is adopted against phoneme labels corresponding to source-text. The TT phase decodes source-text and target-text in a single sequence consecutively.

domain feature of the audio feature to the preset network hidden layer size. After preprocessing, multiple Transformer blocks are stacked for acoustic feature extraction.

$$\hat{\mathbf{h}}_{AS} = \text{Attention}(\text{Linear}(\text{Down-sample}(\mathbf{x}))) \quad (1)$$

Finally, the softmax operator is applied to the result of the affine transformation to obtain the probability of the phoneme sequence. CTC loss is adopted to accelerate the convergence of acoustic modeling. CTC assumes $T_u \leq T_x$, and defines an intermediate alphabet $\mathcal{V}' = \mathcal{V} \cup \{blank\}$. A *path* π is defined as a T_x-length sequence of intermediate labels $\pi = (\pi_1, ..., \pi_{T_x}) \in \mathcal{V}'^{T_x}$. And a many-to-one mapping is defined from paths to output sequences by removing blank symbols and consecutively repeated labels.

The conditional probability of a given labelling $\mathbf{u} \in \mathcal{V}'^{T_u}$ can be modeled by marginalizing over all paths corresponding to it:

$$\begin{aligned} \log p_{ctc}(\mathbf{u}|\mathbf{x}) &= \log \sum_{\pi\in\mathcal{B}^{-1}(\mathbf{u})} p(\pi|\hat{\mathbf{h}}_{AS}) \\ &= \log \sum_{\pi\in\mathcal{B}^{-1}} \sum_{t'=1}^{t} p(\pi_{t'}, t'|\hat{\mathbf{h}}_{AS}) \end{aligned} \quad (2)$$

The distribution over the set $\mathcal{V}'^{T_x}$ of *path* π is defined by the probability of a sequence of conditionally-independent outputs, which can be calculated non-autoregressively. And $p(\pi_{t'}, t'|S)$ is computed by applying the $softmax$ function to $logits$. Finally, the objective training function during AS phase is defined as:

$$\mathcal{L}_{\text{AS}} = -\log p_{ctc}(\mathbf{u}|\mathbf{x}) \quad (3)$$

Acoustic Unit Shrinking The shrinking layer aims at reducing the potential blank frames, and repeated frames. The details can be seen in the sub-figure of the lower left of Figure 1. The method is mainly founded on the studies of Chen et al. (2016); Yi, Wang, and Xu (2019). We adopt the implementation by removing the blank frames and averaging the repeated frames. Without the interruption of blank and repeated frames, the language modeling ability should be better in theory. Blank frames can be detected according to the spike characteristics of CTC probability distribution.

$$\mathbf{h}'_{AS} = \text{Shrink}(\hat{\mathbf{h}}_{AS}, p_{ctc}(\mathbf{u}|\mathbf{x})) \quad (4)$$

Then, similarly, after shrinking, multiple Transformer blocks are stacked to extract higher-level semantic representations and result in the final output $\mathbf{h}_{AS}$.

$$\mathbf{h}_{AS} = \text{Attention}(\mathbf{h}'_{AS}) \quad (5)$$

3.3 Transcription-Translation Modeling

We jointly model the transcription and translation generation in a single shared decoder, which takes the acoustic representation $\mathbf{h}_{AS}$ as the input and generates the source text $\mathbf{z}$ and target text $\mathbf{y}$. This TT phase is stacked with T Transformer blocks, consisting of multi-head attention layers and feed-forward networks.

$$\mathbf{h}_{TT} = \text{Transformer}([\mathbf{z}, \mathbf{y}], \mathbf{h}_{AS}) \quad (6)$$

As shown in Figure 1, the decoder output is the tandem result of the transcription and translation sequences, joined by the task identificator token ("<asr>" for recognition and "<st>" for translation), marked as $[\mathbf{z}, \mathbf{y}]$. That is to say, the model is able to continuously predict the transcription sequence and the translation sequence. The training objective of the TT phase is the cross entropy between prediction sequence and target sequence.

$$\mathcal{L}_{\text{TT}} = -\log p([\mathbf{z}, \mathbf{y}]|\mathbf{x}) \quad (7)$$

Compared with the multi-task learning method, consecutive decoding can make prediction from easy (transcription)

speech	135-19215-0118.wav
phonemes	
	Y UW1 <space> M AH1 S T <space> M EY1 K <space> AH0 <space> D R IY1 M <space> W ER1 L <space> ER0 AW1 N D <space> DH AH0 <space> B R AY1 D
transcription	
	you must make a dream whirl around the bride
translation	
	il faudrait faire tourbillonner un songe autour de l' épousée .

Table 1: An example of the speech-phoneme-transcription-translation quadruples. Phonemes can be converted from the transcription text.

to hard (translation) tasks, alleviating the decoding pressure. For example, when predicting the translation sequence, since the corresponding transliteration sequence has been decoded, that is, the intermediate recognition result of the known speech translation and the source of information for decoding, the translation sequence can be improved.

