NJUNLP's Machine Translation System for CCMT-2020 Uighur → Chinese Translation Task

Dongqi Wang, Zihan Liu, Qingnan Jiang, Zewei Sun, Shujian Huang(✉), and Jiajun Chen

National Key Laboratory for Novel Software Technology, Nanjing University, Nanjing, China
{wangdq,liuzh,jiangqn,sunzw}@smail.nju.edu.cn
{huangsj,chenjj}@nju.edu.cn

Abstract. This paper describes our submitted systems for CCMT-2020 shared translation tasks. We build our neural machine translation system based on Google's Transformer architecture. We also employ some effective techniques such as back translation, data selection, ensemble translation, fine-tuning and reranking to improve our system.

Keywords: Neural machine translation · Back translation · Fine-tuning · Ensemble translation · Reranking

1 Introduction

Neural networks have shown their superiority on machine translation [1,18] and other natural language processing tasks [5]. Self-attention based Transformer [19] has been the dominant architecture for neural machine translation. This paper describes our submission for CCMT-2020 Uighur → Chinese translation task.

We build our system based on Transformer [19] due to its superior performance and parallelism. Several techniques which have been proved effective are employed to boost the performance of our system.

We apply Byte Pair Encoding (BPE) [15] to reduce the sizes of vocabularies and achieve open-vocabulary translation. Tagged back-translation with top-k sampling [2,7,14] is used to improve translation performance with monolingual data. We also train several variants of Transformer such as DynamicConv [20] and Transformer with relative position representations. We select back-translated data by length and alignment features. We average the parameters of several best checkpoints [3] in a single training process to get a better single model. Translation models trained on mixed data are fine-tuned on real data provided by the evaluation organizer. Finally, we translate source texts by ensemble several best performing models and rerank the n-best lists with K-batched MIRA algorithm [4].

With above techniques, our system evaluated with BLEU [13] improved for a large margin. We also tried a few methods used in other neural machine translation systems without seeing significant improvements.

J. Li and A. Way (Eds.): CCMT 2020, CCIS 1328, pp. 76–82, 2020.
https://doi.org/10.1007/978-981-33-6162-1_7

2 Machine Translation System

Since there is far less parallel data for Uighur → Chinese translation, we adopt several effective techniques for alleviating data starvation problem. The following sections describe how we build a well-performing system for Uighur → Chinese translation in low-resource scenario.

2.1 Pre-processing

Table 1. Statistics of pre-processed parallel data.

Translation direction	#Sentence pairs
Uighur → Chinese	165792
Uighur → Chinese (sample 1)	6340403
Uighur → Chinese (sample 2)	6340804

We escape special characters and normalize punctuation characters with Moses [10][1]. Then we tokenize sentences for Chinese with pkuseg [12][2]. Sentences with more than 100 words were removed for both Uighur and Chinese. We also filter parallel data where Chinese sentence is 6 times longer than Uighur sentence or Uighur sentence is 4 times longer than Chinese. We learn word alignment with fast_align [6][3] and filter sentence pairs whose alignment rates are less than 0.6. The statistics of pre-processed parallel data are shown in Table 1. The remaining data is processed by Byte Pair Encoding [15][4], with 32K merge operations for both Uighur and Chinese.

2.2 Architecture

Table 2. Architecture hyper-parameters of Transformer Big in our system.

Hyper-parameter name	Hyper-parameter value
Embedding size	1024
Hidden size	1024
Ffn inner size	4096
Attention heads	16
Dropout	0.2
Label smoothing	0.1

[1] https://github.com/moses-smt/mosesdecoder.
[2] https://github.com/lancopku/pkuseg-python.
[3] https://github.com/clab/fast_align.
[4] https://github.com/rsennrich/subword-nmt.

We adopt Transformer Big as our base model and tune a few architecture hyper-parameters in current setting, which are shown in Table 2. We train all models by optimizing cross entropy loss with label smoothing. Adam optimizer [9] ($\beta 1 = 0.9$, $\beta 2 = 0.98$, $\epsilon = 10^{-9}$) was used for optimization. Learning rate is linearly increased during the first 4000 steps, and then decreased with inverse square root function of steps as in [19]. We train all models on 4 NVIDIA Tesla V100 GPUs.

