A Comparable Study on Model Averaging, Ensembling and Reranking in NMT

Yuchen Liu[†], Long Zhou[†], Yining Wang[†], Yang Zhao[†],
Jiajun Zhang[†], Chengqing Zong[†‡]

[†]University of Chinese Academy of Sciences
National Laboratory of Pattern Recognition, CASIA
[‡]CAS Center for Excellence in Brain Science and Intelligence Technology
{yuchen.liu,long.zhou,yining.wang,yang.zhao,jjzhang,cqzong}@nlpr.ia.ac.cn

Abstract. Neural machine translation has become a benchmark method in machine translation. Many novel structures and methods have been proposed to improve the translation quality. However, it is difficult to train and turn parameters. In this paper, we focus on decoding techniques that boost translation performance by utilizing existing models. We address the problem from three aspects — parameter, word and sentence level, corresponding to checkpoint averaging, model ensembling and candidates reranking which all do not need to retrain the model. Experimental results have shown that the proposed decoding approaches can significantly improve the performance over baseline model.

1 Introduction

Neural machine translation(NMT) has significantly improved the quality of machine translation in recent years, which has shown promising results on multiple language pairs [1, 3, 6, 16, 19]. It builds upon a single and large neural network directly mapping source sentence to associated target sentence. Recently relying entirely on an attention mechanism, the transformer model introduced by Vaswani et al. [19] achieved state-of-the-art results for machine translation.

However, designing a novel and good translation model is a tough work. Training model is also time-consuming and occupies massive computing resources. For example, despite its remarkable success, transformer requires 3.5 days with 8 GPUs on training for a big model. Thus instead of modifying model structure, how to make effective use of the existing models to improve the translation performance is well worth considering.

In this paper, we investigate and practice decoding approaches to boost translation performance without training model again. As shown in Figure 1, we address the problem from three aspects — parameter, word and sentence level. First, we adapt the checkpoint averaging to get a more robust set of parameters, which can utilize multiple checkpoints saved at different timesteps in a single model. For word level, we introduce three ensembling strategies to boost word prediction, including checkpoint ensemble, independent ensemble and different ensemble. Finally, we observe that in validation set if each translation candidate

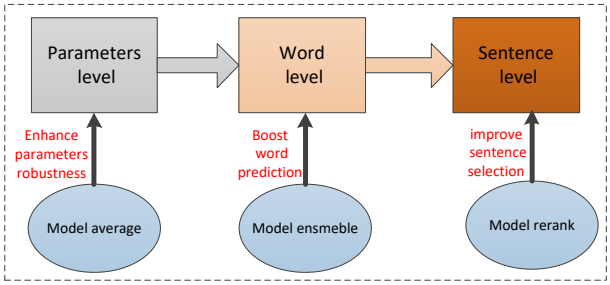

Fig. 1. Our framework for decoding approaches from three aspects: parameters, word and sentence level.

with top sentence-level BLEU score is selected, we can obtain over 18 BLEU points improvement compared with outputs by beam search algorithm. Inspired by this observation, we attempt to select better candidates by reranking techniques, which includes linear regression, pairwise-rank method and minimum bayes risk(MBR) decoding.

We conducted massive experiments on large-scale English-to-Chinese translation. Experimental results have shown that decoding approaches can obtain significant BLEU score improvements over state-of-the-art NMT baseline.

2 Neural Machine Translation

The decoding approaches we discuss can be applied to any neural machine translation model. Here we choose Transformer [19] as baseline model for later experiments. In this section, we will give a brief introduction of Transformer encoder-decoder framework.

Given a set of bilingual data $D = \{(X^{(i)}, Y^{i})\}_{i=1}^{N}$ where both X and Y are a sequence of tokens, the encoder maps a input sequence $X = (x_1, x_2, \cdots, x_n)$ to a sequence of continuous representations $z = (z_1, z_2, \cdots, z_n)$ whose size varies with respect to the source sentence length. The decoder generates an output sequence $Y = (y_1, y_2, \cdots, y_m)$ from the continuous representations. The encoder and decoder are trained jointly to maximize the conditional probability of target sequence given a source sequence:

$$P(Y|X;\theta) = \prod_{j=1}^{N} P(y_j|y_{<j}, x;\theta) \tag{1}$$

Transformer consists of N stacked encoder and decoder layers. Encoder layer consists of two blocks, which is self-attention block followed by a position-wise feed-forward block. Decoder layer has the same architecture as encoder layer except an extra encoder-decoder attention block. Residual connection and layer normalization are used around each block.

