Towards Building a Strong Transformer Neural Machine Translation System

Qiang Wang[1,2], Bei Li[1], Jiqiang Liu[1], Bojian Jiang[1], Zheyang Zhang[1], Yinqiao Li[1,2], Ye Lin[1], Tong Xiao[1,2(✉)], and Jingbo Zhu[1,2]

[1] Natrual Language Processing Lab, Northeastern University, Shenyang, China
{libeinlp,liujiqiang,jiangbojian,zhangzheyang,liyinqiao,linyeneu}@stumail.neu.edu.cn, {xiaotong,zhujingbo}@mail.neu.edu.cn
[2] NiuTrans Inc., Shenyang, China
wangqiangneu@gmail.com

Abstract. Transformer model based on self-attention mechanism [17] has achieved state-of-the-art in recent evaluations. However, it is still unclear how much room there is for improvement of the translation system based on this model. In this paper we further explore how to build a stronger neural machine system from four aspects, including architectural improvements, diverse ensemble decoding, reranking, and post-processing. Experimental results on CWMT-18 Chinese ↔ English tasks show that our approach can consistently improve the translation performance of 2.3–3.8 BLEU points than the strong baseline. Particularly, we find that ensemble decoding with a large number of diverse models is crucial for significant improvement.

1 Introduction

Neural machine translation (NMT) exploits an encoder-decoder framework to model the whole translation process in an end-to-end fashion, and has achieved state-of-the-art performance in many language pairs [14,19]. Among various translation models, the Transformer model [17] based on self-attention mechanism has shown promising results in terms of both translation performance and training speed, compared with previous counterparts, such as GNMT [19] or ConvS2S [3].

Although Transformer has achieved great success, it is still unclear that how much room there is for further improvement of the translation system based on it. To answer this issue, we firstly build a strong single Transformer model equipping with some of the existing technologies as baseline. Concretely, we enhance the baseline with checkpoint ensemble [14] that averages the last N checkpoints of a single training run. To enable open-vocabulary translation, all the words are segmented via byte pair encoding (BPE) [13] for both Chinese and English. Also, we use back-translation technique [12] to leverage the rich monolingual resource. As a result, our baseline can achieve comparable performance with the best reported results in CWMT-17.

J. Chen and J. Zhang (Eds.): CWMT 2018, CCIS 954, pp. 101–110, 2019.
https://doi.org/10.1007/978-981-13-3083-4_10

To exceed the strong baseline, we further explore to improve the system from four aspects, including architectural improvements, diverse ensemble decoding, reranking and post-processing. For architectural improvements, we add relu dropout and attention dropout to improve the generalization ability and increase the inner dimension of feed-forward neural network to enlarge the model capacity [4]. We also use the novel Swish activation function [8] and self-attention with relative positional representations [15]. Next, we explore more diverse ensemble decoding via increasing the number of models and using the models generated by different ways. Furthermore, at most 17 features tuned by MIRA [2] are used to rerank the N-best hypotheses. At last, a post-processing algorithmic is proposed to correct the inconsistent English literals between the source and target sentence. Through these techniques, we can achieve consistent improvement of 2.3–3.8 BLEU points over the baselines. Particularly, we find that ensemble decoding with a large number of diverse models is crucial for significant improvement.

2 The Transformer System

Unlike usual NMT models, Transformer does not require any recurrent units for modeling word sequences of arbitrary length. Instead, it resorts to self-attention and standard feed-forward networks for both encoder and decoder.

On the encoder side, there are L identical stacked layers. Each of them is composed of a self-attention sub-layer and a feed-forward sub-layer. The attention model used in Transformer is scaled dot-product attention[1]. Its output is fed into a position-wise feed-forward network. To ease training, layer normalization [1] and residual connection [5] are used for the output of each sub-layer: $\mathrm{LayerNorm}(x + \mathrm{sublayer}(x))$. Likewise, the decoder has another stack of L identical layers. It has an encoder-decoder attention sub-layer in addition to the two sub-layers used in each encoder layer.

Given this model, we can use the cross entropy loss as the training objective, and make the network learn to update parameters by stochastic gradient descend (SGD).

3 Improvements

We improve the baseline system from four aspects, including architectural improvements, ensemble decoding, reranking and post-processing.

[1] Given a sequence of vectors and a position i, the self-attention model computes the dot-product of the input vectors for each pair of positions (i, j), followed by a rescaling operation and Softmax. In this way, we have an attention score (or weight) for each (i, j). It is then used to generate the output by a weighted sum over all input vectors.

