XMU Neural Machine Translation Online Service

Boli Wang, Zhixing Tan, Jinming Hu, Yidong Chen and **Xiaodong Shi***
School of Information Science and Engineering, Xiamen University, Fujian, China
{boliwang, playinf, todtom}@stu.xmu.edu.cn
{ydchen, mandel}@xmu.edu.cn

Abstract

We demonstrate a neural machine translation web service. Our NMT service provides web-based translation interfaces for a variety of language pairs. We describe the architecture of NMT runtime pipeline and the training details of NMT models. We also show several applications of our online translation interfaces.

1 Introduction

Neural machine translation (NMT) (Bahdanau et al., 2015; Cho et al., 2014; Sutskever et al., 2014) has achieved great success in recent years and significantly outperforms statistical machine translation on various language pairs (Sennrich et al., 2016a; Wu et al., 2016; Zhou et al., 2016). More and more companies and institutes begin to deploy NMT engines on their machine translation services (Wu et al., 2016; Crego et al., 2016).

We published our NMT online service[1] on November 2016. Up to now, we support nine translation directions: Simplified Chinese ↔ English, Simplified Chinese ↔ Tibetan, Uyghur → Simplified Chinese, Mongolian → Simplified Chinese, Indonesian → Simplified Chinese, Vietnamese → Simplified Chinese, and Deutsch → English. We have also implemented Simplified-Traditional Chinese conversion in the same framework.

In this paper, we describe the implementation and deployment of our NMT online service. Different from (Junczys-Dowmunt et al., 2016) and (Stahlberg et al., 2017), which introduced fast and usable decoding tools, we mainly focus on the NMT runtime pipeline architecture and the details of training NMT models. We introduce a language-independent NMT service framework, which is capable with different types of neural decoders. We report effective techniques and tricks to optimize the training of NMT models. We also present several applications of using our online service.

2 System Architecture

We implement a language-independent pipeline framework. The pipeline consists of six abstract interfaces: paragraph analyzer, tokenizer, subword segmenter, decoder, detokenizer, and paragraph reconstructor. To deploy a new NMT engine, we only need to implement the corresponding interfaces for the specific language or directly reuse the existing ones.

The **paragraph analyzers** parse the paragraphs into sentences and records the relationship between sentences and paragraphs. We simply implement a rule-based sentence segmenter for each source language.

Tokenizers used in the online runtime must be identical to the one used in the training time. The Moses[2] tokenizers and truecasers are applied on source languages like English and Deutsch and in-home word segmenters are applied on Chinese and Tibetan. For Mongolian and Uyghur, we first tokenize the sentence using our own tokenizer and then latinize and normalize the sentence to reduce the vocabulary.[3] For Simplified-Traditional Chinese conversion, we simply split sentences into characters.

Subword segmenters are effective to reduce the vocabulary and enable the translation of out-

[*] Corresponding author.

[1] http://nmt.cloudtrans.org/

[2] http://statmt.org/moses/

[3] The details of our tokenization method, including word segmentation, latinization, and normalization, have been described in our technical reports of WMT17, CWMT2017 and WAT2017 translation tasks (Tan et al., 2017a,b; Wang et al., 2017).

The Companion Volume of the IJCNLP 2017 Proceedings: System Demonstrations, pages 29–32,
Taipei, Taiwan, November 27 – December 1, 2017. ©2017 AFNLP

of-vocabulary tokens. We implement two different types of subword segmenters. For languages with explicit boundaries between syllables, like Chinese and Tibetan, we use mixed word/character model (Wu et al., 2016). We keep a shortlist of the most frequent words and split other words into syllables. Unlike (Wu et al., 2016), we do not add any extra prefixes or suffixes to the segmented syllables. For languages without explicit boundaries between syllables, like English, Deutsch, Mongolian and Uyghur, we use the BPE method[4] (Sennrich et al., 2016c).

Different beam search **decoders** are implemented to support different types of neural models.

- **Translation model**: Our NMT model is a modified version of `dl4mt`[5]. Therefore, we implement a variant of AmuNMT C++ decoder[6] to support our NMT models and achieve parallel decoding.