Pre-train the Consecutive Decoder Generally, it is straightforward to use ASR corpus to improve the performance of ST systems, but is non-trivial to utilize MT corpus. Taking advantage of the structure of consecutive decoding, we propose a method to enhance the performance of ST systems by means of external MT paired data. Inspired by translation language modeling (TLM) in XLM (Lample and Conneau 2019), we use a masked loss function to pre-train TT phase. Specifically, we use external data in $\mathcal{A}$ to pre-train the parameters of the TT part. Different from the end-to-end training stage, there is no audio feature as input during pre-training, so cross-attention cannot attend to the output of the previous AS phase. We use an all-zero constant, marked as $\mathbf{h}_{AS_{blank}}$ to substitute the encoded representations ($\mathbf{h}_{AS}$) from TT phase to be consistent with fine-tuning. When calculating the objective function, we mask the loss for prediction of the recognition result, and make the decoder predicts the translation sequence when aware of the input of the transcription sequence. The translation loss of the TT phase during pre-training only includes the masked cross entropy:

$$\mathcal{L}_{\text{TT}_{\text{PT}}} = -\sum_{i=1}^{T_y} \log p(y_i|\mathbf{z}, y_{<i}) \tag{8}$$

We exploit joint learning to integrate our unified ST model. The total training objective is as follows:

$$\mathcal{L} = \alpha\mathcal{L}_{\text{AS}} + (1-\alpha)\mathcal{L}_{\text{TT}} \tag{9}$$

where α is a tunable parameter to balance the objectives of different phases.

4 Experiments

4.1 Dataset and Preprocessing

We conduct experiments on three popular publicly available datasets, including Augmented LibriSpeech English-French dataset (Kocabiyikoglu, Besacier, and Kraif 2018), TED English-German dataset (Jan et al. 2018) and TED English-Chinese dataset (Liu et al. 2019a).

Augmented LibriSpeech English-French Dataset Augmented LibriSpeech is built by automatically aligning e-books in French with English utterances of LibriSpeech. The dataset includes quadruplets: source audio files in English, transcriptions in English, translations in French from the alignment of e-books, and augmented translation references via Google Translate. We experiment on the 100 hours clean train set for training, with 2 hours development set and 4 hours test set, corresponding to 47,271, 1071, and 2048 utterances respectively.

TED English-German Dataset English-German TED is the KIT speech translation corpus, which is built by automatically aligning English audios with SRT transcripts for English and German from TED. The raw data, including long wave files, English transcriptions, and the corresponding German translations, are segmented with time stamps and made forced alignments using the gentle tool kit[1], according to the officially released version. We utilize the attached timestamps to segment a raw long audio into chunks and remove samples missing the target language translation. It should be noted that some transcriptions are not aligned with the corresponding audio well. Noisy data is harmful to models' performance, which can be avoided by data filtering, re-alignment and re-segmentation (Liu et al. 2018). In this paper, we directly use the original data as training data to verify our method, with a size of 272 hours and 171,121 segmentations. We use *dev2010* as validation set, and *tst2010, tst2013, tst2014, tst2015* as test set, corresponding to 653, 1337, 793, 957 and 1177 utterances respectively.

TED English-Chinese Dataset English-Chinese TED is crawled from TED website[2] and released by (Liu et al. 2019a) as a benchmark for speech translation from English audio to Chinese text. Following the previous work (Liu et al. 2019a), we use dev2010 as development set and tst2015 as test set. The raw long audio is segmented based on timestamps for complete semantic information. Finally, we get 524 hours train set, 1.5 hours test set and 2.5 hours test set, corresponding to 308,660, 835, 1223 utterances respectively.

WMT Machine Translation Corpus We use WMT14[3] English-to-French and English-to-German training data, and WMT20[4] English-to-Chinese training data as the external MT parallel corpus ($\in \mathcal{A}$) in the expanded experimental setting for broad reproducibility. We pre-processed all of the data of specific language pairs, and filtered sentence pairs whose total length exceeds 500. We shuffled the data and

[1] https://github.com/lowerquality/gentle

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

[3] https://www.statmt.org/wmt14/translation-task.html

[4] http://www.statmt.org/wmt20/translation-task.html

Method	Enc Pre-train (speech data)	Dec Pre-train (text data)	BLEU
MT system			
Transformer MT	-	-	21.51
Base setting			
LSTM ST (Bérard et al. 2018)	✗	✗	12.90
+pre-train+multitask (Bérard et al. 2018)	✓	✓	13.40
LSTM ST+pre-train (Inaguma et al. 2020)	✓	✓	16.68
Transformer+pre-train (Liu et al. 2019a)	✓	✓	14.30
+knowledge distillation (Liu et al. 2019a)	✓	✓	17.02
TCEN-LSTM (Wang et al. 2019)	✓	✓	17.05
Transformer+ASR pre-train (Wang et al. 2020)	✓	✗	15.97
+curriculum pre-train (Wang et al. 2020)	✓	✗	17.66
COSTT	✗	✗	**17.83**
Expanded setting			
LSTM+pre-train+SpecAugment (Bahar et al. 2019)	✓(236h)	✓	17.00
Multi-task+pre-train (Inaguma et al. 2019)	✓(472h)	✗	17.60
Transformer+ASR pre-train (Wang et al. 2020)	✓(960h)	✗	16.90
+curriculum pre-train (Wang et al. 2020)	✓(960h)	✗	18.01
COSTT	✓(100h)	✓(1M)	**18.23**

Table 2: Performance for MT, ST tasks on Augmented Librispeech English-French test set. Our proposed **COSTT** achieves the best results in both base and expanded settings.

randomly selected a subset of 1 million for the following experiments and analysis.

