The AFRL IWSLT 2020 Systems: Work-From-Home Edition

Brian Ore, Eric Hansen, Timothy Anderson, Jeremy Gwinnup
Air Force Research Laboratory
{brian.ore.1,eric.hansen.5, timothy.anderson.20, jeremy.gwinnup.1}@us.af.mil

Abstract

This report summarizes the Air Force Research Laboratory (AFRL) submission to the offline spoken language translation (SLT) task as part of the IWSLT 2020 evaluation campaign. As in previous years, we chose to adopt the cascade approach of using separate systems to perform speech activity detection, automatic speech recognition, sentence segmentation, and machine translation. All systems were neural based, including a fully-connected neural network for speech activity detection, a Kaldi factorized time delay neural network with recurrent neural network (RNN) language model rescoring for speech recognition, a bidirectional RNN with attention mechanism for sentence segmentation, and transformer networks trained with OpenNMT and Marian for machine translation. Our primary submission yielded BLEU scores of 21.28 on `tst2019` and 23.33 on `tst2020`.

1 Introduction

As part of the evaluation campaign for the 2020 International Workshop on Spoken Language Translation (IWSLT) (Ansari et al., 2020), the AFRL Human Language Technology team submitted an entry to the offline spoken language translation (SLT) task. The goal of this task is to automatically generate cased and punctuated German translations from English audio TED Talks using either a cascade of systems or the end-to-end approach. We chose to build upon our previous work (Ore et al., 2018; Kazi et al., 2016) and adopt the cascade approach of using separate systems to perform speech activity detection, automatic speech recognition (ASR), sentence segmentation, and machine translation (MT). Sections 2 and 3 describe our ASR and MT systems, respectively. Section 4 presents our results on the development set when the data is manually segmented into sentences, and Section 5 descibes our approach to SLT on unsegmented data. Section 6 provides a post-evaluation analysis of our systems based on sentence length, and Section 7 presents our conclusions and future work.

2 Automatic Speech Recognition

This section describes the English ASR system that was developed for the offline speech translation task. First, we sequestered all talks from TEDLIUM-v3 (Hernandez et al., 2018) that were present in `tst2014`, `tst2015`, and `tst2018`. Next, language models (LMs) were estimated on TEDLIUM-v3 and the same subsets of News Crawl and News Discussions described in Ore et al. (2018). The text was formatted as follows:

- Numbers and special symbols were converted to words (*e.g.*, “%” converted to “percent”, “&” converted to “and”, “=” converted to “equals”).
- Punctuation marks and any remaining symbols were removed.
- All text was converted to lowercase.

We used the SRILM `select-vocab` tool[1] to choose a 100,000 word vocabulary. An interpolated bigram language model (LM) was estimated using the SRILM toolkit, and a recurrent neural network (RNN) LM was trained using Kaldi (Povey et al., 2011).

Acoustic models were trained on TEDLIUM-v3 using Kaldi. In a preliminary experiment, we found that training on both TEDLIUM-v3 and CommonVoice did not lead to a reduction in word error rate (WER), so we decided to only use TEDLIUM-v3. The Kaldi system used in these experiments is a factorized time delay neural network (TDNN) with residual network style skip connections. Input Mel frequency cepstral coefficient

[1] Available at: http://www.speech.sri.com/projects/srilm

Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT), pages 103–108
July 9-10, 2020. ©2020 Association for Computational Linguistics
https://doi.org/10.18653/v1/P17

(MFCC) features have standard speed perturbation applied (0.9 & 1.1 factor). The initial Kaldi finite state transducer (FST) was built with the bigram LM, and the resulting lattices were rescored using the pruned RNNLM lattice rescoring algorithm (Xu et al., 2018).

3 Machine Translation

In order to translate the ASR output from the previous section, we construct an English–German MT training corpus from allowable sources provided by the organizers[2]. We then prepare this corpus in a similar manner as described in Gwinnup et al. (2018) and Gwinnup et al. (2019), especially focusing on fastText (Joulin et al., 2016a,b) language-id filtering. As a contrast to Ore et al. (2018) we prepare data for additional systems on this same corpus where the source English text has been transformed to resemble output from our ASR systems. We then train transformer (Vaswani et al., 2017) based MT systems with the OpenNMT (Klein et al., 2018) and Marian (Junczys-Dowmunt et al., 2018) toolkits.

3.1 OpenNMT

The OpenNMT-tf system trained for this task used the default configuration for a transformer network. Two copies of the training data described above were concatenated together. One copy was lowercase and non-punctuated in order to resemble ASR output and an additional copy was cased and with punctuation. This combined corpus was processed with Sentencepiece (Kudo and Richardson, 2018) using a model trained only on the lowercase and non-punctuated corpus. The network was trained for 10 epochs of this training data using a batch size of 1562 with an effective batch size of 24992 using the lazy Adam (Kingma and Ba, 2015) optimizer. The final system was an average of the last 8 checkpoints of the training. Checkpoints were saved every 5000 steps. Results using this models for the punctuated test sets for the WMT news translation task are shown in Table 1 Column A. Column B is results with a model trained only on the cased and punctuated data. Column C is the results with a model trained only on the lowercased unpunctuated data.

