Effectively pretraining a speech translation decoder with Machine Translation data

Ashkan Alinejad, Anoop Sarkar
Simon Fraser University, Burnaby, BC, Canada
{aalineja,anoop}@sfu.ca

Abstract

Directly translating from speech to text using an end-to-end approach is still challenging for many language pairs due to insufficient data. Although pretraining the encoder parameters using the Automatic Speech Recognition (ASR) task improves the results in low resource settings, attempting to use pretrained parameters from the Neural Machine Translation (NMT) task has been largely unsuccessful in previous works. In this paper, we will show that by using an adversarial regularizer, we can bring the encoder representations of the ASR and NMT tasks closer even though they are in different modalities, and how this helps us effectively use a pretrained NMT decoder for speech translation.

1 Introduction

Automatic Speech Translation (AST) aims to directly translate audio signals in the source language into the text words in the target language. For many years, the pipeline of transcribing speech with ASR and then translating with the MT component was a standard method to address the speech translation problem. Having access to lots of data in many language pairs, the cascaded model for speech translation can benefit from well-trained ASR and MT components and generate high-quality translations.

In recent years, it has shown that we can remove the transcription step and build an end-to-end model that is strong enough to compete with the cascaded model (Pino et al., 2019). Such models not only have lower inference latency, but they also do not suffer from the problem of errors that propagate from one component to the next. However, the scarcity of available resources is the main challenge in this task, and a variety of methods are proposed to address this problem. One of the most effective approaches to increase the performance of AST systems is to pretrain the encoder using an ASR model (Bansal et al., 2018). While pretraining the encoder by an ASR model even in different languages shows promising results (Bansal et al., 2019), using a pretrained MT decoder is not beneficial (Berard et al., 2018; Bansal et al., 2018) or slightly improve the result (Sperber et al., 2019) and even in some cases may worsen the results (Bahar et al., 2019).

One explanation for this phenomenon is that the decoder works well only if its input comes from an encoder that it was trained with (Lample et al., 2018). To solve the problem of invariant encoder representations, we make use of an adversarial regularizer in our loss function to bring the output of the ASR encoder closer to the input of MT decoder. We show that this modification can improve the BLEU score by +2.0 BLEU points.

2 Models

2.1 End-to-End Speech Translation

Similar to conventional MT models, the speech translation task generates translated words in the target language, representing as \( \hat{Y} = (\hat{y}_1, \ldots, \hat{y}_m) \), given the sequence of source speech features \( X = (x_1, \ldots, x_n) \). The translation model then minimizes the Cross-Entropy loss \( L_{CE} = \Delta(\hat{Y}, Y) \), where \( \Delta \) is the sum of character-level Cross-Entropy losses.

We use character-level encoding and decoding using Transformer (Vaswani et al., 2017) as the basic architecture of all our models. For the AST and ASR models, we use similar architecture to (Di Gangi et al., 2019b) with an S-Transformer (Gangi et al., 2019). The main difference between transformer and S-Transformer is the way it encodes the input features. S-Transformer encodes the audio features by passing them into two stacked layers of Convolutional Neural Nets (CNN). Then,
it uses a 2D self Attention layer to compute the attention matrix using the second CNN’s output. We followed the architecture of (Vaswani et al., 2017) in our MT model.

The conventional method for training an AST model is to pretrain ASR and NMT models separately and then transferring parameters of the encoder from ASR and the decoder from MT to the AST model, before starting to train via speech translation data.

2.2 Aligning encoder representations
Since we are training the encoder representations of the ASR model and the decoder parameters of the NMT system to work with their own encoder and decoder, pretraining the parameters of the AST model with a speech encoder from ASR and a text decoder from NMT is not ideal. Therefore, we propose to use adversarial training to bring NMT encoder and ASR encoder representations closer together.