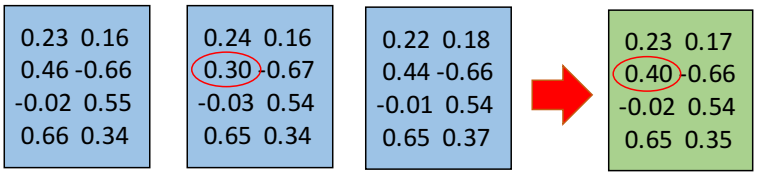

Fig. 2. Model averaging with the number of three

For self-attention and encoder-decoder attention, a multi-head attention block is used to obtain information from different representation subspaces at different positions. Each head corresponds to a scaled dot-product attention, which operates on a query Q, key K and a value V:

$$Attention(Q, K, V) = softmax(\frac{QK^T}{\sqrt{d_k}})V \tag{2}$$

where d_k is the dimension of the key.

For the sake of brevity, we refer readers to Vaswani et al. [19] for additional details regarding the architecture.

3 Methods Description

Many approaches have been proposed to generate better translation results in decoding [11, 10, 17, 9, 15]. In this section, we introduce three decoding techniques that boost translation performance by utilizing existing models, which includes checkpoint averaging, ensembling strategies and different methods of reranking.

3.1 Checkpoint Averaging

Checkpoint averaging is to average trainable parameters which are saved at last timesteps in a single model, when the model is near convergence. Since we use Stochastic Gradient Descent algorithm to optimize model, only a mini-batch of data are used during each step, causing that the parameters may over adapt to one mini-batch. We can get more robust parameters by checkpoint averaging. As illustrated in Figure 2, for example, the value in the red circle of second checkpoint is 0.30, distinct from the corresponding values of other two checkpoints, which means that it may be a noise value. After checkpoint averaging it turns to 0.40, thus a more proper value can be obtained. Vaswani [19] suggested to average the last 20 checkpoints saved every 10 minutes. Here we will make experiments on how many checkpoints to be chosen can obtain the best result.

3.2 Different Ensembling Strategies

Model ensembling is a method to integrate the probability distributions of multiple models before predicting next target word. It has been proved effective

in neural machine translation [11, 10, 22]. We apply three different ensembling strategies as following:

- **checkpoint ensemble** We use the checkpoints saved at different times in a single training model, which do not need to train several models from scratch. It is a cheap way to obtain an ensemble model.
- **independent ensemble** This strategy needs to train N models independently with the same architecture but different initialization ways. Combining N models in different initialization ways as an ensemble model can help to avoid local optimization and obtain better results.
- **different ensemble** Different ensemble trains N models with both different architectures and initialization ways, which is expensive but can yield better and more diverse result.

3.3 Different Reranking Strategies

Reranking is a long-term study in machine translation [13, 9, 17, 15]. In this paper, we apply three reranking strategies to investigate how to better select candidates, which includes linear regression, pairwise-rank method and MBR decoding.

Linear Regression

Linear regression is usually used to model the relationship between a dependent variable and one or more explanatory variables. We denotes sentence-level BLEU score as dependent variable, sentence-level features as explanatory variables. More specifically, we use beam search algorithm to obtain a list of candidate translations. Since validation set has reference, the sentence-level BLEU score of each sentence can be calculated. We then use target side right-to-left model, target-to-source model, n-gram language model, neural language model and SMT model trained by Moses[1] to calculate each sentence's feature scores. We first fit a linear regression model for validation set, then adapt it to test set. For test sentences we estimate the BLEU scores of each candidate, then select the sentence with top score as final output.

Pairwise Rank

In fact, we do not need to know the exact BLEU score of each sentence. The only thing we concern is the order of translation candidates. One of approaches is to reduce the ranking problem as a classification problem by using pairwise sampling [5]. However, ranks may be unreliable for machine translation where candidates sometimes can not be strongly distinguished between each other. To alleviate this problem, we assume the difference among top r translation candidates is not obvious. Thus, we regard the top r of the n-best candidates

[1] http://www.statmt.org/moses/

as good translations and the bottom k as bad translations, where $r + k \geq n$. Then a classification model is trained to split the good translations from the bad translations for each sentence. Besides, the following criteria are proposed:

- We assign a larger margin for the candidates whose ranks are far and a smaller margin for closer pairs. For example, $margin(\mathbf{e}_1, \mathbf{e}_{20}) > margin(\mathbf{e}_1, \mathbf{e}_{10})$
- For the same rank gap, the margin between a high rank and a low rank is larger than that between two low ranks. For example, $margin(\mathbf{e}_1, \mathbf{e}_{10}) > margin(\mathbf{e}_{21}, \mathbf{e}_{30})$. The reason is that the scoring function will be penalized if it can not separate former case, but not for the latter.