- **Transliteration model**: We regard Simplified-Traditional Chinese conversion as a sequence labeling task and resort to a simple transliteration model, which is a single layer bi-directional GRU with a softmax layer on the top. In decoding, we employ a Simplified-Traditional Chinese character conversion table to prune the search space.

We apply the Moses **detokenizer** and truecaser on the output English sentences[7]. We use several heuristic rules to judge whether each space in the output Chinese/Tibetan sentences should be kept or not.

The **paragraph reconstructors** use the results of the paragraph analyzers to restore the output paragraphs.

3 Training Details

3.1 Training Data

We crawled monolingual and parallel data from Internet. We filter out bad sentences and utilize target language monolingual data by back-translation method (Sennrich et al., 2016b).

Before training a translation model, we filter out bad sentence pairs from parallel data according to their ratio of length and alignment scores obtained by `fast-align` toolkit[8].

We use `srilm`[9] to train a 5-gram KN language model on the monolingual data of target language and select monolingual sentences according to their perplexity. We train backward translation models on the parallel data and translate the selected monolingual sentences back to the source language.

In our preliminary experiments, we found that training or tuning on the synthetic training data alone could not improve the performance of NMT models. Therefore, we randomly sample a comparable amount of bilingual sentence pairs from parallel data and mix them up with the synthetic ones.

For resource-rich language pairs, such as Chinese-English, we first train a NMT model on the parallel data and then fine-tune the model on the mixed synthetic data. For low-resource language pairs, we found that tuning pre-trained models on mixed synthetic data can not improve the translation quality. Instead, we directly train NMT models on the mixed synthetic data and achieve significant improvement[10].

For Simplified-Traditional Chinese conversion, we utilize our Traditional Chinese corpora[11] to synthesize training data by using our ruled based Traditional-Simplified Chinese converter[12].

3.2 Hyper-parameters

For translation models, we use Adam optimizer (Kingma and Ba, 2015) ($\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 1\times 10^{-8}$) and set the initial learning rate to 5×10^{-4}. During the training process, we clip the norm of gradient to a predefined value of 5.0 and gradually halve the learning rate. We use dropout (Srivastava et al., 2014) to avoid overfitting with a keep probability of 0.8. For each language pair, we train a variety of NMT models with different data shuffling and random initialization and apply the ensembling method, which is pro-

[4] https://github.com/rsennrich/subword-nmt

[5] https://github.com/nyu-dl/dl4mt-tutorial

[6] https://github.com/emjotde/amunmt

[7] The suffixes adding by BPE segmenters plus the followed spaces are removed first.

[8] https://github.com/clab/fast_align

[9] http://www.speech.sri.com/projects/srilm/

[10] The details of our experiments have been described in our technical reports of WMT17, CWMT2017 and WAT2017 translation tasks (Tan et al., 2017a,b; Wang et al., 2017).

[11] http://cloudtranslation.cc/corpus_tc.html

[12] http://jf.cloudtranslation.cc/

posed by (Sutskever et al., 2014), to generate better translation.

For transliteration models, the settings are almost the same as above, except that we use the RMSprop optimizer (Tieleman and Hinton, 2012) and do not use dropout and ensemble technique.

4 Applications

Using our free online translation interface[13], developers can easily access to NMT engines. We have developed several applications using the NMT interface:

- **Web Page Translator**: We have published a free web page translation interface[14] using NMT engines. When a client requests an URL, we first crawl the web page and extract the text contents, and then call the NMT interface to get the corresponding translations and replace the source contents.

- **Speech-to-speech Translator**: We have published a free speech-to-speech translation service on WeChat Platform as an official account named *self-talker*. When receiving audio messages from users, we use the speech recognition feature of WeChat Platform to get the recognition results and call our NMT interface to get the translation, then pass into Buidu TTS API[15] to synthesize the speech and response to the user. Currently, we only support Chinese-English translation.

- ***Yunyi* CAT Platform**: *Yunyi* is our computer-aided translation platform. Traditionally, CAT systems use example-based or statistics-based MT engines as their backends. Now, on *Yunyi* platform, we provide human translators with NMT engines to achieve better translations and less efforts of post-editing.

