DiDi Labs' End-to-End System for the IWSLT 2020 Offline Speech Translation Task

Arkady Arkhangorodsky, Yiqi Huang, Amittai Axelrod
DiDi Labs
4640 Admiralty Way
Marina del Rey, CA 90292
{arkadyarkhangorodsky, yiqihuang, amittai}@didiglobal.com

Abstract

We describe the DiDi Labs system submitted for the IWSLT 2020 Offline Speech Translation Task (Ansari et al., 2020). We trained an end-to-end system that translates audio from English TED talks to German text, without producing intermediate English text. Our base system used the S-Transformer architecture (Di Gangi et al., 2019b), trained using the MuST-C dataset (Di Gangi et al., 2019a). We extended the system via decoder pre-training, pre-trained speech features, and text translation, but these extensions did not yield improved results.

1 Introduction

The performance of end-to-end speech translation systems at IWSLT has been approaching that of cascaded systems, with the gap shrinking to 1.5 BLEU points in 2019 (Niehues et al., 2019). With additional effort, end-to-end systems could finally surpass cascaded systems. The 2020 task required participants to translate audio from English TED talks to German text.

We trained several different end-to-end speech translation systems. We used the MuST-C dataset to train models for speech translation and speech recognition, the Europarl-ST dataset for speech recognition (Iranzo-Sánchez et al., 2019), and the WMT-19 news commentary dataset for text translation (Tiedemann, 2012). Our best performing model used an encoder that was first pre-trained for English speech recognition, and then fine-tuned for speech translation. This system scored 17.1 BLEU on the MuST-C test set.

2 Experimental Framework

Our models used the S-Transformer architecture of Di Gangi et al.. This is an adaptation of the Transformer architecture (Vaswani et al., 2017) for speech inputs. The encoder performs a 2-D convolution on the audio input before applying self-attention as in the Transformer. Another distinction is that the decoder operates at the character level, instead on the byte-pair encoding (BPE) tokenization that is typically used with transformer models for text. The system uses a 512-dimensional embedding in the self-attention layers. Each of the encoder and decoder have 8-headed attention and 6 self-attention layers. The models have 32,132,040 parameters.

Each of our models were run on a single Nvidia Tesla P-100 GPU. We used a batch size of 8, and the Adam optimizer with a learning rate of 0.005 and an inverse square root warm-up schedule starting from 0.0003 for the first 4000 training steps. Each model was trained for up to 50 epochs, stopping early when validation loss had not decreased for 10 consecutive epochs.

We trained 6 models using different methods. We used the German transcripts and German audio from Europarl-ST for decoder pre-training. We used the WMT News Commentary parallel corpus for text translation. All other experiments used the MuST-C dataset. Table 1 contains the statistics for the corpora we used.

3 Extending S-Transformer

3.1 Naïve Model

Our simplest model was the S-Transformer, trained end-to-end on the MuST-C corpus using English audio inputs and German text outputs. This model was not able to successfully learn the task, achieving a score of 0 BLEU on the MuST-C test set. This is not surprising, as the relationship between the English audio and German text is not obvious without prior knowledge, even to most humans. This model effectively learned to memorize the most common output sentence from

Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT), pages 69–72
July 9-10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

Dataset	Segments	Input	Output
MuST-C training	229,703	EN Audio	EN, DE Text
MuST-C dev	1,423	EN Audio	EN, DE Text
MuST-C test	2,641	EN Audio	EN, DE Text
WMT news commentary	338,285	EN Text	DE Text
Europarl-ST training	12,904	DE Audio	DE Text
Europarl-ST dev	2,603	DE Audio	DE Text

Table 1: Details of the datasets we use in our experiments

the training set ("Vielen Dank"), and produced this as output every time.

3.2 Encoder Pre-Training

The task was too difficult for a naïve system to learn from scratch, so we tried training it in two stages. First, the system was trained to predict English text given the English audio inputs from the MuST-C dataset. This model successfully learned to transcribe English audio, achieving a BLEU score on the MuST-C validation set of 60.45. [1]

We then discarded the *decoder* from this English ASR system. The rest of the model was then fine-tuned to predict German text from English audio. We were thus able to train an end-to-end system in stages without having the intermediate inputs and outputs inherent to a cascaded system.