Test Set	A	B	C
newstest2018	40.02	43.11	22.41
newstest2019	37.55	38.71	19.69

Table 1: OpenNMT system performance under different training corpus conditions.

Results using this model on `tst2014`, `tst2015`, and `tst2018` with cased and punctuated input are shown in Table 2 Column A. Column B is the results with lowercase and non-punctuated input. Column C is with a model trained only on the cased and punctuated data, and Column D is the results with a model trained only on lowercase non-punctuated data.

Test Set	A	B	C	D
tst2014	27.67	26.99	28.43	26.48
tst2015	29.80	28.85	29.72	28.43
tst2018	27.46	25.53	27.81	25.81

Table 2: OpenNMT system performance under different training corpus conditions.

3.2 Marian

Our Marian systems also utilize the transformer (Vaswani et al., 2017) architecture. Network hyperparameters are the same as detailed in Gwinnup et al. (2018). We use the WMT16 `newstest2016` as the validation set during training.

We used the following network configuration:

- 6 layer encoder
- 6 layer decoder
- 8 transformer heads
- Tied embeddings for source, target and output layers
- Layer normalization
- Label smoothing
- Learning rate warm-up and cool-down

A joint Sentencepiece vocabulary with 46k entries was employed, informed by experimentation performed for our WMT19 efforts. With lowercase non-punctuated input, this system yielded the following BLEU scores: 26.58 on `tst2014`, 28.47 on `tst2015`, and 26.57 on `tst2018`.

[2] The majority of the training corpus comes from the preprocessed WMT18 news-translation task data available here: http://data.statmt.org/wmt18/translation-task/preprocessed/de-en/

	Marian	OpenNMT
tst2014	24.80 (6.4)	24.67 (6.2)
tst2015	26.44 (6.6)	26.14 (6.5)
tst2018	23.91 (9.1)	23.09 (8.6)

Table 3: BLEU scores and WERs (in parentheses) on the manually segmented development sets. The MT systems were trained on lowercase non-punctuated English text.

	OpenNMT
tst2014	23.77 (6.2)
tst2015	24.26 (6.5)
tst2018	22.90 (8.6)

Table 4: BLEU scores and WERs (in parentheses) on the manually segmented development sets using the OpenNMT system trained on cased and punctuated English text.

4 Manual Segmentation

In order to evaluate the effect of automatic sentence segmentation on spoken language translation, we manually segmented the tst2014, tst2015, and tst2018 development sets into sentences using the provided reference files. This was done by automatically aligning the reference text using a Kaldi ASR system and then manually correcting any errors. The Kaldi system described in Section 2 was then used to generate ASR transcripts for each utterance. Note that for ASR tasks, a development set is typically used to select the LM scale that minimizes the WER; however, in this task our goal is to choose the best translation. We decided to generate 8 different hypotheses for each utterance by varying the ASR LM scale over 6, 8, 10, ..., 20, translating each utterance, and then selecting the ASR LM scale that yields the best overall BLEU score. Each ASR hypothesis was translated using the MT systems trained on lowercase non-punctuated English text. Compared to selecting the ASR LM scale to minimize WER, this method yields a very minor improvement with Marian (0.06 BLEU on tst2014 and tst2018, 0.17 BLEU on tst2015), but no improvement with OpenNMT. Table 3 shows the case-sensitive BLEU scores and corresponding WER in parentheses.

In a second set of experiments, an automatic punctuator and text recaser were applied to the English ASR text prior to performing translation. Compared to the previous approach, one advantage of this method is that we can train a single MT system to translate both ASR transcripts and text documents. The punctuator was a bidirectional RNN with attention mechanism that was trained on 4M words of English TED data using the Python fork of Ottokar Tilk's punctuator.[3] The punctuated text was recased using Moses and then translated using the OpenNMT system that was trained on cased and punctuated English. Table 4 shows the BLEU scores and corresponding WER in parentheses. Comparing Tables 3 and 4, we can see that using the translation models trained on lowercase non-punctuated English text yields the best results; therefore, we decided to use these MT systems for all remaining experiments discussed in this paper.