3.1 Architectural Improvements

Dropout. The original Transformer only uses residual dropout when the information flow is added between two adjacent layers/sublayers, while the dropouts in feed-forward neural network (e.g. relu dropout) and self attention weights (e.g. attention dropout) are not in use. In practice, we observed the consistent improvements than baseline when we set relu dropout to 0.1 and attention dropout to 0.1, thanks to the regularization effect to overcome the overfitting.

Table 1. Samples of the inconsistent translation of the constant literal between source and target sentence. The subword is split by "@@". The two samples are picked up from CWMT-18 test set.

Source:	ł9P@@ ass@@ p@@ ort
Translation:	so there is the Pas@@ port , which was released last September .
Post-Processing:	so there is the Passport , which was released last September .
Source:	Furious residents have savaged Sol@@ i@@ hull Council saying it was " useless at dealing with the problem ".
Translation:	Ł S@@ ol@@ i@@ h@@ ou@@ s@@ , " ł Ł "
Post-Processing:	Ł Solihull , " ł Ł "

Larger Feed-Forward Network. Limited by the size of GPU memory, we can not directly train a big Transformer model with the batch size as large as the base model. To solve this, we resort to increase the inner dimension (refer to d_{ff}) of feed-forward network while other settings stay the same. It is consistent with the finding of [4] that the transformer model can benefit from larger d_{ff}.

Swish Activation Function. The standard Transformer model has a non-linear expression capability due to the use of Rectified Linear Unit (ReLU) activation function. Recently, [8] propose a new activation function called Swish by the network automatic search technique based on reinforcement-learning. They claim that Swish tends to work better than ReLU on deeper models and can transfer well to a number of challenging tasks. Formally, Swish is computed as:

$$Swish(x) = x \cdot sigmoid(\beta x),$$

where β is either a constant or a learnable parameter. In practice, we replace ReLU with Swish ($\beta = 1$) and do not change any other settings.

Relative Positional Representation. Transformer uses the absolute position encodings based on sinusoids of varying frequency, while [15] point out that the representations of relative position can yield consistent improvement over the absolute counterpart. They equip the representations of both key and value

with some trainable parameters (e.g. a_{ij}^K, a_{ij}^V in [15]) when calculating the self attention. We re-implement this model, and use clipping distance $k = 16$ with the unique edge representations per layer and head. We use both the absolute and relative positional representations simultaneously.

3.2 Diverse Ensemble Decoding

Ensemble decoding is a widely used technique to boost the performance by integrating the predictions of several models, and has been proved effective in the WMT competitions [10,11,18]. Existing experimental results about ensemble decoding mainly concentrate upon a small number of models (e.g. 4 models [10,14,18]). Besides, the ensembled models generally lack of sufficient diversity, for example, [14] use the last N checkpoints of a single training run, while [18] use the same network architecture with different random initializations.

In this paper, we study the effects of more diverse ensemble decoding from two perspectives: the number of models and the diversity of integrated models. We explore at most 16 models for jointly decoding by allocating two models per GPU device in our C++ decoder. In addition to using different random seeds, the ensembled models are generated from more diverse ways, such as different training steps, model sizes and network architectures (see Sect. 3.1).

Every ensembled model is also assigned a weight to indicate the confidence of prediction. In practice, we simply assign the same weight 1.0 for each model. We also study the greedy tuning strategy (randomly initialize all weights firstly, then fix other weights and only tune one weight each time), while there is no significant improvement observed.[2]

3.3 Reranking

We apply the reranking module to pick up a potentially better hypothesis from the n-best generated by ensemble decoding. The used features for reranking include:

- TFs: Translation features. We totally use eight types of translation features, and each type can be represented as a tuple with four elements: (L_s, D_s, L_t, D_t), where $L_s, L_t \in \{ZH, EN\}$ denotes the language of source and target respectively, and $D_s, D_t \in \{L2R, R2L\}$ denotes the direction of source and target sequence respectively. For example, (ZH, L2R, EN, R2L) denotes a system trained on ordinal Chinese → reversed English.
- LM: 5-gram language model of target side[3].
- SM: Sentence similarity. The best hypothesis from the target R2L system is compared to each n-best hypothesis and used to generate a sentence similarity score based on the cosine of the two sentence vectors. The sentence vector is represented by the mean of all word embeddings.

[2] We do not use some more sophisticated tuning methods, such as MERT, MIRA, due to the expensive cost for ensemble decoding, especially with a large beam size.

[3] All language models are trained by KenLM [6].