An overview of our model is depicted in Figure 1. Instead of separately pretraining the ASR and NMT, we propose to update their parameters simultaneously. In order to add explicit incentives to learn multi-modal representations in the encoder, we will train our NMT and ASR models on both Cross-Entropy loss and a new regularization loss. The final training objective for each task can be formulated as:

\[
\text{Loss} = \text{L}_{CE} + \alpha \text{L}_{DISC}
\]

where \( \text{L}_{CE} \) is the Cross-Entropy loss, \( \text{L}_{DISC} \) is the newly added regularization term, and \( \alpha \) is the constant parameter to control the effect of our regularizer. Since \( \text{L}_{DISC} \) is a smaller number compared to \( \text{L}_{CE} \), we set \( \alpha \) to 5 in all our experiments to make the regularizer loss more perceptible during backward propagation. We are also sharing the parameters of the transformer layers in the encoder between AST and MT models. In the following section, we describe the regularizer.

2.3 Adversarial regularizer
Given the embeddings of inputs \( x_i \) in each modalities (speech features for ASR or character embeddings for NMT), the encoder computes the encoder representations \( Z_{x_i} \). By passing \( Z_{x_i} \) to the discriminator, we can train its network by minimizing the loss function \( \text{Loss}_D = -E_{(x_i,m_i)}[\log P_D(m_i|Z_{x_i})] \), where \( m_i \) is the modality of \( x_i \), with \( m_i \in \{\text{ASR, NMT}\} \) and \( P_D \) is the probability of choosing the right modality given the output of encoder.

The encoder of NMT or ASR will be trained in order to deceive the discriminator by minimizing the loss:

\[
\text{L}_{DISC} = -E_{(x_i,m_i)}[\log P_D(m_j|Z_{x_i})]
\]

where \( m_j = \text{ASR} \) if \( m_i = \text{NMT} \) and vice versa. By incorporating this regularizer, we ensure that the encoder representations from different modalities (speech and text) become indistinguishable during training.

Our discriminator consists of a three-layer feed-forward network with 1024 hidden units, followed by a Leaky-ReLU activation function (Lample et al., 2018).

3 Experiments
3.1 Dataset
To evaluate our AST systems, we conducted our experiments on two datasets. For the English-German language pair, we use the MuST-C corpus (Di Gangi et al., 2019a), which consists of 408 hours of speech data aligned with 234K translated sentences. For the English-French language pair, we use the full training set of Translation
Augmented Librispeech (Libri-Trans) corpus (Kocabiyikoglu et al., 2018) with 230 hours of speech aligned with 131K French sentences.

We use LibriSpeech corpus (Panayotov et al., 2015) with 960h of English speeches in order to train our ASR system. Since the test and dev sets of Libri-Trans corpus is part of the ASR LibriSpeech dataset, we remove all utterances from ASR LibriSpeech that share the same (chapter-id, reader-id) pairs with the test and dev sets in the Libri-Trans corpus. For En-De MT training, we use the combination of TED and Opensubtitle2018 corpora 1 2 which contains more than 18M sentence pairs after filtering noisy pairs. The MT training of the English-French language pair uses the En-Fr portion of the WMT14 competition (Bojar et al., 2014).

3.2 Preprocessing and Evaluation

For each speech utterance, we extract 40 Mel-filterbank energy features with a step size of 10 ms and a window size of 25ms. For features extracted from MuSt-C and ASR LibriSpeech, we apply mean and variance normalization for each speaker.

We keep all the texts in our experiments true-case and tokenize them using Moses tokenizer 3. We remove the punctuation from all English texts (both from the target side of ASR and the source side of MT).

For translation tasks (AST and MT), we report BLEU score (Papineni et al., 2002) on tokenized sentences 4. We evaluate our ASR systems using Word Error Rate (WER) 5.