MBR Decoding

MBR decoding is a method to find a candidate with the least expected loss [17]. It measures the similarity of each candidate translation instead of the quality against reference. The Bayes risk of each candidate y is computed by:

$$R(y) = \sum_{y' \in E} \Delta(y, y') p(y'|x) \tag{3}$$

The term $\Delta(y, y')$ is calculated by $1 - BLEU(y, y')$, which denotes the discrepancy between candidate y and candidate y'. The term $p(y'|x)$ is the generating probability of each candidate given by a NMT model. The candidate with lowest Bayes risk means that it is similar to the most candidates in the evidence space.

4 Experiments

4.1 Dataset

We perform our experiments on corpus provided by AI Challenger — the English-Chinese Machine Translation track [2]. This corpus contains about 10 million parallel English-Chinese sentences which are collected from English learning websites and movie subtitles. In our experiments, we first filter the bilingual corpus according to the following criteria:

– Sentences which contain less than 3 words or more than 100 words are removed.
– We use fast_align toolkit to learn a word alignment of sentences pairs. Sentence pairs whose alignment ratio is lower than 0.3 are removed.
– We sort sentence pairs by perplexities and remove the bottom 5 percent sentence pairs.

After filtering we retain about 9 million pairs of training data. We also use monolingual sentences to train n-gram and neural language model for later use in reranking. The English side is preprocessed by tokenizer and lowercaser scripts

[2] https://challenger.ai/competition/translation

in Moses. The Chinese side is segmented by our in-house toolkit. We learn a BPE [12] model with 80k merge operations for both English side and Chinese side, and extract 83k and 78k subwords as source and target vocabularies. The evaluation metric is BLEU as calculated by the `multi-blue.perl` script.

4.2 Training Details

We adopt the Transformer model as our baseline model[3]. All hyper parameter settings are set the same as transformer_big_single_gpu if not specifically mentioned. The dimension of word embedding is set to 1024. The size of attention block is 1024 with 16 heads and the size of feed forward block is set to 4096.

We train the baseline model for a total of 300K steps with Adam optimizer [7] on three GPUs. We set the initial learning rate to 0.1 and apply decay method described in Vaswani et al. [19]. Dropout was applied on residual layer to avoid over-fitting, which is 0.1. At test time, we employ beam search with beam size 4. As for reranking, we set beam size as 50 to generate a list of candidates.

4.3 Results and Comparison on Different Approaches

In this section, we first separately compare different strategies in each approach. Then we report the final results by combining all these approaches. All of the first lines in every table are the baseline model without using any decoding approaches.

Effects on the number of checkpoints for averaging We first test how many checkpoints to be used can get better results. Figure 3 shows the effect on the number of checkpoints for averaging. We obverse that the BLEU scores by applying checkpoint averaging are all above baseline model. As the number of checkpoints increases, the set of parameters becomes more robust which brings more BLEU scores improvement. However too many checkpoints may contaminate parameters with scores decreasing. In our experiments, averaging 40 checkpoints saved in ten-minute interval obtains the best results. Compared with the time needed for training the whole model from scratch, checkpoint averaging only takes a few minutes before decoding, so it is a free way to boost.

Comparison of different ensembling strategies Table 1 shows the BLEU scores with different ensembling strategies. To be fair, all the three ensembling strategies use five models for ensembling. Specifically, checkpoint ensemble uses checkpoints saved at five different time during a single model training process; independent ensemble uses two models initialized by uniform algorithm, two by normal algorithm and one by orthogonal; different ensemble uses two 6-layer encoder-decoder structures with uniform and normal initializer, two 8-layer structures with 24 heads, 960 hidden size and 32 heads, 1024 hidden size

[3] https://github.com/tensorflow/tensor2tensor

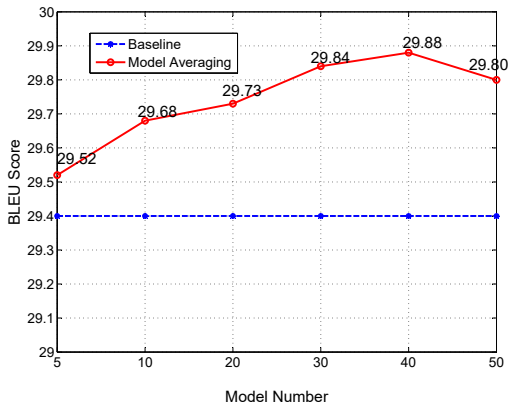

Fig. 3. Translation results (BLEU score) on checkpoint averaging.