Given the above features, we calculate the ranking score by a simple linear model. All weights are tuned on the development set via MIRA. The hypothesis with the highest ranking score is chosen as the refined translation.

3.4 Post-processing

Current NMT system generates the translation word by word[4], which is difficult to guarantee the consistency of some constant literals between source sentence and its translation.

In this section, we focus on the English literals in a Chinese sentence. For example, as shown in Table 1, the literal "Passport" in Chinese sentence is translated into "Pasport" wrongly, and a similar error happens between "Solihull" and its translation "Solihous".

Algorithm 1. Post-processing algorithmic for inconsistent English literals translation.

Input: S: source sentence; T: NMT translation;
Output: T': translation after post-processing
1: Initialize: $T' = T$, create $\mathbb{S}(x, y)$ saves the similarity between x and y
2: Get the set of English literals $\mathbb{EL}$ from Chinese sentence (either S or T)
3: **for** each English literal el in $\mathbb{EL}$ **do**
4: **if** el not in T **then**
5: **for** each y in the set of n-gram of T $(1 \leq n \leq 3)$ **do**
6: $\mathbb{S}(el, y) = sim(el, y)$
7: **end for**
8: **end if**
9: $y^* = argmax_y \mathbb{S}(el, y)$
10: replace el with y^* in T'
11: **end for**

To solve this issue, we propose a post-processing method to correct the unmatched translations for the constant literals, as shown in Algorithm 1. The basic idea is that the English literals appearing in Chinese sentence must be contained in English sentence. The challenge is that how to align the correct literal with its wrong one. In practice, we compute the normalized edit distance as the similarity:

$$sim(x, y) = \frac{D(x, y)}{L_x}, \tag{1}$$

where $D(x, y)$ denotes the edit-distance between x and y, L_x is the length of x. Then, the most similar translated literal is recovered by the original one.

Since the number of Chinese sentences containing the English literals is relatively small, our approach can not significantly improve the BLEU, but we find that it is very effective for human evaluation.

[4] Actually it is subword by subword in this paper.

4 Experiments and Results

4.1 Implement Details

Our systems are based on Transformer [17] implemented on the Tensor2Tensor[5]. We use base Transformer model as described in [17]: 6 blocks in the encoder and decoder networks respectively (word representations of size 512, feed-forward layers with inner dimension 2048, 8 attention heads, residual dropout is set to 0.1). We use negative Maximum Likelihood Estimation (MLE) as loss function, and train all the models using Adam [7] with $\beta_1 = 0.9$, $\beta_2 = 0.98$, and $\epsilon = 10^{-9}$. The learning rate is scheduled as described in [17]: $lr = d^{-0.5} \cdot min(t^{-0.5}, t \cdot 4000^{-1.5})$, where d is the dimension of word embedding, t is the training step number. To enable the open-vocabulary translation, we use byte pair encoding (BPE) [13] for both Chinese and English. All the models are trained for 15 epochs on one machine with 8 NVIDIA 1080 Ti GPUs. We limit source and target tokens per batch to 4096 per GPU, resulting in approximate 25,000 source and 25,000 target tokens in one training batch. We also use checkpoint ensemble by averaging the last 15 checkpoints, which are saved at 10-minute intervals.

For evaluation, we use beam search with length normalization [19]. By default, we use beam size of 12, while the coefficient of length normalization is tuned on development set. We use the home-made C++ decoder as a more efficient alternative to the tensorflow implementation, which is also necessary for our diverse ensemble decoding (Sect. 3.2). The hypotheses with too many consecutive repeated tokens (e.g. beyond the count of the most frequent token in the source sentence) are removed. We report all experimental results on CWMT-18 development set by tokenized case-sensitive BLEU-4 metric[6].

Table 2. Statistics of the training data

Direction	Lang.	Sentences	Tokens	Ave. sentence length
ZH → EN	ZH	7.2M	130M	17.6
	EN	7.2M	134M	18.6
EN → ZH	EN	16.9M	505M	29.9
	ZH	16.9M	465M	27.5

[5] https://github.com/tensorflow/tensor2tensor/tree/v1.0.14. We choose this version because we found that this implementation is more similar to the original model described in [17] than newer versions.

[6] https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl.

4.2 Chinese → English Results

For Chinese → English task, we only use the CWMT corpus as training data. All texts are segmented by home-made word segmentation toolkit[7]. We remove the parallel sentence pairs which is duplicated, exceptional length ratio, or bad alignment score obtained by fast-align[8]. Detailed statistical information of training data is shown in Table 2. Then we learn BPE codes with 32k merge operations from independent Chinese and English text, resulting in the size of source and target vocabulary is 47K and 33K respectively. We also study the effect of merge operations, however no significant gain is found when we shrink or expand the number of merge operations.