To obtain more diversed models for ensembling, we train two variants of vanilla Transformer: Transformer with relative position representations [16] (Relative Transformer) and DynamicConv [20]. Checkpoint averaging [3] is also used to get a stronger model.

2.3 Back-Translation of Monolingual Data

Back-translation has been proved as an effective method for data augmentation of neural machine translation [7,14], especially in low-resource scenarios. With only 165K provided parallel data, Transformer Big performs worse than Transformer Base, seeing Table 3. We train a Chinese → Uighur translation model, taking Transformer Base architecture. Then we apply the trained Transformer to translate large scale monolingual sentences in Chinese to Uighur and construct pseudo Uighur → Chinese translation parallel data.

Table 3. Back-translation with different strategies

Setting	BLEU
Transformer Base w\o BT	38.56
Transformer Big w\o BT	37.01
Transformer Big w\BT (beam search)	45.27
Transformer Big w\BT (top-10 sampling)	45.34
Transformer Big w\BT (top-10 sampling) + tag	46.00

We experiment with several methods to generate synthetic data as proposed in [7], such as beam search and top-k sampling. We find top-k sampling is more effective as shown in Table 3. A possible explanation is that top-k sampling introduce moderate noise into synthetic data, which makes pseudo data generated by top-k sampling contain stronger training signal [8].

It is useful to distinguish real data and synthetic data during training since synthetic data is usually more noised. A simple method distinguish real data and synthetic data is adding a tag in front of each sentences, which is called Tagged Back-Translation [2]. Experimental results in Table 3 proved its effectiveness in Uighur → Chinese translation.

We construct two synthetic datasets (named sample1 and sample2) by top-10 sampling in back-translation and filter sentence pairs with length and alignment features.

2.4 Fine-Tuning

Table 4. Fine-tuning trained models on real data

Model	Before tuning	After tuning
Transformer (sample 1)	46.00	46.96
Transformer (sample 2)	45.32	46.76
Checkpoint averaging (sample 1)	45.53	47.26
Relative transformer (sample 1)	45.56	47.53
Relative transformer (sample 2)	45.40	47.43
Dynamic convolution (sample 1)	45.42	46.82

There is domain divergence between real data and synthetic data, since synthetic sentence pairs are in general domain while real data specific in news domain. We fine-tune translation models trained on mixed data on real data to adapt them specific to target domain.

As indicated in Table 4, fine-tuning trained model on real data boost the performance of translation models for a large margin evaluated by BLEU scores on development set.

2.5 Ensemble Translation

Many literatures [1,18] have shown the effectiveness of ensemble learning for improving translation quality. We translate evaluation source texts by ensembling several diversed and best performing models. Our experimental results in Table 5 present stable increments of translation quality with ensembling more best performing models.

Table 5. Ensemble translation: index i means the i-th model in Table 4

Ensemble selection	BLEU
$4 + 5$ (beam size $= 5$)	47.97
$3 + 4 + 5$ (beam size $= 5$)	48.21
$1 + 3 + 4 + 5$ (beam size $= 5$)	48.51
$1 + 3 + 4 + 5 + 6$ (beam size $= 5$)	48.54
$1 + 2 + 3 + 4 + 5 + 6$ (beam size $= 5$)	48.63
$1 + 2 + 3 + 4 + 5 + 6$ (beam size $= 24$)	48.80

2.6 Reranking

We generate the n-best translation lists by ensembling 6 best performing models with beam size = 24. We hand-craft several features for reranking the n-best lists, including log probability of each single translation model, target-to-source translation score, right-to-left translation score [11], n-gram language model perplexity[5] and beam index. The reranking model is tuned by K-batched MIRA algorithm [4]. BLEU score evaluated on development set achieves 49.17 after reranking.

3 Results

Table 6 shows our systems evaluated by BLEU on development set. For Uighur → Chinese translation, BLEU scores [13] are computed at character level. For the last 4 rows, each model is based on the model described in the previous row.

