Samsung's System for the IWSLT 2019 End-to-End Speech Translation Task

Tomasz Potapczyk, Pawel Przybysz, Marcin Chochowski, Artur Szumaczuk

Samsung R&D Institute, Poland

{t.potapczyk,p.przybysz,m.chochowski,a.szumaczuk}@samsung.com

Abstract

This paper describes the submission to IWSLT 2019 End-to-End speech translation task by Samsung R&D Institute, Poland. We decided to focus on end-to-end English to German TED lectures translation and did not provide any submission for other speech tasks. We used a slightly altered Transformer [1] architecture with standard convolutional layer preparing the audio input to Transformer encoder. Additionally, we propose an audio segmentation algorithm maximizing BLEU score on tst2015 test set.

1. Introduction

This paper describes the submission to IWSLT 2019 End-to-End Speech Translation task by Samsung R&D Institute, Poland.

System architecture and data preparation techniques were designed before IWSLT 2019 data were released. We have been using LibriSpeech corpus [2] to develop these techniques. We evaluated our models on TED 2010 test set. After IWSLT 2019 training data was published we gathered successful techniques to train a systems for this competition. These techniques were used in all three models that will be presented here.

Document structure is as follows. Firstly we describe data preparation and augmentation. Then we provide system specification and training procedure used in our experiments. We describe data segmentation algorithm used to segment test sets TED 2015 and TED 2019. We show results of our experiments with monolingual data and simple recurrent unit (SRU) [3]. Finally we conclude our results.

2. Training Data

To train our system we used only IWSLT 2019 permissible audio corpora - iwslt-corpus, TEDLIUM2[4] and MUST-C corpus[5]. Data preparation process started with data filtering. Then we generated synthetic target sentences with text-to-text machine translation model. We augmented audio input with sox[1]. Finally we included monolingual text data with empty audio input.

[1]SoX - Sound eXchange sox.sourceforge.net v14.4.1

2.1. Data filtration

We trained English ASR system that was used to filter iwslt-corpus and TEDLIUM2 corpora. We removed cases where WER score exceeded 75% when comparing ASR output and English reference. We decided that MUST-C corpus does not need filtration. Additionally we filtered iwslt-corpus with regard to quality of translation using statistical dictionary-based methods. Size of the corpora before and after filtration is shown in Table 1.

Corpora	Orig. size	Filtered	Length
iwslt-corpus (ASR)	171121	158737	224h
+ trans. quality	158737	126817	188h
TEDLIUM2	92973	90715	197h
MUST-C	229703	229703	400h

Table 1: Size (number of audio utterances) of the training corpora before and after filtration. Iwslt-corpus (ASR) is corpus filtered by ASR only. The last column is total audio length after filtration.

2.2. Synthetic target data

TEDLIUM2 corpus did not provide any German translations, therefore we generated synthetic targets using two Transformer Big MT systems trained with different hyperparameters on WMT data - *Paracrawl*, *Europarl* and *OpenSubtitles*. Training data for these systems has been prepared with our in-house data preparation pipeline. We also used synthetic translations as an alternative translation in iwslt-corpus when augmenting it. To diversify target data as much as possible, for each example created in augmentation process, we generated 4 translations, 2 per each MT model. Such a technique was described in Jia et al.[6]. Number of training examples with synthetic data are shown in Table 2.

2.3. Data Augmentation

We augmented the data by processing the audio files with three sox's effects: *tempo*, *speed* and *echo*. We sampled the parameters with uniform distribution within ranges presented in Table 3.

For each file we repeated the process four times. As a result we had five times larger audio corpus with synthetic

Corpora	Ref.	MT-1	MT-2
iwslt-corpus	126817	2x158737	2x158737
TEDLIUM2	0	3x90715	3x90715
MUST-C	229703	0	0

Table 2: Size (number of text lines) of the training corpora with synthetic data. For each model two or three best beam results have been used.

Option	Min value	Max value
tempo	0.85	1.3
speed	0.95	1.05
echo delay	20	200
echo decay	5	20

Table 3: Sox parameters value ranges used in processing of audio data. Echo effect is parametrised by two values.

translations for roughly half of the audio files. The range of *speed* option is very small because we did not want our model to train on an unnaturally sounding samples. The rationale behind using *echo* option is the fact that many TED lectures have a significant echo.

Final number of training audio examples is shown in Table 4.

Corpora	Orig. & Augm.	Length
iwslt-corpus	761765	1084h
TEDLIUM2	544290	1182h
MUST-C	918812	1600h
Total	2224867	3866h

Table 4: Size of the training audio corpora with data augmentation. Number of distinct audio and text pairs.