Table 3. BLEU scores [%] on CWMT-18 development set for Chinese-English translation.

System		beam size	Valid.
Baselines	Transformer-Base	12	24.40
	+checkpoint ensemble	12	24.93
Architecture	$+d_{ff}$ =4096	12	25.31
	+dropout	12	25.75
	baseline+big	12	24.81
	baseline+swish	12	24.98
	baseline+rpr	12	25.21
	baseline+dropout	12	25.63
	baseline+swish+dropout	12	25.52
	baseline+rpr+dropout	12	25.60
Diverse Decoding	4 same models with different random seeds	12	27.40
	4 diverse models	12	27.51
	4 diverse models with large beam	100	27.64
	8 diverse models	100	27.91
	16 diverse models	80	28.12
Re-ranking	17 features	-	28.75
Post-processing	English literal revised	-	**28.76**

Table 3 presents the BLEU scores on CWMT-18 development set of Chinese → English task. First of all, we can see that using checkpoint ensemble brings +0.53 BLEU than the baseline of single model. When we equip the Transformer base model with larger d_{ff} and relu & attention dropout, +0.82 BLEU are improved further. However, to our disappointment, we do not observe consistent improvement via Swish or relative positional representations. Note that the big Transformer model with small batch size (e.g. 2048) even underperforms our baseline, which indicates that the batch size is an essential factor to train a good Transformer model. It is important to notice that although not all model

[7] For Chinese, the word segmentation is done based on unigram language model with Viterbi algorithm.

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

variants can improve the performance orthogonally, we can apply the diversity between these variants to improve other aspects (e.g. ensemble decoding).

Based on the strong single model baseline, we firstly study the conventional ensemble decoding: 4 models with different random seeds, resulting in a significant gain of 1.65 BLEU point. Then we use 4 models with different architectures: *baseline*, $d_{ff} = 4096$, *dropout* and $d_{ff} = 4096 + \text{dropout}$, then an interesting result is that the diverse ensemble decoding is superior than the ensemble of $d_{ff} = 4096 + dropout$, which provides an evidence that diverse models may be more important than homogeneous strong models. The beam size of 100 is a bit better than 12. This result is inconsistent with previous work claiming that larger beam size can badly drop down the performance [16], which needs to be invested further. Additionally, we expand the number of models from 4 to 8 and 16[9], the overall performances are further improved +0.27 and +0.52 respectively. For 16 models ensemble decoding, we arrange every two models on one GPU via our C++ decoder.

Then we rerank the n-best from diverse ensemble decoding (at most 80 candidates) with 17 features[10], we achieve +0.57 BLEU improvement thanks to the complementary information brought by the features. At last, we do post-processing for the reranking output, but almost no effect on BLEU due to limited English literals are found in Chinese sentences (Table 4).

Table 4. BLEU scores [%] on CWMT-18 development set for English → Chinese translation.

System		beam size	Valid.
Baselines	Transformer-Base	12	24.83
	+checkpoint ensemble	12	25.33
Architecture	+d_{ff}=4096	12	25.77
	+dropout	12	25.85
Diverse Decoding	4 same models with different random seeds	12	26.48
	4 diverse models	12	26.74
	4 diverse models + big beam	100	26.68
	10 diverse models	80	27.18
Re-ranking	4 features	-	27.51
Post-processing	English literal revised	-	**27.61**

4.3 English → Chinese Results

For English → Chinese translation, in addition to the CWMT corpus, we additionally uses part of UN and News-Commentary combined data and pseudo

[9] The types of used models include *baseline*, d_{ff}, *dropout*, $d_{ff} + dropout$, *Swish*, *RPR* (relative position representation), *big* (Transformer big model with small batch size) and baseline-epoch20 (training 20 epochs rather than 15).

[10] Seven (ZH, EN, L2R, L2R) models, four (ZH, EN, L2R, R2L) models, one (ZH, EN, R2L, L2R) feature, one (ZH, EN, R2L, R2L) feature, one (EN, ZH, R2L, L2R) feature, one (EN, ZH, R2L, R2L) feature, one LM feature, one SM feature.

parallel data from back-translation. The UN and News-Commentary combined data is selected by XenC [9][11] according to the *xmu* Chinese monolingual corpus from CWMT, and *xin_cmn* monolingual corpus is used for back-translation. Data preprocessing is same as Sect. 4.2, resulting in 7.2M CWMT corpus, 3.5M UN and News-Commentary combined corpus, and 6.2M pseudo parallel data. Then 32k merge operations are used for BPE.