5 Automatic Segmentation

In the previous section, we evaluated our ASR and MT systems on audio clips that were manually segmented into sentences. This section considers the task where we are given an audio stream that must be automatically segmented. First, we evaluated a speech activity detector (SAD) on each audio file. We used the same neural network based SAD as described in our IWSLT 2018 paper. Automatic segmentation of the test data was performed by evaluating the SAD, applying a dynamic programming algorithm to choose the best sequence of states, and defining utterance boundaries at the midpoint of each non-speech segment longer than 0.5 seconds. Next, the Kaldi ASR system was evaluated on each utterance using the same ASR LM scales found in the previous section. Two different methods were investigated for partitioning the time-aligned words into sentences. In the first method, we simply used the utterance boundaries from the SAD to define the sentence boundaries. For the second method, we evaluated the automatic punctuator from Section 4 on each utterance, and then defined additional sentence boundaries at words that ended with a period, exclamation, or question mark.

Table 5 shows the case-sensitive BLEU scores and corresponding WER obtained using each method. Comparing the two sentence segmentation methods, we can see that defining additional sentence boundaries with the punctuator led

[3] Available at: https://pypi.org/project/punctuator

	Marian		OpenNMT	
	SAD	SAD+punctuator	SAD	SAD+punctuator
tst2014	23.32 (7.0)	22.52 (7.0)	22.52 (6.6)	22.86 (6.6)
tst2015	23.93 (6.7)	23.59 (6.7)	23.69 (6.5)	23.90 (6.5)
tst2018	21.60 (9.4)	21.56 (9.4)	20.86 (9.0)	21.32 (9.0)

Table 5: BLEU scores and WERs (in parentheses) obtained on the automatically segmented development sets. Sentence boundaries were defined using (1) the SAD, or (2) a combination of the SAD marks and the automatic punctuator.

to an overall decrease in BLEU when translating with Marian, but an improvement with OpenNMT. Compared to the results obtained with the manual segments in Table 3, we find that the Marian BLEU score decreased by 1.48 on tst2014, 2.51 on tst2015, and 2.31 on tst2018; similarly, the OpenNMT BLEU score decreased by 1.81 on tst2014, 2.24 on tst2015, and 1.77 on tst2018. Lastly, if we compare the OpenNMT systems (which used the same ASR LM scale to minimize WER and maximize BLEU) in Tables 3 and 5, we can see that automatically segmenting the data yields no change in WER on tst2015, and an increase of 0.4% on tst2014 and tst2018.

Our primary submission to the IWSLT 2020 offline speech translation task can be summarized as follows. First, a neural network based SAD was used to segment each audio file into utterances. Next, ASR transcripts were generated using Kaldi and an automatic punctuator was applied to further split each utterance into sentences. Lastly, an OpenNMT system was used to translate the lowercase non-punctuated English into cased and punctuated German. As a contrasting system, we also submitted the translations obtained using Marian. The processing pipeline for the Marian system was identical to the OpenNMT system, except that we did not apply the automatic punctuator (*i.e.*, the sentence boundaries were defined solely on the pause durations from the SAD). The OpenNMT system yielded BLEU scores of 21.28 on tst2019 and 23.33 on tst2020; the Marian system yielded BLEU scores of 21.48 on tst2019 and 23.21 on tst2020.

6 Post-Evaluation Analysis

In Section 5 we found that defining additional sentence boundaries using an automatic punctuator led to a worse performance with Marian, but improved performance with OpenNMT. This led us to wonder if the automatic punctuator was actually helping to identify more correct sentence boundaries, or simply producing shorter sentences that were better translated with OpenNMT. Based on this idea, we decided to analyze how sentence length affects translation performance with each of our systems. First, the English reference text from dev2010, tst2010, tst2013, tst2014, tst2015, and tst2018 was processed using the same steps as described in Section 2 to match the expected MT input. Marian and OpenNMT were then used to translate each sentence, and the BLEU score was calculated for sentences where the English source included 1-9, 10-19, 20-29, 30-39, 40-49, and 50+ words. Table 6 shows the results obtained, including the number of sentences assigned to each group. These results show that for sentences longer than 50 words, the BLEU score drops substantially with the OpenNMT system.

#Words	#Sentences	Marian	OpenNMT
1-9	2266	30.62	28.12
10-19	2889	28.57	28.71
20-29	1456	28.16	28.39
30-39	665	26.55	26.88
40-49	275	25.78	25.34
50+	241	27.10	19.85

Table 6: BLEU scores on the reference text grouped by sentence length.

As a final experiment, we re-evaluated our submitted OpenNMT system, but only inserted additional sentence boundaries if the English ASR utterance was longer than 50 words. This yielded the following BLEU scores: 23.21 on tst2014, 23.89 on tst2015, and 21.37 on tst2018. Compared with the OpenNMT SAD+punctuator results in Table 5, this represents a +0.35 BLEU improvement on tst2014 and similar results on tst2015 and tst2018.

