End-to-end Speech Translation System Description of LIT for IWSLT 2019

Mei Tu, Wei Liu, Lijie Wang, Xiao Chen, Xue Wen

Speech Lab & Language Understanding Lab of Language Intelligence Team, Beijing

{mei.tu,wei.liu,lijie.wang,xiao.chen,xue.wen}@samsung.com

Abstract

This paper describes our end-to-end speech translation system for the speech translation task of lectures and TED talks from English to German for IWSLT Evaluation 2019. We propose layer-tied self-attention for end-to-end speech translation. Our method takes advantage of sharing weights of speech encoder and text decoder. The representation of source speech and the representation of target text are coordinated layer by layer, so that the speech and text can learn a better alignment during the training procedure. We also adopt data augmentation to enhance the parallel speech-text corpus. The En-De experimental results show that our best model achieves 17.68 on tst2015. Our ASR achieves WER of 6.6% on TED-LIUM test set. The En-Pt model can achieve about 11.83 on the MuST-C dev set.

1. Introduction

End-to-end Speech Translation is a promising task that attracts a lot of attention in recent years [1-5,11]. Compared to a traditional cascaded system [9,10] that performs ASR and MT separately, the end-to-end speech translation model complies with the encoder-decoder structure, which has advantages including 1) avoiding compounding errors between ASR and MT; 2) performing lower latency without the separate ASR component.

However, the translation quality speech translation is still hard to achieve the SOTA text translation performance. The main reason lies that the parallel speech-text translation corpus is hard to obtain, so that the learning process becomes a low-resource learning task. Previous studies [1-4] tend to address the training data sparseness by introducing multi-task learning. With multi-task learning framework, the translation model can get benefits from ASR corpus and MT corpus separately. The study [5] directly generates the speech data given text data by TTS system, but the generated data may face the low diversity in speaker voices in the synthetic data.

The state-of-the-art speech translation models mostly adopt the architecture in [1]. The encoder consists of CNN layers for down-sampling and stacked RNN cells to deeply represent the speech features, while the decoder performs attentive RNN under autoregressive mechanism. Roughly speaking, the encoder is for projecting the down-sampled source speech feature into a deep semantic space. The decoder is for generating text tokens based on the speech deep representation. When the encoder and decoder networks become deep, the information from source low-level layers is hard to pass to the high level in the decoder, even with an attention mechanism, which causes alignment information reduction. The CNN layers aggravate the information reduction after down-sampling.

The information reduction problem was discussed in the previous works [7,8] in text translation. They coordinate the encoder layers and decoder layers, so that the decoder neuron can also refer to the low-level neuron of source sentences to generate target sentences. Inspired by the mentioned works, in this paper, we introduce a layer-wise tied structure for end-to-end speech translation. We build connections between speech representation neurons and text representation neurons, which shorten the path between target text and source speech. We replace the RNN cell in encoder and decoder with self-attention to speed up the training and inference.

To make full use of the allowed data resource, we also use text translation data augmentation methods.

The rest of the paper is organized as follows. We first describe the processing for speech and text training data in Section 2, following is our full system and the training details. The experiments and results are presented in Section 4.

2. Data Processing

The paper focuses on IWSLT end-to-end speech translation task from English to German/Portugal. All experiments were performed under requirements of IWSLT 2019 evaluation campaign speech translation task. All the training data are listed in Table 1, Table 2 and Table 3.

Table 1. Speech training data

Corpus	# of seg.	Speech hours
TED-LIUM2	92973	212h
TED-LIUM3	268263	452h
IWSLT-no-label	948	180h
How2	184949	297h

Table 2. text training data

Corpus	Raw
Europarl	1.7M
ParaCrawl corpus	36.35M
Common Crawl corpus	2.39M
News Commentary v13	0.28M
Rapid corpus of EU press releases	1.32M
Open Subtitles2018	22.51M

Table 3. Parallel speech translation corpus

Corpus	Speech Hour	Text sentences
MuST-C	400h	0.22M
IWSLT-label	271h	0.17M

2.1. Speech data preprocessing

For TED-LIUM2, we follow the approach implemented in the Kaldi toolkit [25] to do the cleaning and re-segmenting[1]. After this process, the data size has been reduced to 145 hours from 212 hours. Based on these data, we trained a TDNN neural acoustic model as an initial model which used 8 layers TDNN architecture. We use 40-dimension mel frequency cepstral coefficient (MFCC) and 200-dimension i-vector as the input feature.