5 Conclusion and Future Works

We presented the architecture and training details of our NMT online service, as well as several applications of using our translation interfaces. Currently, We have completed our main implementation and are in the process of testing new features, including the incremental update of translation models and the support of user-defined translation memories and lexicons. We plan to support more language pairs in the future, especially the low-resource ones, including $\{$Thai, Malay, Hindi$\}$ $\leftrightarrow$ Simplified Chinese.

Acknowledgments

This work was supported by the Natural Science Foundation of China (Grant No. 61573294), the Ph.D. Programs Foundation of Ministry of Education of China (Grant No. 20130121110040), the Foundation of the State Language Commission of China (Grant No. WT135-10) and the National High-Tech R&D Program of China (Grant No. 2012BAH14F03).

References

Dzmitry Bahdanau, KyungHyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *Proceedings of EMNLP*, pages 1724–1734.

Josep Maria Crego, Jungi Kim, Guillaume Klein, Anabel Rebollo, Kathy Yang, Jean Senellart, Egor Akhanov, Patrice Brunelle, Aurelien Coquard, Yongchao Deng, Satoshi Enoue, Chiyo Geiss, Joshua Johanson, Ardas Khalsa, Raoum Khiari, Byeongil Ko, Catherine Kobus, Jean Lorieux, Leidiana Martins, Dang-Chuan Nguyen, Alexandra Priori, Thomas Riccardi, Natalia Segal, Christophe Servan, Cyril Tiquet, Bo Wang, Jin Yang, Dakun Zhang, Jing Zhou, and Peter Zoldan. 2016. Systrans pure neural machine translation systems. *arXiv preprint arXiv:1610.05540*.

Marcin Junczys-Dowmunt, Tomasz Dwojak, and Hieu Hoang. 2016. Is neural machine translation ready for deployment? A case study on 30 translation directions. In *Proceedings of the 9th International Workshop on Spoken Language Translation (IWSLT)*.

Diederik Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *Proceedings of ICLR*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Edinburgh neural machine translation systems for WMT 16. *arXiv preprint arXiv:1606.02891*.

[13] http://nmt.cloudtrans.org/nmt?src=<UrlEncodedSourceText>&lang=<LanguagePairCode>

[14] http://nmt.cloudtrans.org/url?url=<UrlEncodedOriginalUrl>&dir=<LanguagePairCode>

[15] http://yuyin.baidu.com/

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Improving neural machine translation models with monolingual data. In *Proceddings of ACL*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016c. Neural machine translation of rare words with subword units. In *Proceedings of ACL*, pages 1715–1725.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *The Journal of Machine Learning Research*, 15(1):1929–1958.

Felix Stahlberg, Eva Hasler, Danielle Saunders, and Bill Byrne. 2017. SGNMT – a flexible NMT decoding platform for quick prototyping of new models and search strategies. *arXiv preprint arXiv:1707.06885*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Zhixing Tan, Boli Wang, Jinming Hu, Yidong Chen, and Xiaodong Shi. 2017a. XMU neural machine translation systems for WMT 17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*, pages 400–404.

Zhixing Tan, Boli Wang, Xiansong Ji, Bingyansen Wu, Jinming Hu, Yidong Chen, and Xiaodong Shi. 2017b. XMU neural machine translation systems for CWMT 2017. In *Proceddings of the 13th China Workshop on Machine Translation*.

Tijmen Tieleman and Geoffrey Hinton. 2012. Lecture 6.5-rmsprop: Divide the gradient by a running average of its recent magnitude. *COURSERA: Neural Networks for Machine Learning*.

Boli Wang, Zhixing Tan, Jinming Hu, Yidong Chen, and Xiaodong Shi. 2017. XMU neural machine translation systems for WAT 2017. In *Proceedings of the 4th Workshop on Asian Translation (WAT2017)*.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Jie Zhou, Ying Cao, Xuguang Wang, Peng Li, and Wei Xu. 2016. Deep recurrent models with fast-forward connections for neural machine translation. *Transactions of the Association for Computational Linguistics*, 4:371–383.