By first learning the simpler task of speech recognition, the system was able to make sense of the audio input before attempting to learn to translate it. This system was the strongest that we trained, achieving a BLEU score of 17.1 on the MuST-C test set.

3.3 Decoder Pre-Training

Pre-training the encoder using the simpler speech recognition task was successful, so we attempted to similarly pre-train just the *decoder*, except for German speech recognition instead.

We started by training a German ASR system using the same initial S-Transformer architecture as in Section 3.2. Here we trained the ASR system on German audio inputs and German text outputs from the Europarl-ST dataset. This system successfully learned to transcribe German audio, achieving a score on the Europarl-ST validation set of 36.9 BLEU.

The rest of the training was analogous to the pre-trained encoder system: the *encoder* of this model was discarded, then the model was trained on the speech translation task. However, this model performed similarly to the naïve system.

This suggests that just learning the input audio without a corresponding text in the same language remains a key challenge. This is perhaps not surprising, as audio input and a text transcript operate at different timescales: text inputs have atomic elements, but audio inputs are not only subdivisible via faster sampling, but also overlapping in time if the stride distance is short.

3.4 Combining Pre-Trained Encoder and Pre-Trained Decoder

Although we were not able to fine-tune the pre-trained decoder system of Section 3.3 to produce a strong speech translation model, we wondered if it could still could be a useful addition to a system with a pre-trained encoder. We fine-tuned an end-to-end model that started with the encoder trained for English ASR, and the decoder trained for German ASR. However, this model was only about as good as using only the pre-trained encoder. Perhaps this approach could produce stronger results if the encoded representations of the encoder and decoder were aligned to one another, as occurs when learning seq2seq models from scratch.

4 Using wav2vec Inputs

The MuST-C corpus represents the input audio using 40-dimensional Mel-Filterbank features. Schneider et al. (2019) presented `wav2vec`: unsupervised pre-training to learn speech representations, with improved speech recognition results. We attempted to apply this same approach to speech translation, replacing the Mel-Filterbank features with `wav2vec` features as input to the system.

We use the pre-trained model released in the `fairseq` library[2] to compute features for the

[1] We used the BLEU score instead of standard ASR metrics to simplify our implementation. This metric was mainly used to determine whether or not the model was useful as a starting point for fine-tuning; the value of the score was less significant.

[2] https://github.com/pytorch/fairseq/tree/master/examples/wav2vec

MuST-C dataset. wav2vec features are 512-dimensional vectors, but the Mel-Filterbank features are 40-element vectors. We applied principal component analysis (PCA) to reduce the wav2vec vectors to 40 dimensions to match the existing architecture. To reduce the computational load, we simply computed the PCA transformation on the first segment of the training set, and then applied the same transformation matrix to each subsequent sample.

We then attempted to pre-train the encoder for English ASR using the same S-Transformer architecture as before, in Section 3.2. However, this model does not successfully learn to transcribe English audio during pre-training. After fine-tuning, it cannot translate English audio and also gets a score of 0 BLEU.

We suspected our dimensionality reduction from 512 to 40 was too crude, losing too much information. To see if this was the case, we also attempted to use the full 512-dimensional wav2vec features as input, and increased the system layer widths accordingly. However, computational constraints limited us to only training on 20,000 segments of the MuST-C training set. However, this model also does not successfully learn to transcribe English audio during pre-training. After fine-tuning, it still cannot translate English audio and also gets a score of 0 BLEU.

5 Text translation multi-task training

Strong text translation systems are often trained on many millions of sentences, if they are available. Transcribing audio and translating is more expensive than finding parallel sentences, so the MuST-C corpus is considerably smaller than text translation corpora. We hypothesized that additional training on translation data would improve performance.