For TED-LIUM3, we follow the way the same as TED LIUM2, then we got clean data of 385 hours.

For IWSLT Speech Translation data, 270 hours of data are labeled, but some transcripts do not match well to their corresponding audio, we do forced alignment with the initial model and reduce the size to 226 hours. And about 180 hours of unlabeled data are recognized by our initial model, then we splice the fragments to at most 20 seconds. It will be recognized by ASR system which described in Section 3.2.1 to get the transcripts

There are only fbank and pitch features for How2 data, so we don't do cleaning or segmenting.

To increase the amount of training set, we apply speed perturbation (except for How2 corpus) with speed factors 0.9 and 1.1.

After data filtration and speed perturbation, the total speech data includes, TED LIUM2 of 435 hours, TED LIUM3 of 1154 hours, labeled IWSLT of 680 hours, no-labeled IWSLT of 540 hours, How 2 of 297 hours, MuST-C En-De of 1200 hours, MuST-C En-Pt of 375 hours.

2.2. Text data preprocessing

We only use text training data that contains parallel data for data augmentation in the end-to-end Speech Translation Task. The text data includes TED data, data provided by WMT2018 and OpenSubtitles2018. The data is preprocessed before training. Sentences longer than 100 words are removed. For one sentence pair, if the length rate of source/target is less than 1/2 or larger than 2, it will be removed. We then use n-gram language model to cross filter the corpus following [28]. For an English sentence, if its perplexity computed by English language model is larger than the perplexity computed by German language model, it will be removed. For a German sentence, if its perplexity computed by German language model is larger than the perplexity computed by English language model, it will be removed.

After filtering about 37M parallel En-De sentences in total are used for training.

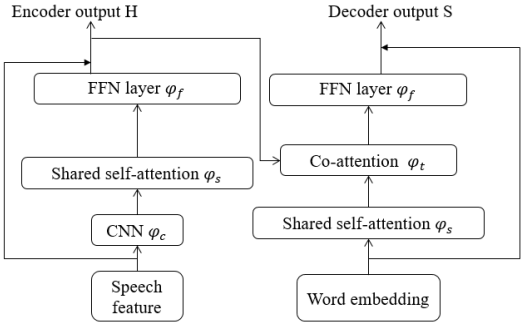

Fig.1. The proposed architecture

3. System Description

3.1. Speech Translation with Tied Layer Structure

Our speech translation model structure is depicted in Fig.1. The encoder consists of CNN layers and self-attention layers. It takes speech filter-bank feature, delta and delta-delta [21] as 3-channel input. The input passes a 2-layer CNN with kernel size of hyper-parameter *k* and max-pooling with stride of 2. The output of the CNN passes through a stacked self-attention based network. The decoder consists of self-attention network and co-attention network. The self-attention network in the encoder and decoder is layer-wise tied.

The proposed method can be described mathematically as follows. Given an input speech filter-bank feature $\mathrm{x} \in R^{t \times f}$, where *t* is the feature length and *f* is the frequency channel. we stack x, x-delta, x-delta-delta as the input of CNN with shape [*t*, *f*, *depth*], where the *depth* is 3. The encoder consists of 2-layer CNN module φ_{c}, L blocks of self-attention module and feed-forward layer, represented as φ_s^l and φ_f^l for *l*'th block. The decoder consists of L blocks of self-attention module, target-source co-attention module and feed-forward layer, represented as φ_s^l, φ_t^l and φ_f^l for the *l*'th block.