4.2 Experimental Setup

Our acoustic features are 80-dimensional log-Mel filter banks extracted with a step size of 10ms and window size of 25ms and extended with mean sub-traction and variance normalization. The features are stacked with 5 frames to the right. For all source language text data, we lower case all the texts, tokenize and remove the punctuation to make the data more consistent with the output of ASR. For target French and German text data, we lower case all the texts, tokenize and apply normalize punctuations with the Moses scripts[5]. For target Chinese text data, we use the raw released segmented results. For English-French and English-German datasets, we apply BPE[6] (Sennrich, Haddow, and Birch 2015) to the combination of source and target text to obtain shared subword units. And for English-Chinese dataset, we apply BPE to the source text and target text respectively. The number of merge operations in BPE is set to 8k for all datasets. In order to simplify, we use the open-source grapheme to phoneme tool[7] to map the transcription to the phoneme sequence (An example in Table 1). The alphabet of labels $\mathcal{V}$ includes the union of subword vocabulary and phoneme vocabulary, plus a few special symbols (including "<asr>", "<st>" and "blank"). For English-French and English-German corpora, we report case-insensitive BLEU scores by `multi-bleu.pl`[8] script for the evaluation of translation. And for English-Chinese corpus, we report character-level BLEU scores. We use word error rates (WER) and phoneme error rates (PER) to evaluate the transcription and phoneme sequences, respectively.

We use a similar hyperparameter setting with the base Transformer model (Vaswani et al. 2017). For English-French Dataset, the number of transformer blocks is set to 8 and 4 for the acoustic-semantic (AS) phase and the transcription-translation (TT) phase, respectively. For English-German and English-Chinese Datasets, the number of transformer blocks is set to 12 and 6 for the acoustic-semantic (AS) phase and the transcription-translation (TT) phase, respectively. And phoneme supervision is added to the middle layer of AS phase for all datasets. SpecAugment strategy (Park et al. 2019) is adopted to avoid overfitting with frequency masking (F = 30, mF = 2) and time masking (T = 40, mT = 2). All samples are batched together with 20000-frame features by an approximate feature sequence length during training. We train our models on 1 NVIDIA V100 GPUs with a maximum number of 400k training steps. We use the greedy search decoding strategy for our experimental settings. The maximum decoding length is set to 500 for our models with consecutive decoding and 250 for other methods on all datasets. α in Equation 9 is set to 0.5 for all datasets (We have searched the value of α using a step of 0.2.). We design different workflows for our method training from scratch and training with pre-training the consecutive decoder. More details are in the Appendix.

5 Results

5.1 Baselines

We compare with systems in different settings:

[5] https://github.com/moses-smt/mosesdecoder

[6] https://github.com/rsennrich/subword-nmt

[7] https://github.com/Kyubyong/g2p

[8] https://github.com/moses-smt/mosesdecoder/scripts/generic/multi-bleu.perl

Method	Enc Pre-train (speech data)	Dec Pre-train (text data)	tst2010	tst2013	tst2014	tst2015	Avg
MT system							
Transformer MT	-	-	25.72	27.87	22.23	23.58	24.85
Base setting							
ESPnet (Inaguma et al. 2020)	✗	✗	13.77	12.50	11.50	12.68	12.61
+enc pre-train	✓	✗	14.46	13.12	11.62	11.30	12.63
+enc dec pre-train	✓	✓	14.98	13.54	12.33	11.67	13.13
Transformer+ASR pre-train (Wang et al. 2020)	✓	✗	-	15.35	-	-	-
+curriculum pre-train (Wang et al. 2020)	✓	✗	-	16.27	-	-	-
COSTT	✗	✗	**19.54**	**16.30**	**14.53**	**16.42**	**16.70**
Expanded setting							
Multi-task+pre-train (Inaguma et al. 2019)	✓(472h)	✗	-	14.60	-	-	-
CL-fast* (Kano, Sakti, and Nakamura 2018)	✓(479h)	✗	-	14.33	-	-	-
TCEN-LSTM (Wang et al. 2019)	✓(479h)	✓(40M)	17.61	17.67	15.73	14.94	16.49
Transformer+curriculum pre-train (Wang et al. 2020)	✓(479h)	✓(4M)	-	18.15	-	-	-
COSTT	✓(272h)	✓(1M)	**21.31**	**18.63**	**16.20**	**17.72**	**18.47**

Table 3: Performance (BLEU) for MT, ST tasks on English-German TED test sets. *: re-implemented by Wang et al. (2020). Our proposed **COSTT** consistently achieves the best performance across all test sets.

Method	Enc Pre-train (speech data)	Dec Pre-train (text data)	BLEU
MT system			
Transformer MT	-	-	23.19
Base setting			
Transformer+pre-train (Liu et al. 2019a)	✓	✓	16.80
+knowledge distillation (Liu et al. 2019a)	✓	✓	19.55
Multi-task+pre-train* (Inaguma et al. 2019)(re-implemented)	✓	✗	20.45
COSTT	✗	✗	**21.12**
Expanded setting			
COSTT	✓(524h)	✓(1M)	**22.16**

Table 4: Performance for MT, ST tasks on English-Chinese TED test set. *: re-implemented. Our proposed **COSTT** achieves the best results.

Base setting: ST models are trained with only ST triple corpus.