3.3 Model settings

For both En-De and En-Fr tasks, we followed the architecture in (Di Gangi et al., 2019b). We use six Transformer layers of size 512 in the encoder and decoder with eight attention heads. The size of feed-forward mechanism is 1024. The embedding layer in the encoder for the AST task contains two layers of 2D CNNs (Lecun et al., 1998) followed by a ReLU activation function. Each CNN layer has 16 output channels, with a stride of (2, 2). We run all our models on two GeForce GTX 1080 GPUs with 12GB RAM each. The total number of parameters and run-time of our models in Table 1.

Table 1: The number of parameters and run-time of our models on MuST-C dataset (En-De) and Libri-Trans dataset (En-Fr).

<table>
<thead>
<tr>
<th></th>
<th>En-De</th>
<th>En-Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>cascadde NMT</td>
<td>45M 27</td>
<td>45M 13</td>
</tr>
<tr>
<td>cascadde ASR</td>
<td>31M 22.5</td>
<td>31M 18.2</td>
</tr>
<tr>
<td>AST</td>
<td>31M 34</td>
<td>31M 18</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Task</th>
<th>En-De</th>
<th>En-Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>cascadde</td>
<td>18.76</td>
<td>15</td>
</tr>
<tr>
<td>AST + ASR pre</td>
<td>18.71</td>
<td>14.7</td>
</tr>
<tr>
<td>AST + ASR pre + MT pre</td>
<td>19.05</td>
<td>15.3</td>
</tr>
<tr>
<td>AST + regularizer</td>
<td>20.24</td>
<td>17.01</td>
</tr>
</tbody>
</table>

Table 2: Results of AST models trained only with AST data. The performance is measured with BLEU score on MuST-C test set.

3.4 Training settings

In all our models, we use the Adam optimizer (Kingma and Ba, 2014) with an initial learning rate of 0.00005. During the first 6000 warm-up updates, we increase it linearly to 0.003, then decrease it with inverse square root decay (Vaswani et al., 2017). The number of warm-up updates in our MT systems is 8000.

4 Results

In this section, we analyze the effect of our regularizer on two different settings: (A) When we only have access to AST data (section 4.1) and (B) When we can benefit from External data (section 4.2). For each setting, we run experiments on four different models:

1. The cascaded model
2. AST model with pretrained ASR encoder
3. AST model with pretrained ASR encoder and MT decoder
4. Our proposed model with adversarial loss.

5. https://github.com/belambert/asr-evaluation
### Table 3: BLEU scores of AST models, trained with both AST and external ASR and MT data.

<table>
<thead>
<tr>
<th>Task</th>
<th>En-De</th>
<th>En-Fr</th>
</tr>
</thead>
<tbody>
<tr>
<td>cascaded</td>
<td>21.06</td>
<td>19.21</td>
</tr>
<tr>
<td>AST + ASR pre</td>
<td>19.01</td>
<td>16.13</td>
</tr>
<tr>
<td>AST + ASR pre + MT pre</td>
<td>19.12</td>
<td>16.27</td>
</tr>
<tr>
<td>AST + regularizer</td>
<td>20.81</td>
<td>17.7</td>
</tr>
</tbody>
</table>

#### 4.1 Using only AST data

Table 2 shows the performance of AST models for En-De and En-Fr language pairs. When the cascaded model is restricted to use small AST datasets merely, the model will not be strong enough to beat an AST model with a pretrained encoder and decoder. We should also note that unlike (Bansal et al., 2019; Bahar et al., 2019), where transferring decoder parameters were not effective, in all our AST models, we could only beat the cascaded model by pretraining the decoder.

The last row in the table gives the AST model results, which uses adversarial regularizer during the pretrain step. As we can see, training the NMT and the ASR models simultaneously can help pretrained components be compatible with each other and improve the final performance by 1.2 and 1.7 BLEU scores for En-De and En-Fr language pairs respectively.

#### 4.2 Using both AST and External data

Limiting the training data for the speech translation models to AST datasets is not a realistic assumption for many language pairs, and in practice, the cascaded model can greatly benefit from the large amounts of NMT and ASR corpora.