Then the encoding process is formulated as:

$$\mathrm{h}_i^0 = \varphi_{\mathrm{c}}(\mathrm{x}, \mathrm{x}', \mathrm{x}''; \mathrm{k}), \tag{1}$$

$$\mathrm{h}_i^l = \varphi_{\mathrm{f}}^{\mathrm{l}}\left(\varphi_{\mathrm{s}}^{\mathrm{l}}\left(\mathrm{H}^{\mathrm{l}-1}\right)\right), 0 < l < L \tag{2}$$

where *k* is the kernel-size hyper-parameter of φ_{c}. We get the CNN output with φ_{c}. $\mathrm{H}^{\mathrm{l}} = \{h_1^l, h_2^l, \ldots, h_t^l\}$. After L blocks of self-attention and feed-forward layer, we get H^{L} in the last layer.

The decoding process at $\mathrm{j}+1$'th step is the same with that in [8]. We directly refer them here.

$$\mathrm{s}_{\mathrm{j}+1}^l = \varphi_{\mathrm{f}}^{\mathrm{l}}(\varphi_t^l(\varphi_{\mathrm{s}}^{\mathrm{l}}(\mathrm{S}_{\mathrm{j}+1}^{l-1}), H^L)), \tag{3}$$

$$\mathrm{S}_{\mathrm{j}+1}^l = S_j^{\mathrm{l}} \cup \{s_{j+1}^l\}. \tag{4}$$

where *S* represents the hidden states at the target side. For any l that $< 0 \leq \mathrm{l} < \mathrm{L}$, parameters of $\varphi_{\mathrm{s}}^{\mathrm{l}}$ are shared. The hyper parameter *k* in (1) is selected based on the average speech rate. For a language whose average rate is *v* phoneme/sec, supposing the frame rate is *f* frames/sec, the frames per phoneme can be *f*/*v*. We suggest $k = f/v$. In this report, we use *k*=9.

[1] https://github.com/kaldi-asr/kaldi/blob/master/egs/tedlium/s5_r2/local/run_cleanup_segmentation.sh

3.2. Data Augmentation-MT synthesis

We know that the performance of end-to-end system largely depends on the data quality and quantity. We can improve the quality via the preprocessing described in Section 2. For quantity, the original parallel speech translation data is less than 0.5M after filtering duplicated data. It is far from enough for training a speech translation system. So we train an ASR and MT system firstly to do data augmentation.

3.2.1. ASR system

The ASR system in data augmentation is for English speech transcription. With fast development of end-to-end ASR techniques, the end-to-end models can achieve comparable or even better performance compared to the traditional ASR.

In our ASR system we consider the following structures,

- CTC, in which a blank token is leveraged for handling differences in the length of input acoustic features and output tokens [12–16].

- Attention based Seq2Seq, which are language models conditioned on input speech. In this method, an attention mechanism is utilized for automatically determining which acoustic features should be used to predict the next token [17–22].

- Recurrent neural network (RNN), transducers and recurrent neural aligners have been developed for use in online decoding [23, 24].

For implementation, we use espnet[26] tool for end-to-end ASR training. The input features is passed in to a 2-blocks of VGG-like layer. Each layer comprises 64 kernels with shape 3×1 and a stride of 2×1, followed by layer normalization and ReLU activation.

The followed encoder network is represented by 5-layer bidirectional long short-term memory (BLSTM) with 1024 hidden units per layer. We use a location-aware attention mechanism. The attention vectors are fed into a 2-layer LSTM decoder. When training and decoding, it adopts hybrid CTC/attention.

Espnet combines the log probability of RNN LM during decoding by using the shallow-fusion technique. We use all the sentences in Table 1 to train a BPE encoder with vocab size is 5000. Add then we got a vocabulary which size is 5053 to train an RNN LM.

As an evidence verification, we also conduct experiments with traditional TDNN system which consists of 6 layers TDNN architecture. Each hidden layer contains 850 units. 40 dimensional static MFCC and 100 dimension i-vector are extracted for training.

3.2.2. Text MT system

We use Transformer[29] to train the text machine translation system following the hyper-parameter settings based on Tensor2Tensor transformer relative big settings. The transformer is a 6-layer model with model size of 1024, a feed forward network size of 8192, and 16 heads relative attention. The model is trained on the full dataset described in Table 2 and filtered following Section 2.2.

To match the output of ASR, we remove all the punctuation on the source side except “”. For both sides, we apply tokenization and BPE[30]. The vocabulary size is about 20K for both sides.