Like Chinese → English, using checkpoint ensemble can bring a gain of +0.5 BLEU solidly. At the same time, increasing the dimension of d_{ff} and activate more dropout are proved effective again. The biggest difference from Chinese → English is that diverse ensemble decoding improves the performance at most +1.33 BLEU when we integrate 10 models. However, increasing either the number of models or the diversity is helpful for ensemble decoding. As for reranking, although we only use four (EN, ZH, L2R, R2L) models as features due to time constraint. there is still +0.33 BLEU improvement obtained. At last, post-processing makes an more obvious effect for English → Chinese translation than Chinese → English.

5 Conclusion

We build a strong Transformer neural machine translation system as baseline, and have achieved comparable performance than the CWMT-17 best ensembled results. Beyond the baseline, we further improve the performance from four aspects, including architectural improvements, diverse ensemble decoding, reranking and post-processing. Experimental results show that our approach can improve 2.3–3.8 BLEU points consistently. Particularly, we find that increasing the number of models and the diversity of models is crucial for ensemble decoding.

Acknowledgments. This work was supported in part by the National Science Foundation of China (No. 61672138 and 61432013), the Fundamental Research Funds for the Central Universities.

References

1. Ba, J.L., Kiros, R., Hinton, G.E.: Layer normalization. CoRR abs/1607.06450 (2016). http://arxiv.org/abs/1607.06450
2. Chiang, D., Marton, Y., Resnik, P.: Online large-margin training of syntactic and structural translation features, pp. 224–233. Association for Computational Linguistics (2008)
3. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. ArXiv e-prints, May 2017
4. Hassan, H., et al.: Achieving human parity on automatic Chinese to English news translation. arXiv preprint arXiv:1803.05567 (2018)

[11] https://github.com/antho-rousseau/XenC.

5. He, K., Zhang, X., Ren, S., Sun, J.: Identity mappings in deep residual networks. In: Leibe, B., Matas, J., Sebe, N., Welling, M. (eds.) ECCV 2016. LNCS, vol. 9908, pp. 630–645. Springer, Cham (2016). https://doi.org/10.1007/978-3-319-46493-0_38
6. Heafield, K.: KenLM: faster and smaller language model queries. In: Proceedings of the EMNLP 2011 Sixth Workshop on Statistical Machine Translation, Edinburgh, Scotland, United Kingdom, pp. 187–197, July 2011. https://kheafield.com/papers/avenue/kenlm.pdf
7. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. arXiv preprint arXiv:1412.6980 (2014)
8. Ramachandran, P., Zoph, B., Le, Q.V.: Searching for activation functions (2018)
9. Rousseau, A.: Xenc: an open-source tool for data selection in natural language processing. Prague Bull. Math. Linguist. **100**, 73–82 (2013)
10. Sennrich, R., et al.: The University of Edinburgh's neural MT systems for WMT17. In: WMT 2017, pp. 389 (2017)
11. Sennrich, R., Haddow, B.: Linguistic input features improve neural machine translation. In: Proceedings of the First Conference on Machine Translation: Volume 1, Research Papers, pp. 83–91. Association for Computational Linguistics (2016)
12. 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)
13. 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, ACL 2016, Volume 1: Long Papers, 7–12 August 2016, Berlin, Germany (2016). http://aclweb.org/anthology/P/P16/P16-1162.pdf
14. Sennrich, R., Haddow, B., Birch, A.: Edinburgh neural machine translation systems for WMT 16. In: Proceedings of the First Conference on Machine Translation: Volume 2, Shared Task Papers, pp. 371–376. Association for Computational Linguistics (2016)
15. Shaw, P., Uszkoreit, J., Vaswani, A.: Self-attention with relative position representations. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers), vol. 2, pp. 464–468 (2018)
16. Tu, Z., Liu, Y., Shang, L., Liu, X., Li, H.: Neural machine translation with reconstruction. In: AAAI, pp. 3097–3103 (2017)
17. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems, pp. 6000–6010 (2017)
18. Wang, Y., et al.: Sogou neural machine translation systems for WMT17. In: Proceedings of the Second Conference on Machine Translation, pp. 410–415 (2017)
19. Wu, Y., et al.: Google's neural machine translation system: bridging the gap between human and machine translation. arXiv preprint arXiv:1609.08144 (2016)