respectively, and one 10-layer encoder with 6-layer decoder structure. Model ensembling can indeed boost translation result, while the improvements by three ensembling strategies are different. Among these strategies, different ensemble outperforms the other two strategies by about 1.2 and 0.6 points respectively. This strategy is expensive for training but more likely to yield better and more diverse results. This is consistent with the wisdom that there is no free lunch.

Table 1. Translation results (BLEU score) on different ensembling strategies.

System	**BLEU**
baseline	29.40
checkpoint ensemble	29.55(+0.15)
independent ensemble	30.18(+0.78)
different ensemble	30.76(+1.36)

Comparison of different reranking strategies In our experiment, we first generate 50-list of translation candidates by beam search algorithm, then calculate sentence-level BLEU for each candidate on validation set. To our surprise, we observe a remarkable gap between the output generated by beam search algorithm and the candidate selected by sentence-level BLEU. As illustrated in Figure 4, if we select each candidate with top 1 sentence-level BLEU, the corpus-level BLEU can reach 47.50; even if we select all the candidates with 20th sentence-level BLEU, we can also get a final BLEU of 29.77. However, the BLEU score of the output generated by beam search algorithm is only 29.40. Thus we think reranking has a great potential to further improve translation performance. We then attempt three reranking strategies. From Table 2, we observe

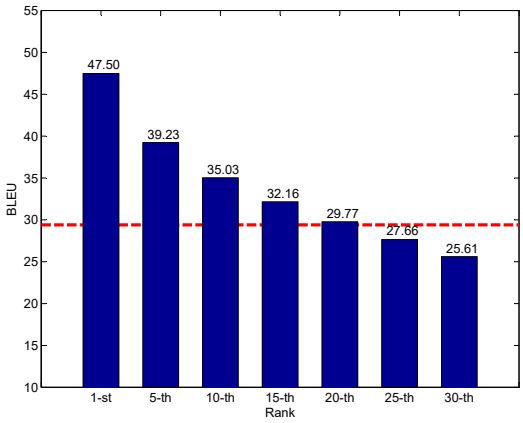

Fig. 4. Translation results (BLEU score) based on sentence-level BLEU.

that linear regression and MBR decoding can significantly improve the BLEU scores, while pairwise rank cannot well distinguish between the good translation candidate with the bad one. One possible reason is that the selected sentence-level features may not linearly separable. Though reranking by linear regression obtains an improvement, it heavily relies on the selected features which are expensive to extract. From this perspective, MBR decoding is the best strategy since all it needs is just a list of translation candidates. Although these strategies can improve the BLEU scores, it is still far away from the best result. We will further investigate this challenge in the future.

Table 2. Translation results (BLEU score) on different reranking strategies.

System	**BLEU**
baseline	29.40
linear regression	30.10(+0.70)
pairwise rank	29.88(+0.44)
MBR decoding	30.13(+0.73)

Results combining three approaches We choose the best strategy in each approach mentioned above, which are the averaging of 40 checkpoints, different ensemble and MBR decoding respectively. Combining all these approaches, we can obtain significant BLEU score improvements over baseline model. We list the BLEU scores of our proposed model in Table 3. Specifically, after checkpoints averaging, we can get an improvement of 0.54 BLEU points over baseline. We further obtain 0.99 BLEU points improvement with different ensembling strategies. By applying reranking method, another improvement of 0.69 BLEU scores can be achieved. It confirms the effectiveness of these decoding techniques.

Table 3. Translation results (BLEU score) for English-to-Chinese translation.

System	**BLEU**
baseline	29.40
+checkpoint averaging	29.94(+0.54)
+model ensembling	30.93(+1.53)
+reranking	31.62(+2.22)

5 Related Work

Recently many novel structures and methods have been proposed to improve the translation quality in neural machine translation. Most of the existing approaches focus on designing better models [18, 8, 19], augmenting data with large-scale monolingual corpus [2, 21], integrating SMT techniques [14, 4, 20]. In spite of modifying the model structure, our work mainly focuses on improving translation quality by using decoding techniques, which is somehow easier to implement.