Table 3 summarizes the effects of adding external training data to our experiments. Adding external data can boost the performance of the cascaded model and by comparing Table 2 and 3, we can see that the additional NMT and ASR data can improve the translation quality of the cascaded model by +2 BLEU scores, while it can barely affect the AST model with pretrained encoder and the decoder. Consequently, the gap between the AST model and the cascaded system increases by around +3 BLEU scores for En-Fr and +2 BLEU scores for the En-De language pair.

As we can see in the last row of Table 3, adding our proposed pretraining step can help the model perform better during training, and compared to the conventional pretraining step, we can see an increase of more than 1 BLEU point in each language pair. Although the cascaded model by having access to all the pretrained parameters (the encoder and decoder of both NMT and ASR) still has better translation quality, we can bring the performance of an end-to-end model closer to it by adding the new regularizer. It is also important to note that since we are not changing the final structure of the AST model, most of the other techniques for further improving the translation quality, such as data augmentation, which was examined in previous studies (McCarthy et al., 2020; Park et al., 2019) can also be applied. But we won’t study them in this paper.

### 5 Related Work

The cascaded pipeline of transcribing speech signals and then translating them using an MT component (Ney, 1999; Cho et al., 2017) was for many years the standard design of speech translation systems (Inaguma et al., 2019). The idea of having an end-to-end structure for this task showed promising results in the works of (Adams et al., 2016; Duong et al., 2016; Bérard et al., 2016; Anastasopoulos et al., 2016; Anastasopoulos and Chiang, 2017; Bansal et al., 2017). After the success of (Weiss et al., 2017) in creating a powerful model for ST systems, more recent studies focused on exploring their power, and one of the main approaches to boost the performance of such models is to make use of available data from other tasks, such as ASR and NMT. (Weiss et al., 2017; Anastasopoulos and Chiang, 2018; Sperber et al., 2019) show that multitask learning can be effective and (Jia et al., 2019; Pino et al., 2019; Park et al., 2019; McCarthy et al., 2020) investigate various data augmentation techniques. The impact of pretraining the encoder with ASR model is also studied in (Berard et al., 2018; Bansal et al., 2018, 2019). In experiments of (Bahar et al., 2019; Bansal et al., 2019) the performance gain of pretraining the decoder with an MT model was marginal.

(Kano et al., 2020) addresses the ASR encoder and MT decoder gap problem by proposing a “Transcoder” and use smooth-L1 loss to bring ASR hidden representation close to MT encoder hidden representation.

The idea of modifying loss function in AST models was also discussed in (Sperber et al., 2019). Their formulation of the additional loss is different from ours, and they use their additional loss...
function in a different NMT architecture from ours.

The idea of adding adversarial regularizer was discussed in other tasks such as unsupervised MT (Lample et al., 2018) or zero-shot translation (Pham et al., 2019). The closest research to our work is (Arivazhagan et al., 2019), which uses a similar adversarial network to bring encoder representations closer together. However, they apply their model to the zero-shot machine translation task, with a different architecture. They also apply their regularizer to the representations of the different languages with the same modalities.

6 Conclusion

In this paper, we study the impact of pretraining an AST decoder using an MT model and propose a method to make the pretraining step more effective. We show that we can align the latent representations of different modalities by using adversarial loss and make the ASR encoder more compatible with the MT decoder. Our experiments demonstrate that we can improve the performance by around 1.5 BLEU points on two language pairs compared to conventional pretraining methods.

Acknowledgments

We would like to thank the anonymous reviewers for their helpful comments. The research was partially supported by the Natural Sciences and Engineering Research Council of Canada grants NSERC RGPIN-2018-06437 and RGPAS-2018-522574 and a Department of National Defence (DND) and NSERC grant DGDND-2018-00025.

References


pages 12–58, Baltimore, Maryland, USA. 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. In *INTERSPEECH*. 