Expanded setting: ST models are trained with ST triple corpus augmented with external ASR and MT corpus. In the context of expanded setting, Bahar et al. (2019) apply the SpecAugment (Park et al. 2019) with a total of 236h of speech for ASR pre-training. Inaguma et al. (2019) combine three ST datasets of 472h training data to train a multilingual ST model. Wang et al. (2019) introduce an additional 272h ASR corpus and 41M parallel data from WMT18 to enhance the ST.

MT system: Text translation models are trained with manual transcribed transcription-translation pairs, which can be regarded as the upper bound of speech translation tasks.

5.2 Main Results

We conduct experiments on three public datasets.

	Method	BLEU
$En \rightarrow Fr$	Pipeline	17.58
	COSTT	**18.23**
$En \rightarrow De$	Pipeline	17.40
	COSTT	**18.63**
$En \rightarrow Zh$	Pipeline	21.36
	COSTT	**22.16**

Table 5: **COSTT** versus cascaded systems on Augmented Librispeech En-Fr test set, En-De TED tst2013 set and En-Zh tst2015 set. "Pipeline" systems consist of separate ASR and MT models trained independently.

Results on Augmented Librispeech For En-Fr experiments, we compared the performance with existing end-to-end methods in Table 2. Clearly, **COSTT** outscored the previous best results by more than 0.5 BLEU in the base setting and 0.6 BLEU in the expanded setting, respectively. Specifically, in the base setting, the model we proposed outperformed ESPnet, which was equipped with both a well pre-trained encoder and decoder. We also achieved better results than a knowledge distillation baseline in which an MT model was introduced to teach the ST model (Liu et al. 2019a). Different from previous work, **COSTT** can make full use of the machine translation corpus. With an additional 1 million sentence pairs, we achieve +0.7 BLEU score improvements (17.51 v.s. 18.23). This proposal promises great potential for the application of the **COSTT**. In a nutshell, simple yet effective, **COSTT** achieves the best performance in this benchmark dataset in terms of BLEU.

Results on English-German TED For En-De experiments, we compared the performance with existing end-to-end methods in Table 3. Unlike that of Librispeech English-French, this dataset is noisy, and the transcriptions do not align well with the corresponding audios. As a result, there is a wide gap between the performance of the ST system and the upper bound of the ST (MT). We suppose it would be more beneficial to carry out data filtering. Overall, our method had +0.5 BLEU score advantage as compared to previous competitors on tst2013 in the expanded setting. This trend is consistent with that in the Librispeech dataset.

Results on English-Chinese TED For En-Zh experiments, we compared the performance with existing end-to-end methods in Table 4. **COSTT** outperformed the previous best methods obviously by more than 1.5 BLEU in the base setting and 1.7 BLEU in the expanded setting, respectively. Especially, under the base setting, **COSTT** exceeded the Transformer-based ST model augmented by knowledge distillation with a big margin, proving the validity of our unified framework.

Comparison with Cascaded Systems In Table 5, we compare the performance of our E2E models with the cascaded systems. It shows that E2E models are outstanding or comparable on all En→Fr/De/Zh tasks, proving our method's capacity to combine the separate ASR and MT tasks in a model.

5.3 Ablation Study

We use an ablation study to evaluate the importance of different modules in our methods. The results in Table 6 show that all the methods adapted are positive for the model performance, and the benefits of different parts can be superimposed. Models with consecutive decoding are able to predict both the recognition and translation, for which we also report WER and PER to evaluate the performance of different modeling phase. It has been proved that consecutive decoding brings the gain of 1 BLEU compared with the base model and pre-training decoder can bring improvements to all three metrics.

5.4 Case Study on English-French

The cases in Table 7 shows that **COSTT** has obvious structural advantages in solving missed translation, mistranslation, and fault tolerance. For instance: #1, the base model missed the translation of "yes" in the audio, whereas our method produced a completely correct translation. After listening to the original audio, it is suspected that the missing translation is due to an unusual pause between "doctor" and "yes". #2, the base model mistranslated the "aboard" in the audio into "vers l' avant"("forward" in English), yet our method could correctly translate it into "a bord" based on the correct transcription prediction. The reason for the mistranslation may be that the audio clips are pronounced similarly, thus confusing the translation model. #3, the base model translated erroneously most of the content, and our model also predicted "today" in the audio as "to day". However, in the end, our method was able to predict the translation result completely and correctly.

	BLEU↑	WER↓	PER↓
COSTT	**18.23**	**14.60**	**10.30**
w/o PT Dec	17.51	15.30	11.90
w/o CD	16.57	-	-
w/o Shrink	16.40	-	-
w/o AS loss *	15.48	-	-
w/o AS loss	11.24	-	-

Table 6: Benefits of each component in **COSTT** on En-Fr test set. "PT Dec" stands for pre-training the successive decoder. "CD" represents using the consecutive decoder. "*" means using ASR pre-training as initialization.

Speech #1	766-144485-0043.wav
Transcript	said the doctor yes
Target	dit le docteur , oui .
Base ST	dit le docteur .
COSTT	<asr> said the doctor yes <ast> dit le docteur , oui .
Speech #2	2488-36617-0066.wav
Transcript	i rushed aboard
Target	je me précipitai à bord.
Base ST	je me précipitai vers l' avant .
COSTT	<asr> i rushed aboard <ast> je me précipitai à bord .
Speech #3	766-144485-0098.wav
Transcript	is there any news today
Target	y a-t-il des nouvelles aujourd' hui ?
Base ST	est-ce que j' ai déjà utilisé aujourd' hui ?
COSTT	<asr> is there any news to day <ast> y a-t-il des nouvelles aujourd' hui ?

