End-to-End Speech Translation with Adversarial Training

Xuancai Li[1], **Kehai Chen**[2], **Tiejun Zhao**[1] **and Muyun Yang**[1]
[1] Harbin Institute of Technology, Harbin, China
[2]National Institute of Information and Communications Technology, Kyoto, Japan
xcli@hit-mtlab.net, khchen@nict.go.jp, {tjzhao,yangmuyun}@hit.edu.cn

Abstract

End-to-end speech translation usually leverages audio-to-text parallel data to train an available speech translation model which has shown impressive results on various speech translation tasks. Due to the artificial cost of collecting audio-to-text parallel data, the speech translation is a natural low-resource translation scenario, which greatly hinders its improvement. In this paper, we proposed a new adversarial training method to leverage target monolingual data to relieve the low-resource shortcoming of speech translation. In our method, the existing speech translation model is considered as a Generator to gain a target language output, and another neural Discriminator is used to guide the distinction between outputs of speech translation model and true target monolingual sentences. Experimental results on the CCMT 2019-BSTC dataset speech translation task demonstrate that the proposed methods can significantly improve the performance of the end-to-end speech translation.

1 Introduction

Typically, a traditional speech translation (ST) system usually consists of two components: an automatic speech recognition (ASR) model and a machine translation (MT) model. Firstly, the speech recognition module transcribes the source language speech into the source language utterances (Chan et al., 2016; Chiu et al., 2018). Secondly, the machine translation module translates the source language utterances into the target language utterances (Bahdanau et al., 2014). Due to the success of end-to-end approaches in both automatic speech recognition and machine translation, researchers are increasingly interested in end-to-end speech translation. And, it has shown impressive results on various speech translation tasks (Duong et al., 2016; Bérard et al., 2016, 2018).

However, due to the artificial cost of collecting audio-to-text parallel data, speech translation is a natural low-resource translation scenario, which greatly hinders its improvement. Actually, the audio-to-text parallel data has only tens to hundreds of hours which are equivalent to about hundreds of thousands of bilingual sentence pairs. Thus, it is far from enough for the training of a high-quality speech translation system compare to bilingual parallel data of millions or even tens of millions for training a high-quality text-only NMT. Recently, there have some recent works that explore to address this issue. Bansal et al. (2018) pre-trained an ASR model on high-resource data, and then fine-tuned the ASR model for low-resource scenarios. Weiss et al. (2017) and Anastasopoulos and Chiang (2018) proposed multi-task learning methods to train the ST model with ASR, ST, and NMT tasks simultaneously. Liu et al. (2019) proposed a Knowledge Distillation approach which utilizes a text-only MT model to guide the ST model because there is a huge performance gap between end-to-end ST and MT model. Despite their success, these approaches still need additional labeled data, such as the source language speech, source language transcript, and target language translation.

In this paper, we proposed a new adversarial training method to leverage target monolingual data to relieve the low-resource shortcoming of end-to-end speech translation. The proposed method consists of a generator model and a discriminator model. Specifically, the existing speech translation model is considered as a Generator to gain a target language output, and another neural Discriminator is used to guide the distinction between outputs of speech translation model and true target monolingual sentences. In particular, the Generator and the Discriminator

Proceedings of the 1st Workshop on Automatic Simultaneous Translation, pages 10–14

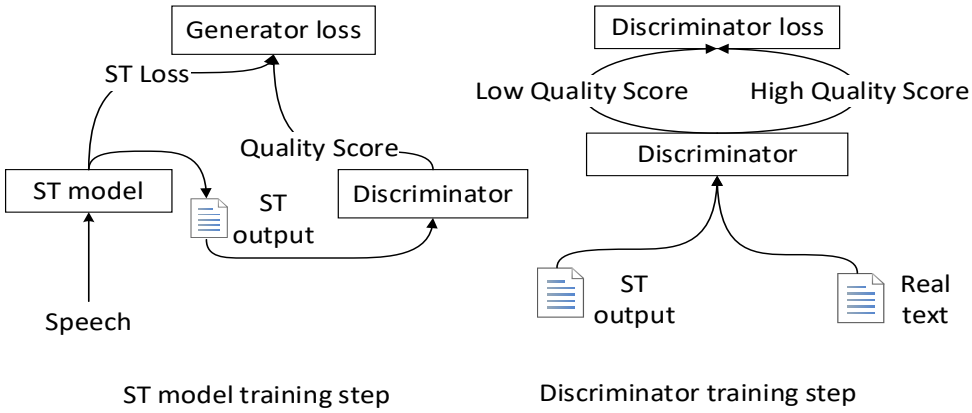

Figure 1: Proposed end-to-end speech translation with adversarial training

are trained iteratively to challenge and learn from each other step by step to gain a better speech translation model. Experimental results on CCMT 2019-BSTC dataset speech translation task demonstrate that the proposed methods can significantly improve the performance of the end-to-end speech translation system.