Vaswani et al. [19] proposed to use checkpoint averaging method to obtain lower variance and more stable translation results. However, they did not explain how to choose checkpoints and how many checkpoints to be used can obtain better results. Sennrich et al. [11] first applied the method of checkpoint ensembling in WMT16, then they further tried independent ensembling in WMT17 [10], which achieved a significant improvement compared to the former strategy.

To get better final output, various reranking methods have been explored. Shen et al. [13] introduced two novel perceptron-inspired reranking algorithms that improve on the quality of machine translation. Kumar [17] presented MBR decoding for statistical machine translation aiming to minimize expected loss of translation errors under loss functions that measure translation performance. However, these methods are mainly applied to statistical machine translation. Here, we apply them to neural machine translation to explore how good the selected candidate can be by each reranking strategy.

6 Conclusion

In this work, we boost translation performance by making effective use of the existing models with three decoding techniques. Experiments have shown that these decoding techniques can obtain significant improvement over baseline. We point out that reranking has a great potential for improving translation performance, however, a wide gap between the oracle selection still exists. In the future, we will further investigate a better way to shrink this gap.

Acknowledgments

The research work descried in this paper has been supported by the National Key Research and Development Program of China under Grant No. 2016QY02D0303.

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate. In Proceedings of ICLR (2015)
2. Cheng, Y., Xu, W., He, Z., He, W., Wu, H., Sun, M., Liu, Y.: Semi-Supervised Learning for Neural Machine Translation. In Proceedings of ACL (2016)
3. Gehring, J., Auli, M., Grangier, D., Yarats, D., Dauphin, Y.N.: Convolutional sequence to sequence learning. arXiv preprint arXiv:1705.03122 (2017)
4. He, W., He, Z., Wu, H., Wang, H.: Improved neural machine translation with SMT features. In Proceedings of AAAI (2016)
5. Herbrich, R.: Large margin rank boundaries for ordinal regression. Advances in Large Margin Classifiers (2000)
6. Kalchbrenner, N., Blunsom, P.: Recurrent continuous translation models. In Proceedings of EMNLP (2013)
7. Kingma, D., Ba, J.: Adam: A method for stochastic optimization. In Proceedings of ICLR (2015)
8. Mi, H., Sankaran, B., Wang, Z., Ittycheriah, A.: A coverage embedding model for neural machine translation. arXiv preprint arXiv:1605.03148 (2016)
9. Och, F.J.: Minimum error rate training in statistical machine translation. In Proceedings of ACL (2003)
10. Sennrich, R., Birch, A., Currey, A., Germann, U., Haddow, B., Heafield, K., Barone, A.V.M., Williams, P.: The university of edinburgh's neural mt systems for wmt17. arXiv preprint arXiv:1708.00726 (2017)
11. Sennrich, R., Haddow, B., Birch, A.: Edinburgh neural machine translation systems for wmt16. arXiv preprint arXiv:1606.02891 (2016)
12. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In Proceedings of ACL (2016)
13. Shen, L., Sarkar, A., Och, F.J.: Discriminative reranking for machine translation. In Proceedings of HLT-NAACL (2004)
14. Shen, S., Cheng, Y., He, Z., He, W., Wu, H., Sun, M., Liu, Y.: Minimum risk training for neural machine translation (2015)
15. Shu, R., Nakayama, H.: Later-stage Minimum Bayes-Risk Decoding for Neural Machine Translation. arXiv preprint arXiv:1704.03169 (2017)
16. Sutskever, I., Vinyals, O., Le, Q.V.: Sequence to sequence learning with neural networks. In Proceedings of NIPS (2014)
17. Tromble, R.W., Kumar, S., Och, F., Macherey, W.: Minimum bayes-risk decoding for statistical machine translation. In Proceedings of HLT-NAACL (2004)
18. Tu, Z., Lu, Z., Liu, Y., Liu, X., Li, H.: Modeling coverage for neural machine translation. In Proceedings of ACL (2016)
19. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L., Polosukhin, I.: Attention is all you need. Advances in Neural Information Processing Systems (2017)
20. Wang, X., Lu, Z., Tu, Z., Li, H., Xiong, D., Zhang, M.: Neural machine translation advised by statistical machine translation. In Proceedings of AAAI (2017)
21. Zhang, J., Zong, C.: Exploiting source-side monolingual data in neural machine translation. In Proceedings of EMNLP (2016)
22. Zhou, L., Hu, W., Zhang, J., Zong, C.: Neural system combination for machine translation. In Proceedings of ACL (2017)