Table 7: Examples of speech translation generated by **COSTT** and the baseline ST model. Words in red highlight the difference. Words underlined, as generated by **COSTT**, contributes to the improved translation results.

6 Conclusion

We propose **COSTT**, a novel and unified training framework for jointly speech recognition and speech translation. We use the consecutive decoding strategy to realize the sequential prediction of the transcription and translation sequences, which is more in line with human cognitive principles. By pre-training the decoder, we can directly mask better use of the parallel data of MT. Additionally, CTC auxiliary loss, and *shrinking* operation strategies are adopted to enhance our method benefiting from the flexible structure. Experimental results prove the effectiveness of our framework and it has great prospects for promoting the application of speech translation.

References

Anastasopoulos, A.; and Chiang, D. 2018. Tied multitask learning for neural speech translation. *arXiv preprint arXiv:1802.06655* .

Bahar, P.; Bieschke, T.; and Ney, H. 2019. A comparative study on end-to-end speech to text translation. *arXiv preprint arXiv:1911.08870* .

Bahar, P.; Zeyer, A.; Schlüter, R.; and Ney, H. 2019. On using specaugment for end-to-end speech translation. *arXiv preprint arXiv:1911.08876* .

Bansal, S.; Kamper, H.; Livescu, K.; Lopez, A.; and Goldwater, S. 2018. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. *arXiv preprint arXiv:1809.01431* .

Beck, D.; Cohn, T.; and Haffari, G. 2019. Neural Speech Translation using Lattice Transformations and Graph Networks. In *Proceedings of the Thirteenth Workshop on Graph-Based Methods for Natural Language Processing (TextGraphs-13)*, 26–31.

Bérard, A.; Besacier, L.; Kocabiyikoglu, A. C.; and Pietquin, O. 2018. End-to-end automatic speech translation of audiobooks. In *ICASSP*, 6224–6228. IEEE.

Bérard, A.; Pietquin, O.; Servan, C.; and Besacier, L. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *arXiv preprint arXiv:1612.01744* .

Chen, Z.; Zhuang, Y.; Qian, Y.; and Yu, K. 2016. Phone synchronous speech recognition with ctc lattices. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* 25(1): 90–101.

Cheng, Q.; Fang, M.; Han, Y.; Huang, J.; and Duan, Y. 2019. Breaking the Data Barrier: Towards Robust Speech Translation via Adversarial Stability Training. *arXiv preprint arXiv:1909.11430* .

Cheng, Y.; Tu, Z.; Meng, F.; Zhai, J.; and Liu, Y. 2018. Towards robust neural machine translation. *arXiv preprint arXiv:1805.06130* .

Chuang, S.-P.; Sung, T.-W.; Liu, A. H.; and Lee, H.-y. 2020. Worse WER, but Better BLEU? Leveraging Word Embedding as Intermediate in Multitask End-to-End Speech Translation. *arXiv preprint arXiv:2005.10678* .

Di Gangi, M. A.; Negri, M.; Cattoni, R.; Roberto, D.; and Turchi, M. 2019. Enhancing transformer for end-to-end speech-to-text translation. In *Machine Translation Summit XVII*, 21–31. European Association for Machine Translation.

Di Gangi, M. A.; Negri, M.; and Turchi, M. 2019. Adapting transformer to end-to-end spoken language translation. In *INTERSPEECH 2019*, 1133–1137. International Speech Communication Association (ISCA).

Di Gangi, M. A.; Nguyen, V.-N.; Negri, M.; and Turchi, M. 2020. Instance-based Model Adaptation for Direct Speech Translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7914–7918. IEEE.

Duong, L.; Anastasopoulos, A.; Chiang, D.; Bird, S.; and Cohn, T. 2016. An attentional model for speech translation without transcription. In *NAACL*, 949–959.

Fitzgerald, E.; Hall, K. B.; and Jelinek, F. 2009. Reconstructing false start errors in spontaneous speech text .

Fügen, C. 2008. *A system for simultaneous translation of lectures and speeches*. Ph.D. thesis, Verlag nicht ermittelbar.

Graves, A.; Fernández, S.; Gomez, F.; and Schmidhuber, J. 2006. Connectionist temporal classification: labelling unsegmented sequence data with recurrent neural networks. In *ICML*, 369–376. ACM.

Inaguma, H.; Duh, K.; Kawahara, T.; and Watanabe, S. 2019. Multilingual end-to-end speech translation. *arXiv preprint arXiv:1910.00254* .

Inaguma, H.; Kiyono, S.; Duh, K.; Karita, S.; Soplin, N. E. Y.; Hayashi, T.; and Watanabe, S. 2020. ESPnet-ST: All-in-One Speech Translation Toolkit. *arXiv preprint arXiv:2004.10234* .

Indurthi, S.; Han, H.; Lakumarapu, N. K.; Lee, B.; Chung, I.; Kim, S.; and Kim, C. 2019. Data Efficient Direct Speech-to-Text Translation with Modality Agnostic Meta-Learning. *arXiv preprint arXiv:1911.04283* .