3.2.3. Pipeline Based Data Augmentation

We apply the above ASR and text MT system on all the dataset in Table 1 to obtain the English transcripts and their corresponding synthetic German translation. After translation, we also filter the data as described in Section 2.2, as the generated target sentences contain noise.

4. Experiments

In this section, we report our experiments for the IWSLT 2018 speech translation evaluation TED task under end-to-end condition. We mainly test our systems on tst2015 for translation experiments and TED-LIUM test set for ASR experiments. Case sensitive BLEU is used for our translation evaluation metric. WER is used for the ASR evaluation metric.

For speech features, we use 40-dimensional filter banks, and cepstral mean variance normalization (CMVN) is performed at the speaker level to mitigate recording variations.

For the layer-tied speech translation model, the hyper-parameter kernel-size k in CNN is set as 9 in our experiments. The model size of shared self-attention is 512, the feed forward network size is 1024. We use 10-layer blocks for encoder and decoder. Considering the memory capacity, we do not use the same big setting like text MT model described in Section 3.2.2.

4.1. Results of ASR

We begin by investigating the impact of CTC’s weight and LM’s weight which use for computing the cross entropy and decoding respectively. The model described in Section 3.2.1 is training with only the TED-LIUM2 data.

Table 4. WER with different CTC’s weight (LM’s weight is 0.1)

CTC’s weight	WER(%)
0.0	17.4
0.2	13.1
0.5	**13.0**
0.8	13.6

Table 5. WER with different LM’s weight (CTC’s weight is 0.5)

LM’s weight	WER(%)
0.1	13.0
0.5	13.0
0.7	**12.5**
0.8	12.8

We see that when we set CTC’s weight as 0.5 meanwhile LM’s weight is 0.7, it gets the best performance. And then we train the E2E ASR model in such weights with different training data. And we apply SpecAugment[27] for data augmentation at Exp. A2~A4.

Table 6. WER for speech recognition with different training data on TED-LIUM test set

Exp.	Data set	WER(%)
A0	TED-LIUM2	12.5
A1	A0 + IWSLT	9.5
A2	A1 + SpecAugment	8.8
A3	A2 + How2 + MuST-C	8.2

A4	A3 + TED-LIUM3	**6.6**

We compare the E2E and cascade ASR in the following table. For TDNN, as the feature of How2 data is different from other data, we exclude How2 data here. From the table, we can see that the E2E system outperforms the cascade along with increasing training data, which proves that the end-to-end system depends largely on data amount.

Table 7. WER (%) with different networks

	TDNN	E2E
TED-LIUM2	11.3	12.5
All training data	10.2	6.6

4.2. Results of text MT

We remove the punctuation on source side of testing sets with the same rules as for the training set. Table 8 shows the BLEU scores tested on tst13~15 for the text MT system described in Section 3.2.2.

Table 8. BLEU scores of text MT system

	Tst10	Tst11	Tst12	Tst13	Tst14	Tst15
En-De	33.17	30.91	31.84	31.81	28.97	29.99

As our system doesn't focus on the cascade speech translation task, we only use ASR and MT for data augmentation. So we don't conduct any special optimization for cascade speech translation. The BLEU score of cascade ASR and MT on tst15 is about 18.50. It can be seen as a test-bed for cascade result but not used as the submitted model for the evaluation task.

4.3. Results of End-to-End Speech Translation

In this section, we describe our experimental results of end-to-end speech translation. As a matter of fact, the training time of end-to-end speech translation is much more than text translation, even the amount of the training corpus for text translation is much more than speech translation corpus. It is hard to achieve complete convergence sometimes. So in our work we just keep training as long time as we can for all the models.

Shown in Table 9, we can find the performance of end-to-end speech translation with only must-c corpus is lower than text translation by about 20 points. The data augmentation can bring about 8.13 points improvement.

Table 9. Data augmentation effect on En-De direction

	Tst15 BLEU
Must-c	9.55
+ IWLST-labeled+ data-augmentation	17.68

As the memory capacity limitation, running a deeper model like 10-layer transformer causes Out-of-Memory issue in our server. So we share the self-attention layers in our model as described in Section 3.1. But according to our experience, the over-parameterized network can always generate better translation, while the shared attention may cause the quality reduction due to the reduced model capacity. So we compare the shared attention and the original transformer in Table 9 and Table 10.