2 Proposed Method

The framework for the method of adversarial training consists of a generator and a discriminator. In this paper, Generator is the existing end-to-end ST model, which is based on the encoder-decoder model with an attention mechanism (Bérard et al., 2016). The discriminator is a model based on a convolutional neural network, and the output is a quality score. The discriminator is aiming to get higher quality scores for real text and lower quality scores for the output of the ST model in the discriminator training step. In other words, the discriminator is expected to distinguish the input text as much as possible. Meanwhile, our method can not only leverage the ground truth to supervise the training of ST model, but also make use of the discriminator to enhance the output of the ST model by using target monolingual data, as shown in Figure 1.

2.1 Generator

For the end-to-end speech translate, we chose an encoder-decoder model with attention. It takes as an input sequence of audio features $x = (x_1, x_2, \cdots, x_t)$ and a output sequence of words $y = (y_1, y_2, \cdots, y_m)$. The speech encoder is a pyramid bidirectional long short term memory (pBLSTM) (Chan et al., 2016; Hochreiter and Schmidhuber, 1997). It transforms the speech feature $x = (x_1, x_2, \cdots, x_t)$ into a high level representation $H = (h_1, h_2, \cdots, h_n)$, where $n \leq t$. In the pBLSTM, the outputs of two adjacent time steps of the current layer are concatenated and passed to the next layer.

$$h_j^i = \text{pBLSTM}(h_{j-1}^i, [h_{2j}^{i-1}, h_{2j+1}^{i-1}]). \quad (1)$$

Also, the pBLSTM can reduce the length of the encoder input from t to n. In our experiment, we stack 3 layers of the pBLSTM, so we were able to reduce the time step 8 times. The decoder is an attention-based LSTM, and it is a word-level decoder.

$$\begin{aligned} c_i &= \text{Attention}(s_i, h), \\ s_i &= \text{LSTM}(s_{i-1}, c_{i-1}, y_{i-1}), \\ y_i &= \text{Generate}(s_i, c_i), \end{aligned} \quad (2)$$

where the Attention function is a location-aware attention mechanism (Chorowski et al., 2015), and the Generate function is a feed-forward network to compute a score for each symbol in target vocabulary.

2.2 Discriminator

Discriminator takes either real text or ST translations as input and outputs a scalar QS as the quality score. For the discriminator, we use a traditional convolution neural network (CNN) (Kalchbrenner et al., 2016) which focuses on capturing local repeating features and has a better computational efficiency than recurrent neural network (RNN) (LeCun et al., 2015). The real text of the target language is encoded as a sequence of one-hot vectors $y = (y_1, y_2, \cdots, y_m)$, and the output generated by the ST model is denoted as a sequence of vectors $\widetilde{y} = (\widetilde{y}_1, \widetilde{y}_2, \cdots, \widetilde{y}_n)$. The sequence of vectors y or $\widetilde{y}$ are given as

input to a single layer neural network. The output of the neural network is fed into a stack of two one-dimensional CNN layers and an average pooling layer. Then we use a linear layer to get the quality score. Training the discriminator is easy to overfit because the probability distribution for ST model output is different from the one-hot encoding of the real text. To address this problem, we used earth-mover distance in WGAN (Martin Arjovsky and Bottou, 2017) to estimate the distance between the ST model output and real text. The loss function of the discriminator is the standard WGAN loss, and adds a gradient penalty(Gulrajani et al., 2017). Formally, the loss function of the discriminator as below:

$$\begin{aligned}\text{Loss}_D = \lambda_1\{\mathbb{E}_{\widetilde{y}\sim P_{st}}[D(\widetilde{y})] - \mathbb{E}_{y\sim P_{real}}[D(y)]\} \\ + \lambda_2\mathbb{E}_{\hat{y}\sim P_{\hat{y}}}[(\bigtriangledown_{\hat{y}}||D(\hat{y})|| - 1)^2], \quad (3)\end{aligned}$$

where λ_1 and λ_2 are hyper-parameter, P_{st} is the distribution of ST model $\widetilde{y}$ and P_{real} is the distribution of real text y, $D(y)$ is the quality score for y given by discriminator, $\hat{y}$ are samples generate by randomly interpolating between $\widetilde{y}$ and y.