Jan, N.; Cattoni, R.; Sebastian, S.; Cettolo, M.; Turchi, M.; and Federico, M. 2018. The iwslt 2018 evaluation campaign. In *International Workshop on Spoken Language Translation*, 2–6.

Jan, N.; Cattoni, R.; Sebastian, S.; Negri, M.; Turchi, M.; Elizabeth, S.; Ramon, S.; Loic, B.; Lucia, S.; and Federico, M. 2019. The IWSLT 2019 evaluation campaign. In *16th International Workshop on Spoken Language Translation 2019*.

Jia, Y.; Johnson, M.; Macherey, W.; Weiss, R. J.; Cao, Y.; Chiu, C.-C.; Ari, N.; Laurenzo, S.; and Wu, Y. 2019. Leveraging weakly supervised data to improve end-to-end speech-to-text translation. In *ICASSP*, 7180–7184. IEEE.

Kano, T.; Sakti, S.; and Nakamura, S. 2018. Structured-based curriculum learning for end-to-end english-japanese speech translation. *arXiv preprint arXiv:1802.06003* .

Kocabiyikoglu, A. C.; Besacier, L.; and Kraif, O. 2018. Augmenting Librispeech with French translations: A multimodal corpus for direct speech translation evaluation. *arXiv preprint arXiv:1802.03142* .

Lample, G.; and Conneau, A. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291* .

Liu, D.; Liu, J.; Guo, W.; Xiong, S.; Ma, Z.; Song, R.; Wu, C.; and Liu, Q. 2018. The USTC-NEL Speech Translation system at IWSLT 2018. *arXiv preprint arXiv:1812.02455* .

Liu, D.; Spanakis, G.; and Niehues, J. 2020. Low-Latency Sequence-to-Sequence Speech Recognition and Translation by Partial Hypothesis Selection. *arXiv preprint arXiv:2005.11185* .

Liu, F.-H.; Gu, L.; Gao, Y.; and Picheny, M. 2003. Use of statistical N-gram models in natural language generation for machine translation. In *2003 IEEE International Conference on Acoustics, Speech, and Signal Processing, 2003. Proceedings.(ICASSP'03).*, volume 1, I–I. IEEE.

Liu, Y.; Xiong, H.; He, Z.; Zhang, J.; Wu, H.; Wang, H.; and Zong, C. 2019a. End-to-End Speech Translation with Knowledge Distillation. *arXiv preprint arXiv:1904.08075* .

Liu, Y.; Zhang, J.; Xiong, H.; Zhou, L.; He, Z.; Wu, H.; Wang, H.; and Zong, C. 2019b. Synchronous Speech Recognition and Speech-to-Text Translation with Interactive Decoding. *arXiv preprint arXiv:1912.07240* .

Matusov, E.; Mauser, A.; and Ney, H. 2006. Automatic sentence segmentation and punctuation prediction for spoken language translation. In *International Workshop on Spoken Language Translation (IWSLT) 2006*.

Park, D. S.; Chan, W.; Zhang, Y.; Chiu, C.-C.; Zoph, B.; Cubuk, E. D.; and Le, Q. V. 2019. Specaugment: A simple data augmentation method for automatic speech recognition. *arXiv preprint arXiv:1904.08779* .

Pino, J.; Puzon, L.; Gu, J.; Ma, X.; McCarthy, A. D.; and Gopinath, D. 2019a. Harnessing indirect training data for end-to-end automatic speech translation: Tricks of the trade. In *Proceedings of the 16th International Workshop on Spoken Language Translation (IWSLT)*.

Pino, J.; Puzon, L.; Gu, J.; Ma, X.; McCarthy, A. D.; and Gopinath, D. 2019b. Leveraging Out-of-Task Data for End-to-End Automatic Speech Translation. *arXiv preprint arXiv:1909.06515* .

Salazar, J.; Kirchhoff, K.; and Huang, Z. 2019. Self-attention networks for connectionist temporal classification in speech recognition. In *ICASSP*, 7115–7119. IEEE.

Salesky, E.; and Black, A. W. 2020. Phone Features Improve Speech Translation. *arXiv preprint arXiv:2005.13681* .

Salesky, E.; Burger, S.; Niehues, J.; and Waibel, A. 2018. Towards fluent translations from disfluent speech. In *2018 IEEE Spoken Language Technology Workshop (SLT)*, 921–926. IEEE.

Salesky, E.; Sperber, M.; and Black, A. W. 2019. Exploring phoneme-level speech representations for end-to-end speech translation. *arXiv preprint arXiv:1906.01199* .

Salesky, E.; Sperber, M.; and Waibel, A. 2019. Fluent translations from disfluent speech in end-to-end speech translation. *arXiv preprint arXiv:1906.00556* .

Sennrich, R.; Haddow, B.; and Birch, A. 2015. Neural machine translation of rare words with subword units. *arXiv preprint arXiv:1508.07909* .

Sperber, M.; Neubig, G.; Niehues, J.; and Waibel, A. 2017. Neural lattice-to-sequence models for uncertain inputs. *arXiv preprint arXiv:1704.00559* .

Sperber, M.; Neubig, G.; Niehues, J.; and Waibel, A. 2019a. Attention-Passing Models for Robust and Data-Efficient End-to-End Speech Translation. *TACL* 7: 313–325.

Sperber, M.; Neubig, G.; Pham, N.-Q.; and Waibel, A. 2019b. Self-Attentional Models for Lattice Inputs. *arXiv preprint arXiv:1906.01617* .