Table 11 shows that the shared attention can reduce the parameters, which reduce memory accordingly. So with the same memory limitation, the shared attention can achieve deeper layers. Notice that we compare the model capacity with the same other hyper-parameter settings, like unit number, FFN size.

Table 11. Shared attention vs. non-shared attention

	Parameters	Model capacity
non-shared structure	67M	6 layers
Shared attention	49M	10 layers

Table 12 compares the BLEU scores between shared attention and non-shared attention. As the memory consumption reduces because of sharing weights, we can use bigger setting. With bigger model setting, the BLEU scores improve on both tst13 and tst14. It is possible that if the server memory is enough, the over-parameterized big model may get further improvement.

Table 12 BLEU score of shared/non-shared attention on En-De direction

	layer	Model size	FFN	Tst13	Tst14
non-shared	6	512	1024	15.51	14.12
Shared attention	6	1024	4096	15.55	14.93

For En-Pt direction, we adopt the tied-layer Transformer upon on MuST-C En-Pt corpus. But we don't apply data augmentation for this direction, because we don't have enough En-Pt parallel corpus to train the text translation module. So the whole data we use is only MuST-C data and get the BLEU score of 11.83 on MuST-C dev data.

5. Conclusions

In this paper, we present our end-to-end speech translation system for IWSLT 2019 evaluation. Our results show that the system with data augmentation performs significantly better than the raw data baseline. The shared attention with bigger model size and feed forward network perform better than non-shared attention, which is especially suitable for the limited memory. We also find that there still exists large room to improve the speech translation quality in order to achieve the performance of text machine translation.

6. Acknowledgements

We would like to thank the organizing committees of IWSTL 2019. We thank Mengxia Zhai for her support during competition. We thank Yue Song for her work during her internship.

7. References

[1] R. J. Weiss, J. Chorowski, N. Jaitly, Y. Wu, and Z. Chen, "Sequence-to-sequence models can directly translate foreign speech," in *Proc. Interspeech*, 2017