2.4. Monolingual data

Similarly to pipeline systems, quality of End-to-end speech translation system depends on accuracy of extracted audio features from the source input as well as on quality of target generation. E2E system can be improved by either introducing more variation of input speech or more variation of targets. In Transformer decoder architecture the target self-attention layer works as a language model - depending on the previous decoded symbols it predicts the next one without looking at encoder output. This led us to believe we could add monolingual target data to the training corpus. To such monolingual data we attached an empty audio input to train just the self-attention part of the decoder. To choose this monolingual data we randomly selected 15 million sentences from *Paracrawl* and *Europarl* and trained a language model on these sentences. Next we trained a language model on TED corpora [7] and used cross-entropy difference scoring [8] to choose 2 million sentences closest to TED talks. BLEU scores of this model can be found in Table 7. Our test on a non-agumented data showed significant improvement of translation quality (15.23 vs 16.74 BLEU), however in the end, it was not the case when data was agumented. To our best knowledge such an approach has never been described in literature before.

3. E2E Speech Translation System

In this section we will describe architecture of end-to-end spoken language translation system.

3.1. ASR Transformer for SLT

As a baseline system we used Transformer architecture and hparams transformer_librispeech_v2 for automatic speech recognition implemented in TensorFlow. The targets, however were translation instead of transcripts. The Transformer has hidden layer of size 384, convolutional (kernel size 9) feed forward layer of size 1536, 2-head self-attention, 6 encoder layers and 4 decoder layers. Audio data is turned into log mel spectrogram with frame size of 25 ms, frame step 10 ms and 80 filters. To log mel spectrograms we apply 2D 3x3 convolution twice with stride 2x2 and 128 filters and then 3x80 convolution to reduce the spectrogram to a vector, exactly like in the case of ASR.

3.2. Dense Feed Forward Layer in Decoder

We also proposed a change to the baseline ASR architecture: use dense feed forward layer of the same size in the decoder layer instead of convolution. The rationale behind it being the fact that standard text Transformer uses such feed forward layer for generating translation.

For output of the decoder we use a standard representation used in text to text translation - subword data tokenizations with dictionary of size 32k.

3.3. Dual learning: ASR and SLT tasks

Additionally we introduced a second decoder with ASR task, making it a multitask setup similar to[9]. A separate dictionary of size 32k was used for this task. In such a setup loss is calculated with two targets - one in English and one in German. Two decoders with different weights are simultaneously trained on these targets. An experiment on non-augmented data showed almost 2 BLEU increase (15.23 vs 17.15 on tst2010) compared to the same model trained on a single task.

3.4. SRU Recurrent layers in Encoder

The sequences processed by the Transformer encoder are at least 4 times longer in the case of speech translation than in the case of text translation. We tried to contract these sequences further with convolutions to be able to use deeper and still fast encoder. Unfortunately, this resulted in a signif-

icant reduction of BLEU. The best BLEU score was achieved after introducing 4 layers of Simple recurrent unit and then kernel 3, stride 2 convolution applied twice before the encoder. Number of encoder layers were increased to 8 and and embedding size to 512 without loosing speed of the decoding.

3.5. Spectrogram masking

To augment data even more we implemented spectrogram masking technique described in Park et al. [10] This technique involves masking the spectrogram for a range of frequencies and period of time. In our implementation we chose to introduce three such masks for frequency. The width of frequency range is selected randomly between values 5 and 10. This means that out of 80 filters 15 to 30 are masked. In time we chose one mask for every 300 time steps. Again, length of such mask is random between 10 and 20 time steps.

3.6. Training process: Adam Multistep optimizer

We trained our primary model on 4 GTX 1080 Ti GPUs for about a week, which resulted in 800k steps. *SRU* model was trained slightly longer - for 1 million steps. The model trained on additional monolingual data was trained for 3 million steps. Instead of using standard Adam optimizer, we used Adam Multistep optimizer updating weights every 32 batches. As a result effective batch size is increased. Without Multistep optimizer, models did not learn at all, possibly because batch size in this case is just a few utterances per card. Our early experiments on LibriSpeech data showed the best performance for multistep value of 32, for higher values the model trained much more slowly. In the case of all trainings 10% dropout was applied.

3.7. Model averaging

For a final validation we averaged last 7 checkpoints of the training. Averaging checkpoints almost always resulted in higher BLEU scores. We experimented with continuation of training after averaging but it did not give any better results.

4. Segmentation

This year's IWSLT formula allows the submissions to be based on a custom audio segmentation, which directs a part of the research effort towards finding the optimal method of splitting the input of a end-to-end model. The considered segmentation methods can be described as time- and feature-based. The time-based algorithms split the audio file in consecutive windows of constant or varying size and are completely ignoring the content of the audio, while the feature-based solutions extract and analyze specially designed traits of the input to enhance the division process.