Sperber, M.; and Paulik, M. 2020. Speech Translation and the End-to-End Promise: Taking Stock of Where We Are. *arXiv preprint arXiv:2004.06358* .

Stoian, M. C.; Bansal, S.; and Goldwater, S. 2020. Analyzing ASR pretraining for low-resource speech-to-text translation. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 7909–7913. IEEE.

Sung, T.-W.; Liu, J.-Y.; Lee, H.-y.; and Lee, L.-s. 2019. Towards End-to-end Speech-to-text Translation with Two-pass Decoding. In *ICASSP*, 7175–7179. IEEE.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, Ł.; and Polosukhin, I. 2017. Attention is all you need. In *NIPS*, 5998–6008.

Vila, L. C.; Escolano, C.; Fonollosa, J. A.; and Costa-jussà, M. R. 2018. End-to-End Speech Translation with the Transformer. In *IberSPEECH*, 60–63.

Vydana, H. K.; Karafi'at, M.; Zmolikova, K.; Burget, L.; and Cernocky, H. 2020. Jointly Trained Transformers models for Spoken Language Translation. *arXiv preprint arXiv:2004.12111* .

Wang, C.; Wu, Y.; Liu, S.; Yang, Z.; and Zhou, M. 2019. Bridging the Gap between Pre-Training and Fine-Tuning for End-to-End Speech Translation. *arXiv preprint arXiv:1909.07575* .

Wang, C.; Wu, Y.; Liu, S.; Zhou, M.; and Yang, Z. 2020. Curriculum Pre-training for End-to-End Speech Translation. *arXiv preprint arXiv:2004.10093* .

Weiss, R. J.; Chorowski, J.; Jaitly, N.; Wu, Y.; and Chen, Z. 2017. Sequence-to-sequence models can directly translate foreign speech. *arXiv preprint arXiv:1703.08581* .

Yi, C.; Wang, F.; and Xu, B. 2019. ECTC-DOCD: An End-to-end Structure with CTC Encoder and OCD Decoder for Speech Recognition. *Proc. Interspeech 2019* 4420–4424.

Zhang, P.; Chen, B.; Ge, N.; and Fan, K. 2019. Lattice transformer for speech translation. *arXiv preprint arXiv:1906.05551* .

7 Appendix

7.1 Workflows for Different Settings

We design different workflows for our method training from scratch (marked as workflow #1, seen in Algorithm 1) and training with pre-training the consecutive decoder (marked as workflow #2, seen in Algorithm 2). For workflow #1, ST model is totally supervised training from scratch with $(\mathbf{x}, \mathbf{u}, \mathbf{y}) \in \mathcal{S}$, as Equation 9. For workflow #2, training is done as the following three steps: *a)* pre-training the consecutive decoder (ConDec) with $(\mathbf{z}', \mathbf{y}') \in \mathcal{A}$ with cross-entropy, as Equation 8. *b)* pre-training the acoustic modeling (AM) with $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}$ with CTC loss, as Equation 3. *c)* fine-tuning the ST model with $(\mathbf{x}, \mathbf{u}, \mathbf{z}, \mathbf{y}) \in \mathcal{S}$, as Equation 9 (the same as workflow #1). Workflow #2 is determined after many attempts to better avoid the catastrophic forgetting of pre-trained knowledge. Figure 2 shows the convergence curve on the English-French validation set of the two workflows. It proves that workflow #2 with pre-training the consecutive decoder can get a better initialization and converge better benefiting from our flexible model structure.

Algorithm 1 Workingflow #1 for our ST models

```
# training from scratch ($\theta^0_{AS} \rightarrow \theta^1_{AS}, \theta^0_{TT} \rightarrow \theta^1_{TT}$)
while not converged do
    supervised training ST with $(\mathbf{x}, \mathbf{u}, \mathbf{y}) \in \mathcal{S}$
end while
return ST with $\theta^1_{AS}, \theta^1_{TT}$
```

Algorithm 2 Workingflow #2 for our ST models

```
# pre-training ConDec ($\theta^0_{AS} \rightarrow \theta^0_{AS}, \theta^0_{TT} \rightarrow \theta^1_{TT}$)
while not converged do
    CE loss guided supervised training ConDec with $(\mathbf{z}', \mathbf{y}') \in \mathcal{A}$
end while
# pre-training AM ($\theta^0_{AS} \rightarrow \theta^1_{AS}, \theta^1_{TT} \rightarrow \theta^1_{TT}$)
while not converged do
    CTC loss guided supervised training AM with $(\mathbf{x}, \mathbf{u}) \in \mathcal{S}$
end while
# fine-tuning ST ($\theta^1_{AS} \rightarrow \theta^2_{AS}, \theta^1_{TT} \rightarrow \theta^2_{TT}$)
while not converged do
    Supervised training ST with $(\mathbf{x}, \mathbf{u}, \mathbf{z}, \mathbf{y}) \in \mathcal{S}$
end while
return ST with $\theta^2_{AS}, \theta^2_{TT}$
```

7.2 Parameters of ST systems

The parameter sizes of different systems are shown in Table 8. The pipeline system needs a separate ASR model and MT model, so its parameters are doubled. Our method **COSTT** only needs the same parameters as the vanilla end-to-end model, but it can achieve superior performance thanks to the consecutive decoding mechanism.