[2] A. Anastasopoulos and D. Chiang, “Tied multitask learning for neural speech translation,” in *Proc. NAACL-HLT*, 2018
[3] S. Bansal, H. Kamper, K. Livescu, A. Lopez, and S. Goldwater, “Pre-training on high-resource speech recognition improves low-resource speech-to-text translation,” *arXiv preprint arXiv:1809.01431*, 2018
[4] A. Berard, L. Besacier, A. C. Kocabiyikoglu, and O. Pietquin, ´ “End-to-end automatic speech translation of audiobooks,” in *Proc. ICASSP*, 2018
[5] Jia, Ye, et al. "Leveraging weakly supervised data to improve end-to-end speech-to-text translation." ICASSP 2019-2019 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP). IEEE, 2019.
[6] W. Chan, N. Jaitly, Q. Le, and O. Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” in *Proc. ICASSP*, 2016.
[7] He, Tianyu, et al. "Layer-wise coordination between encoder and decoder for neural machine translation." Advances in Neural Information Processing Systems. 2018.
[8] Xia, Y., He, T., Tan, X., Tian, F., He, D., & Qin, T. (2019). Tied Transformers: Neural Machine Translation with Shared Encoder and Decoder. AAAI 2019.
[9] F. Casacuberta, M. Federico, H. Ney, and E. Vidal, “Recent efforts in spoken language translation,” *IEEE Signal Processing Magazine*, vol. 25, no. 3, 2008.
[10] S. Bansal, H. Kamper, A. Lopez, and S. Goldwater, “Towards speech-to-text translation without speech recognition,” *arXiv preprint arXiv:1702.03856*, Feb. 2017.
[11] L. Duong, A. Anastasopoulos, D. Chiang, S. Bird, and T. Cohn, “An attentional model for speech translation without transcription,” in *Proceedings of NAACL-HLT*, 2016, pp. 949–959
[12] Hasim Sak, Andrew Senior, Kanishka Rao, and Francoise Beaufays, “Fast and accurate recurrent neural network acoustic models for speech recognition,” In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 1468–1472, 2015.
[13] Geoffrey Zweig, Chengzhu Yu, Jasha Droppo, and Andreas Stolcke, “Advances in all-neural speech recognition,” In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 4805–4809, 2017.
[14] Kartik Audhkhasi, Bhuvana Ramabhadran, George Saon, Michael Picheny, and David Nahamoo, “Direct acoustics-to-word models for English conversational speech recognition,” In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 959–963, 2017.
[15] Hagen Soltau, Hank Liao, and Hasim Sak, “Neural speech recognizer: Acoustic-to-word LSTM model for large vocabulary speech recognition,” In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 3707–3711, 2017.
[16] Kartik Audhkhasi, Brian Kingsbury, Bhuvana Ramabhadran, George Saon, and Michael Picheny, “Building competitive direct acoustics-toword models for English conversational speech recognition,” In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 4759–4763, 2018.
[17] Dzmitry Bahdanau, Jan Chorowski, Dmitriy Serdyuk, Philemon
Brakel, and Yoshua Bengio, “End-to-end attention-based large vocabulary speech recognition,” In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 4945–4949, 2015.
[18] Liang Lu, Xingxing Zhang, Kyunghyun Cho, and Steve Renals, “A study of the recurrent neural network encoder-decoder for large vocabulary speech recognition,” In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 3249–3253, 2015.
[19] William Chan, Navdeep Jaitly, Quoc Le, and Oriol Vinyals, “Listen, attend and spell: A neural network for large vocabulary conversational speech recognition,” In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 4960–4964, 2016.
[20] Liang Lu, Xingxing Zhang, and Steve Renals, “On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition,” In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 5060–5064, 2016.
[21] Chung-Cheng Chiu, Tara N. Sainath, YonghuiWu, Rohit Prabhavalkar, Patrick Nguyen, Zhifeng Chen, Anjuli Kannan, Ron J.Weiss, Kanishka Rao, Ekaterina Gonina, Navdeep Jaitly, Bo Li, Jan Chorowski, and Michiel Bacchiani, “State-of-the-art speech recognition with sequenceto-sequence models,” In Proc. International Conference on Acoustics, Speech, and Signal Processing (ICASSP), pp. 4774–4778, 2018.
[22] Albert Zeyer, Kazuki Irie, Ralf Schluter, and Hermann Ney, “Improved training of end-to-end attention models for speech recognition,” In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH), pp. 7–11, 2018.
[23] Kanishka Rao, Hasim Sak, and Rohit Prabhavalkar, “Exploring architectures, data and units for streaming end-to-end speech recognition with RNN-transducer,” In Proc. Automatic Speech Recognition and Understanding Workshop (ASRU), pp. 193–199, 2017.
[24] Hasim Sak, Matt Shannon, Kanishka Rao, and Francoise Beaufays, “Recurrent neural aligner: An encoder-decoder neural network model for sequence to sequence mapping,” In Proc. Annual Conference of the International Speech Communication Association (INTERSPEECH),
pp. 1298–1302, 2017.
[25] Daniel Povey, Arnab Ghoshal, Gilles Boulianne, Lukas Burget, Ondrej Glembek, Nagendra Goel, Mirko Hannemann, Petr Motlicek, Yanmin Qian, Petr Schwarz, et al., “The kaldi speech recognition toolkit,” in IEEE 2011 workshop on automatic speech recognition and understanding. IEEE Signal Processing Society, 2011, number EPFL-CONF-192584.
[26] Shinji Watanabe, Takaaki Hori, Shigeki Karita, et al. “ESPnet_End-to-End_Speech_Processing_Toolkit”
[27] Daniel S. Park, William Chan, Yu Zhang, et al. SpecAugment: A Simple Data Augmentation Method for Automatic Speech Recognition.
[28] Yuguang Wang, Liangliang Shi, Linyu Wei, et al. The Sogou-TIIC Speech Translation System for IWSLT 2018.
[29] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. Attention is all you need. CoRR, 2017.
[30] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In Proceedings of ACL 2016.