To acquire a simple yet effective segmentation algorithm, the method of choice was based on a silence periods between utterances. The reasoning behind such a selection is quite intuitive. In general, the speakers tend to make longer pauses between separate sentences than between the words in a single sentence. To incorporate this observation, the designed method, further called DIV, utilizes the divide-and-conquer approach. Firstly we segment the audio with Audacity tool[2]

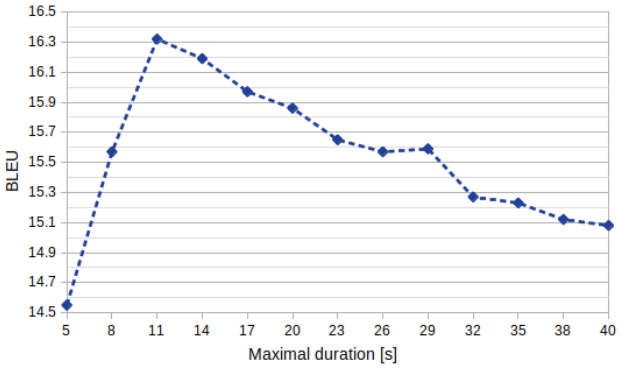

Figure 1: Dependency of BLEU on maximal segment lenght duration

Parameter	tst2010
Max silence	26 -db
Silence min duration	0.2s
Label starting point	0.2s
Label ending point	0.3s

Table 5: Audacity parameters for silence and speech recognition

to detect speech and salience periods. The method recursively splits the audio file and later its parts into 2 recordings at the point where the utterances are separated by the longest silence period. The algorithm finishes when no further splits are possible, that is when the lengths of all created parts are not longer than the user-specified threshold or contain a single utterance. To find the optimal value of the threshold parameter, multiple values were tested using the IWSLT 2015 dataset. The duration of 11 seconds was found to be the most beneficial yielding the BLEU score of 16.32. An analysis of different segmentation algorithms and the parameter tuning of a selected method allowed to observe the significant influence of a segmentation method on the produced translation. For comparison, the use of pre-processed segmentation resulted in a BLEU score of 12.38 while the sub-optimally parameterized DIV algorithm scored 15.12 BLEU, see Figure 1.

5. Evaluation

Table 6 presents experiment performed on the LibriSpeech training data set evaluated on TED 2010. In the case of these trainings data augmentation was applied only once.

[2]Audacity audacityteam.org v2.3.2

Model	tst2010
ASR Transformer for SLT	
baseline	12.76
+ Model averaging	12.99
+ Dense FF in Decoder	13.20
+ Spectogram masking	13.74
+ Data augmentation (speed x1)	14.85
+ 8 head attention	15.31
+ Data augmentation (speed + echo x1)	15.56

Table 6: BLEU scores for models trained on LibriSpeech data. Each subsequent model includes all the previous techniques in the table.

Table 7 shows comparison of the results for models trained on full permissible audio corpora. Primary system is a ASR Transformer with dense feed-forward layer in decoder, spectrogram masking, trained with a dual ASR task and averaged snapshots. *SRU* system is the same as above but recurrent layers in encoder were added. Finally *Mono* system is the same as Primary but trained with additional monolingual data.

Model	tst2010	tst2015	tst2019
Primary	25.81	21.29	19.96
SRU	25.70	19.08	18.83
Mono	23.99	20.81	19.36
ASR+MT	23.58	19.96	-

Table 7: BLEU scores for our three models. *SRU* model is an alternative architecture with simple recurrent unit and *Mono* model was trained with additional monolingual data. Additionally we compare the results to our general purpose ASR+MT pipeline system trained on unconstrained data.

6. Conclusions

We presented three end-to-end speech translation models. Our primary model achieved quality comparable to pipeline production systems Table 7. Unfortunately, introducing monolingual data to the training did not result in higher BLEU score on TED test sets. However this model might be better on other domains. We also showed an alternative architecture with slightly lower quality than primary system. This alternative architecture could be further improved in order to achieve higher decoding speed. Based on these results we conclude E2E models will challenge pipeline systems in the near future.

7. References

[1] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, 2017, pp. 5998–6008.

[2] V. Panayotov, G. Chen, D. Povey, and S. Khudanpur, "Librispeech: An asr corpus based on public domain audio books," *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5206–5210, 2015.

[3] T. Lei, Y. Zhang, S. I. Wang, H. Dai, and Y. Artzi, "Simple recurrent units for highly parallelizable recurrence," in *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, 2018, pp. 4470–4481.

[4] A. Rousseau, P. Deléglise, and Y. Esteve, "Enhancing the ted-lium corpus with selected data for language modeling and more ted talks." in *LREC*, 2014, pp. 3935–3939.

[5] M. A. Di Gangi, R. Cattoni, L. Bentivogli, M. Negri, and M. Turchi, "Must-c: a multilingual speech translation corpus," in *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 2019, pp. 2012–2017.

[6] Y. Jia, M. Johnson, W. Macherey, R. J. Weiss, Y. Cao, C.-C. Chiu, N. Ari, S. Laurenzo, and Y. Wu, "Leveraging weakly supervised data to improve end-to-end speech-to-text translation," in *ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*. IEEE, 2019, pp. 7180–7184.

[7] M. Cettolo, C. Girardi, and M. Federico, "Wit3: Web inventory of transcribed and translated talks," in *Conference of European Association for Machine Translation*, 2012, pp. 261–268.

[8] R. C. Moore and W. D. Lewis, "Intelligent selection of language model training data," in *ACL*, 2010.

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

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