7.3 Effects of Shrinking Mechanism

In order to verify whether the shrinking mechanism has achieved the expected effect, we collected the sequence length of the encoded hidden layer before and after shrinking and the length distribution of the gold phoneme sequence. As shown in Figure 3, the sequence length of the shrunk acoustic unit and the distribution of phoneme length are almost the same. According to statistics in Table 9, for more than 90% of the samples, the absolute error between the length of the shrunk acoustic unit and the length of the gold phoneme sequence is within 3. Moreover, the length of the shrunk acoustic unit is significantly reduced compared to the length of the original acoustic feature. The results show that the shrinking mechanism can detect blank frames and repeated frames well, while reducing the computational resources and preventing memory overflow.

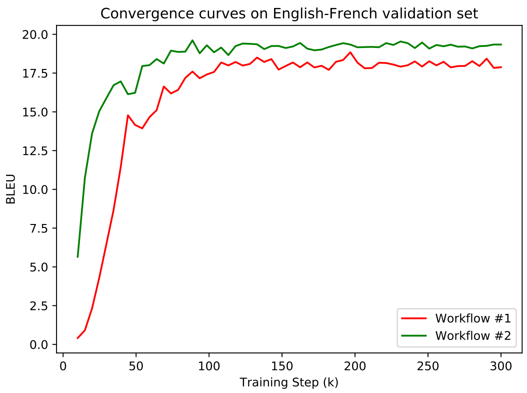

Figure 2: BLEU scores on Augmented Librispeech validation set for different workflows.

Model	Params
Pipeline	110M
E2E	55M
COSTT (12 L)	55M
COSTT (18 L)	76M

Table 8: Statistics of parameters of different ST systems. E2E: the vanilla end-to-end ST system.

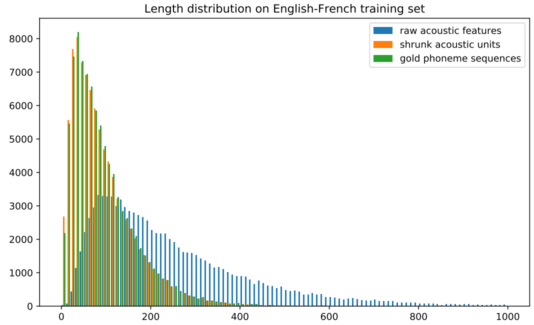

Figure 3: Length distribution of the raw acoustic features, the shrunk acoustic units and the gold phoneme sequences on English-French training set

7.4 Effects of Layers after Shrinking

As mentioned in Section 3.2, our model stacks additional Transformer blocks after the shrinking operation. We have conducted simplified experiments on the English-French dataset with a vanilla speech translation model without con-

Error Range	0	1	2	3	4	5	6	7	8	9
Probability	0.32	0.66	0.83	0.91	0.95	0.97	0.98	0.99	0.99	0.99

Table 9: Statistics of the absolute error between the length of shrunk acoustic unit and the length of the gold phoneme sequence.

$6Enc + 6Dec$	12.70
$6Enc\_shrinking + 6Dec$	11.34
$6Enc\_shrinking + 6Enc + 6Dec$	16.46

Table 10: The number represents the layers of Transformer block contained in the corresponding module.

secutive decoding to demonstrate the importance of the additional encoding layers after shrinking. The output of the encoded layer uses the CTC loss as the supervision, and we use the subword of transcriptions in the source language as the acoustic labels. Results can be seen in Table 10. The experimental results show that directly inputting the shrunk encoded output to the decoder will cause performance loss. And stacking additional encoding layers after shrinking can bring significant performance improvements. We conjecture that there is a lack of semantic encoding modules between acoustic encoding and linguistic decoding. In addition, the relationship between the hidden states after shrinking has changed a lot, and an additional network structure is required to re-extract high-level encoded features.

7.5 Compared with 3-stage Pipeline

In the case study of Table 7, we have listed some examples of errors in transcription recognition, but COSTT can still correctly predict the translation sequence, which proves that COSTT can solve the error propagation problem to some extent. In a pipeline system that includes the phoneme stage, the phoneme recognition error will also lead to error propagation. But in **COSTT**, the phoneme sequence is only the intermediate supervision used during training, not necessary during inferring. Moreover, end-to-end training can alleviate the error propagation between different stages. We believe that the more stages, the greater the advantage of our method. We have built a 3-stage system consisting of acoustics-to-phoneme (A2P), phoneme-to-transcript (P2T), and transcript-to-translation (T2T) stages. A2P is a phoneme recognition model based on the CTC loss function, which uses phoneme error rate (PER) to evaluate performance (the lower the better). Both P2T and T2T use the sequence-to-sequence model based on Transformer and BLEU is the evaluation criterion (the higher the better). The performance of each module is shown in Table 11. The performance of different systems in Table 12 prove that with the increase of stages, the problem of error propagation becomes more and more serious, which shows the benefits of the **COSTT** method.

Stage	A2P(PER)	P2T(BLEU)	T2T(BLEU)
Performance	10.30	92.08	21.51

Table 11: Performance of each module of our 3-stage Pipeline.

System	BLEU
3-stage Pipeline	12.22
2-stage Pipeline	17.58
COSTT	**18.23**

Table 12: **COSTT** versus 2-stage Pipeline and 3-stage Pipeline on Augmentated Librispeech En-Fr test set.