We pre-trained an English to German MT system that shared the decoder with our S-Transformer system in Section 3.2, in order to improve the decoder's translation ability. This model used a standard transformer encoder, not the S-Transformer. Unfortunately after training, this model was not able to successfully learn to translate text, though this same corpus has been successfully used in previous work (Barrault et al., 2019). We did not conduct further experiments trying to use the shared decoder in this model for speech translation.

Model	BLEU
1. Baseline S-Transformer model	0.00
2. #1 + encoder pre-trained on English ASR	**17.1**
3. #1 + decoder pre-trained on German ASR	0.00
4. #1 + #2 + #3	16.8
5. #2 + wav2vec preprocessing	0.00
6. #1 + text translation multi-task training	0.00

Table 2: BLEU scores of our experiments, evaluated on the MuST-C test set

6 Results and Conclusion

Table 2 contains our experimental results. The model using an encoder pre-trained for English speech recognition performed best. Combining this model with a decoder pre-trained for German speech recognition performed roughly similarly.

We have presented several different experiments in training end-to-end speech translation system based on the S-Transformer architecture. Unfortunately, none of the experiments we presented were able to improve performance on the MuST-C test set relative to the models of Di Gangi et al. (2019c). With more work, the ideas we attempted could produce stronger systems in the future.

References

Ebrahim Ansari, Amittai Axelrod, Nguyen Bach, Ondrej Bojar, Roldano Cattoni, Fahim Dalvi, Nadir Durrani, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Kevin Knight, Xutai Ma, Ajay Nagesh, Matteo Negri, Jan Niehues, Juan Pino, Elizabeth Salesky, Xing Shi, Sebastian Stüker, Marco Turchi, and Changhan Wang. 2020. Findings of the IWSLT 2020 Evaluation Campaign. *Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020).*

Loïc Barrault, Ondřej Bojar, Marta R. Costa-Jussà, Christian Federmann, Mark Fishel, Yvette Graham, Barry Haddow, Matthias Huck, Philipp Koehn, Shervin Malmasi, Christof Monz, Mathias Müller, Santanu Pal, Matt Post, and Marcos Zampieri. 2019. Findings of the 2019 Conference on Machine Translation (WMT19). *WMT Conference on Machine Translation.*

Mattia A. Di Gangi, Roldano Cattoni, Luisa Bentivogli, Matteo Negri, and Marco Turchi. 2019a. MuST-C: a Multilingual Speech Translation Corpus. *NAACL (North American Association for Computational Linguistics).*

Mattia A. Di Gangi, Matteo Negri, Roldano Cattoni, Roberto Dessi, and Marco Turchi. 2019b. Enhancing Transformer for End-to-End Speech-to-Text Translation. *MT Summit.*

Mattia A. Di Gangi, Matteo Negri, and Marco Turchi. 2019c. Adapting Transformer to End-to-End Spoken Language Translation. *INTERSPEECH*.

Javier Iranzo-Sánchez, Joan Albert Silvestre-Cerdà, Javier Jorge, Nahuel Roselló, Adrià Giménez, Albert Sanchis, Jorge Civera, and Alfons Juan. 2019. Europarl-ST: A Multilingual Corpus For Speech Translation Of Parliamentary Debates. *arXiv [cs.CL]*.

Jan Niehues, Roldano Cattoni, Sebastian Stüker, Matteo Negri, Marco Turchi, Elizabeth Salesky, R. Sanabria, Loïc Barrault, Lucia Specia, and Marcello Federico. 2019. The IWSLT 2019 Evaluation Campaign. *IWSLT (International Workshop on Spoken Language Translation)*.

Steffen Schneider, Alexei Baevski, Ronan Collobert, and Michael Auli. 2019. wav2vec: Unsupervised Pre-Training for Speech Recognition. *arXiv [cs.CL]*.

Jörg Tiedemann. 2012. Parallel data, tools and interfaces in OPUS. *LREC (International Conference on Language Resources and Evaluation)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention Is All You Need. *NeurIPS (Neural Information Processing Systems)*.