2.3 Adversarial Training

Both the ST model and the discriminator are trained iteratively from scratch. For the ST model training step, the parameters of discriminator are fixed. We train the ST model by minimizing the sequence loss Loss_{ST} which is the cross-entropy between the ground truth and output of the ST model. And at the same time, the discriminator generates a quality score QS for the output of the ST model. Formally, the final loss function in the training process is as follows,

$$\text{Loss}_G = \lambda_{st}\text{Loss}_{ST} - (1-\lambda_{st})QS, \quad (4)$$

where $\lambda_{st} \in$ [0,1] is hyper-parameter. For the discriminator training step, the parameters of ST model are fixed. The discriminator uses the probability distribution of the ST model output and the real text for training. The specific learning process is shown in Algorithm 1. Note that the discriminator is only used in the training of the model while it is not used during the decoding. Once the training ends, the ST model implicitly utilizes the translation knowledge learned from discriminator to decode the input audio.

Algorithm 1 Adversarial Training

Require: G, the Generator; D, the Discriminator; dataset(X,Y), speech translation parallel corpus.
Ensure: $G^{'}$, generator after adversarial training.

for iteration of adversarial training **do**
  for iteration of training G **do**
    Sample a subset(X_{batch},Y_{batch}) from dataset(X,Y)
    $Y^{'}_{batch}$=$G(_{batch})$
    Use Eq.4 as loss function and compute the loss
    Update parameters of G with optimization algorithm
  end for
  for iteration of training Discriminator D **do**
    Sample a subset(X_{batch},Y_{batch}) from dataset(X,Y)
    $Y^{'}_{batch}$=$G(_{batch})$
    Let Y_{batch} as y , $Y^{'}_{batch}$ as $\widetilde{y}$, use Eq.3 as loss function and compute the loss
    Update parameters of D with optimization algorithm
  end for
end for

3 Experiment

3.1 Data Set

We conduct experiments on CCMT 2019-BSTC (Yang et al., 2019) which is collected from the Chinese mandarin talks and reports as shown in Table 1. It contains 50 hours of real speeches, including three parts, the audio files in Chinese, the transcripts, and the English translations. We keep the original data partitions of the data set and segmented the long conversations used for simultaneous interpretation into short utterances.

Dataset	Utterances	Hours
Train	28239	41.4
Valid	956	1.3
Test	569	1.5

Table 1: Size of the CCMT 2019-BSTC.

3.2 Experimental Settings

We process Speech files, to extract 40-dimensional Filter bank features with a step size of 10ms and window size of 25ms. To shorten the training time, we ignored the utterances in the corpus

that were longer than 30 seconds. We lowercase and tokenize all English text, and normalize the punctuation. a word-level vocabulary of size 17k is used for target language in English. Then the text data are represented by sequences of 1700-dimensional one-hot vectors. Our ST model uses 3 layers of pBLSTM with 256 units per direction as the encoder, and 512-dimensional location-aware attention was used in the attention layer. The decoder was a 2 layers LSTM with 512 units and 2 layers neural network with 512 units to predict words in the vocabulary. For the discriminator model, we use a linear layer with 128 units at the bottom of the model. Then, using 2 layers one-dimensional CNN, from bottom to top, the window size is 2, the stride is 1, and the window size is 3, the stride is 1. Adam (Kingma and Ba, 2014) was used as the optimization function to train our model, which has a learning rate of 0.0001 and a mini-batch size of 8. The hyper-parameters λ_{st}, λ_1 and λ_2 are 0.5, 0.0001 and 10 respectively. And the train frequency of the ST model is 5 times then the discriminator.