Table 6. Translation quality evaluated by BLEU on development set

System	Uighur → Chinese
Transformer Base	38.56
Transformer Big	37.01
+ Back Translation	46.00
+ Fine-tuning	46.96
+ Ensembling	48.80
+ Reranking	49.17

We can see that back-translation, fine-tuning, ensemble translation and reranking consistently boost the performance of the Uighur → Chinese translation system. During these techniques, back-translation is most effective in low-resource scenario.

4 Conclusion

This paper presents our submission for CCMT-2020 Uighur → Chinese translation task. We obtain a strong baseline system by tuning Google's Transformer Big architecture and continually improve it by back-translation, fine-tuning, ensembing and reranking.

[5] https://github.com/kpu/kenlm.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In: ICLR 2015: International Conference on Learning Representations 2015 (2015)
2. Caswell, I., Chelba, C., Grangier, D.: Tagged back-translation. In: Proceedings of the Fourth Conference on Machine Translation (Volume 1: Research Papers), pp. 53–63 (2019)
3. Chen, H., Lundberg, S., Lee, S.I.: Checkpoint ensembles: ensemble methods from a single training process. arXiv preprint arXiv:1710.03282 (2017)
4. Cherry, C., Foster, G.: Batch tuning strategies for statistical machine translation. In: Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 427–436 (2012)
5. Devlin, J., Chang, M.W., Lee, K., Toutanova, K.: Bert: pre-training of deep bidirectional transformers for language understanding. In: NAACL-HLT 2019: Annual Conference of the North American Chapter of the Association for Computational Linguistics, pp. 4171–4186 (2019)
6. Dyer, C., Chahuneau, V., Smith, N.A.: A simple, fast, and effective reparameterization of IBM model 2. In: Proceedings of the 2013 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, pp. 644–648 (2013)
7. Edunov, S., Ott, M., Auli, M., Grangier, D.: Understanding back-translation at scale. In: EMNLP 2018: 2018 Conference on Empirical Methods in Natural Language Processing, pp. 489–500 (2018)
8. Hu, B., Han, A., Zhang, Z., Huang, S., Ju, Q.: Tencent minority-mandarin translation system. In: Huang, S., Knight, K. (eds.) CCMT 2019. CCIS, vol. 1104, pp. 93–104. Springer, Singapore (2019). https://doi.org/10.1007/978-981-15-1721-1_10
9. Kingma, D.P., Ba, J.L: Adam: a method for stochastic optimization. In: ICLR 2015: International Conference on Learning Representations 2015 (2015)
10. Koehn, P.: Open source toolkit for statistical machine translation. In: Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions, pp. 177–180 (2007)
11. Liu, L., Utiyama, M., Finch, A.M., Sumita, E.: Agreement on target-bidirectional neural machine translation. In: 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL HLT 2016 - Proceedings of the Conference, pp. 411–416 (2016)
12. Luo, R., Xu, J., Zhang, Y., Ren, X., Sun, X.: PKUSEG: a toolkit for multi-domain Chinese word segmentation. arXiv preprint arXiv:1906.11455 (2019)
13. Papineni, K., Roukos, S., Ward, T., Zhu, W.J.: BLEU: a method for automatic evaluation of machine translation. In: Proceedings of 40th Annual Meeting of the Association for Computational Linguistics, pp. 311–318 (2002)
14. Sennrich, R., Haddow, B., Birch, A.: Improving neural machine translation models with monolingual data. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 86–96 (2016)
15. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), vol. 1, pp. 1715–1725 (2016)

16. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: NAACL HLT 2018: 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, volume 2, pp. 464–468 (2018)
17. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., Salakhutdinov, R.: Dropout: a simple way to prevent neural networks from overfitting. J. Mach. Learn. Res. **15**(1), 1929–1958 (2014)
18. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In: Advances in Neural Information Processing Systems, vol. 27, pp. 3104–3112 (2014)
19. Vaswani, A.: Attention is all you need. In: Proceedings of the 31st International Conference on Neural Information Processing Systems, pp. 5998–6008 (2017)
20. Wu, F., Fan, A., Baevski, A., Dauphin, Y.N. and Auli, M.: Pay less attention with lightweight and dynamic convolutions. In: ICLR 2019: 7th International Conference on Learning Representations (2019)