7 Conclusion and Future Work

With our systems, we found that automatic sentence segmentation led to a decrease of up to -2.51 BLEU. The punctuator that we used provides functionality to specify the pause duration after each word when training the punctuator. This could be obtained by automatically aligning the original TED training transcripts; however, due to limited computational resources while working at home, we were not able to investigate this feature. In addition to text features and pause durations, other acoustic features might also prove useful for automatically identifying sentence boundaries. Alternatively, it may be interesting to resegment the MT training data to better match the ASR segmentation, although this would probably have to be done in an automatic fashion to take advantage of available text-only parallel corpora.

References

Ebrahim Ansari, Nguyen Bach, Ondrej Bojar, Roldano Cattoni, Marcello Federico, Christian Federmann, Jiatao Gu, Fei Huang, Ajay Nagesh, Matteo Negri, Jan Niehues, Elizabeth Salesky, Sebastian Stüker, and Marco Turchi. 2020. Findings of the IWSLT 2020 Evaluation Campaign. In *Proceedings of the 17th International Conference on Spoken Language Translation (IWSLT 2020)*, Seattle, USA.

Jeremy Gwinnup, Tim Anderson, Grant Erdmann, and Katherine Young. 2018. The AFRL WMT18 systems: Ensembling, continuation and combination. In *Proceedings of the Third Conference on Machine Translation: Shared Task Papers*, pages 394–398, Belgium, Brussels. Association for Computational Linguistics.

Jeremy Gwinnup, Grant Erdmann, and Tim Anderson. 2019. The AFRL WMT19 systems: Old favorites and new tricks. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 203–208, Florence, Italy. Association for Computational Linguistics.

François Hernandez, Vincent Nguyen, Sahar Ghannay, Natalia A. Tomashenko, and Yannick Estève. 2018. TED-LIUM 3: twice as much data and corpus repartition for experiments on speaker adaptation. *CoRR*, abs/1805.04699.

Armand Joulin, Edouard Grave, Piotr Bojanowski, Matthijs Douze, Hérve Jégou, and Tomas Mikolov. 2016a. Fasttext.zip: Compressing text classification models. *arXiv preprint arXiv:1612.03651*.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2016b. Bag of tricks for efficient text classification. *arXiv preprint arXiv:1607.01759*.

Marcin Junczys-Dowmunt, Roman Grundkiewicz, Tomasz Dwojak, Hieu Hoang, Kenneth Heafield, Tom Neckermann, Frank Seide, Ulrich Germann, Alham Fikri Aji, Nikolay Bogoychev, André F. T. Martins, and Alexandra Birch. 2018. Marian: Fast neural machine translation in C++. In *Proceedings of ACL 2018, System Demonstrations*, pages 116–121, Melbourne, Australia. Association for Computational Linguistics.

Michaeel Kazi, , Elizabeth Salesky, Brian Thompson, Jonathan Taylor, Jeremy Gwinnup, Tim Anderson, Grant Erdmann, Eric Hansen, Brian Ore, Katherine Young, and Michael Hutt. 2016. The MITLL-AFRL IWSLT-2016 systems. In *Proc. of the 13th International Workshop on Spoken Language Translation (IWSLT'16)*, Seattle, Washington.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Guillaume Klein, Yoon Kim, Yuntian Deng, Vincent Nguyen, Jean Senellart, and Alexander Rush. 2018. OpenNMT: Neural machine translation toolkit. In *Proceedings of the 13th Conference of the Association for Machine Translation in the Americas (Volume 1: Research Papers)*, pages 177–184. Association for Machine Translation in the Americas.

Taku Kudo and John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 66–71, Brussels, Belgium. Association for Computational Linguistics.

Brian Ore, Eric Hansen, Timothy Anderson, and Jeremy Gwinnup. 2018. The AFRL IWSLT 2018 systems: What worked, what didn't. In *Proc. of the 15th International Workshop on Spoken Language Translation (IWSLT'18)*, Bruges, Belgium.

D. Povey, A. Ghoshal, G. Boulianne, L. Burget, O. Glembek, N. Goel, M. Hannemann, P. Motlicek, Y. Qian, P. Schwarz, J. Silovsky, G. Stemmer, and K. Vesely. 2011. The Kaldi Speech Recognition Toolkit. In *Proceedings of the Automatic Speech Recognition and Understanding Workshop*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Opinions, interpretations, conclusions and recommendations are those of the authors and are not necessarily endorsed by the United States Government. Cleared for public release on 08 May 2020. Originator reference number RH-20-121048. Case number 88ABW-2020-1696.

Hainan Xu, Tongfei Chen, Dongji Gao, Yiming Wang, Ke Li, Nagendra Goel, Yishay Carmiel, Daniel Povey, and Sanjeev Khudanpur. 2018. A pruned rnnlm lattice-rescoring algorithm for automatic speech recognition. In *2018 IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP)*, Calgary, AB, Canada.