We used the BLEU (Papineni et al., 2002) metric to evaluate our ST models. We try five settings on Speech Translation. The Pipeline model cascades an ASR and an MT model. For the ASR model, we use an end-to-end speech recognition model similar to LAS and trained on CCMT 2019-BSTC. For the MT model, we use open source toolkit OpenNMT (Klein et al., 2017) to train an NMT model. The end-to-end model (described in section 2) does not make any use of source language transcripts. The pre-trained model is the same as the end-to-end model, but its encoder is initialized with a pre-trained ASR model. And the pre-trained ASR model is trained using Aishell (Bu et al., 2017), a 178 hours Chinese Mandarin speech corpus, which has the same language as our chosen speech translation corpus. The multitask model is a one-to-many method, where the ASR and ST tasks share an encoder. The Adversarial Training is the approach proposed in this paper.

3.3 Results

Table 2 shows the result of the different models on the validation set of CCMT 2019-BSTC. From this result, we can find that the end-to-end methods including pre-trained, multitask and Adversarial Training all get results comparable to the Pipeline model. Among them, the pre-trained model gets the best results. Our analysis is that this model uses a larger scale of speech corpus for pre-training, thus introducing more information into the model. We can see that the Adversarial Training method can obtain 19.1 BLEU, which is an improvement of 1.4 BLEU over the end-to-end baseline model, and even better than the multitask method. The multitasking approach uses transcription of source language speech, and our proposed approach is superior to it without using other information.

Model	ST
pipeline	19.4
end-to-end	17.7
pre-trained	20.4
multitask	18.9
Adversarial Training	19.1

Table 2: BLEU scores of the speech translation experiments

4 Conclusion

In this paper, we present the Adversarial Training approach to improve the end-to-end speech translation model. We applied GAN to the speech translation task and achieved good results in the experimental results. Since GAN's structure is used, this method can be applied to any end-to-end speech translation model. Unlike the multitask, pre-trained, and knowledge distillation previously proposed, this method requires the use of additional parallel corpus, which is very expensive to collect. In the future, we will experiment with unpaired text in order to be able to use this method to utilize an infinite amount of spoken text.

References

Antonios Anastasopoulos and David Chiang. 2018. Tied multitask learning for neural speech translation. *arXiv preprint arXiv:1802.06655*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Sameer Bansal, Herman Kamper, Karen Livescu, Adam Lopez, and Sharon Goldwater. 2018. Pre-training on high-resource speech recognition improves low-resource speech-to-text translation. *arXiv preprint arXiv:1809.01431*.

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

Alexandre Bérard, Olivier Pietquin, Christophe Servan, and Laurent Besacier. 2016. Listen and translate: A proof of concept for end-to-end speech-to-text translation. *arXiv preprint arXiv:1612.01744*.

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

W. Chan, N. Jaitly, Q. Le, and O. Vinyals. 2016. Listen, attend and spell: A neural network for large vocabulary conversational speech recognition. In *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4960–4964.

C. Chiu, T. N. Sainath, Y. Wu, R. Prabhavalkar, P. Nguyen, Z. Chen, A. Kannan, R. J. Weiss, K. Rao, E. Gonina, N. Jaitly, B. Li, J. Chorowski, and M. Bacchiani. 2018. State-of-the-art speech recognition with sequence-to-sequence models. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 4774–4778.

Jan K Chorowski, Dzmitry Bahdanau, Dmitriy Serdyuk, Kyunghyun Cho, and Yoshua Bengio. 2015. Attention-based models for speech recognition. In *Advances in neural information processing systems*, pages 577–585.

Long Duong, Antonios Anastasopoulos, David Chiang, Steven Bird, and Trevor Cohn. 2016. 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*, pages 949–959.

Ishaan Gulrajani, Faruk Ahmed, Martin Arjovsky, Vincent Dumoulin, and Aaron C Courville. 2017. Improved training of wasserstein gans. In *Advances in neural information processing systems*, pages 5767–5777.

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

Nal Kalchbrenner, Lasse Espeholt, Karen Simonyan, Aaron van den Oord, Alex Graves, and Koray Kavukcuoglu. 2016. Neural machine translation in linear time. *arXiv preprint arXiv:1610.10099*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Jean Senellart, and Alexander M Rush. 2017. Opennmt: Open-source toolkit for neural machine translation. *arXiv preprint arXiv:1701.02810*.

Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature*, 521(7553):436–444.

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

SC Martin Arjovsky and Leon Bottou. 2017. Wasserstein generative adversarial networks. In *Proceedings of the 34 th International Conference on Machine Learning, Sydney, Australia*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

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

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