Checkpoint Reranking: An Approach To Select Better Hypothesis For Neural Machine Translation Systems

Vinay Pandramish
International Institute of
Information Technology,
Hyderabad
pandramish.vinay@research.iiit.ac.in

Dipti Misra Sharma
International Institute of
Information Technology,
Hyderabad
dipti@iiit.ac.in

Abstract

In this paper, we propose a method of reranking the outputs of Neural Machine Translation (NMT) systems. After the decoding process, we select a few last iteration outputs in the training process as the N-best list. After training a Neural Machine Translation (NMT) baseline system, it has been observed that these iteration outputs have an oracle score higher than baseline up to 1.01 BLEU points compared to the last iteration of the trained system.We come up with a ranking mechanism by solely focusing on the decoder's ability to generate distinct tokens and without the usage of any language model or data. With this method, we achieved a translation improvement up to +0.16 BLEU points over baseline.We also evaluate our approach by applying the coverage penalty to the training process.In cases of moderate coverage penalty, the oracle scores are higher than the final iteration up to +0.99 BLEU points, and our algorithm gives an improvement up to +0.17 BLEU points.With excessive penalty, there is a decrease in translation quality compared to the baseline system. Still, an increase in oracle scores up to +1.30 is observed with the re-ranking algorithm giving an improvement up to +0.15 BLEU points is found in case of excessive penalty.The proposed re-ranking method is a generic one and can be extended to other language pairs as well.

1 Introduction

Neural Machine Translation(NMT) has brought excellent results in the field of Machine TranslationSutskever et al. (2014); Bahdanau et al. (2014); Cho et al. (2014) due to generation of high-quality translations for different language pairs. Yet even higher quality can be achieved by combining multiple models by techniques like ensembles Hansen and Salamon (1990) and reranking Shen et al. (2004). Our work deals with how Neural Machine Translation (NMT) can achieve better results explicitly with reranking methods.

Neural Machine Translation has an encoder-decoder architecture that is jointly trained to maximize the probability of target given source sentences. It first encodes the source sentence into a single vector, and the decoder predicts it.With the Attention Mechanism, it tries to apply weights to the input sentence at each time step. Recent approaches like the transformer model Vaswani et al. (2017) have achieved the state-of-the-art results for Machine Translation.

Neural Machine Translation (NMT) however, leads to over-translation and under-translation as it tends to ignore the past alignment information, and it is effectively tackled by introducing a coverage vector Tu et al. (2016). Other approaches such as Mi et al. (2016a) and Mi et al. (2016b) too solve the coverage problem in NMT. Without the coverage vector, it could result in a decrease in translation quality.

We propose a method that selects a better hypothesis giving high importance to distinct words generated from decoder without the usage of any language model or data.After applying the proposed reranking method, an overall improvement in translation quality is observed as compared to the baseline system.

The rest of the paper is organized as follows; Section 2 discusses the work related to re-utilizing existing models for Machine Translation. Section 3 describes our approach for Checkpoint based Reranking. In Section 4, we present our Reranking Algorithm. In Section 5, we demonstrate all of our Experiments along with the results obtained, and finally, the paper is concluded in Section 6 with future directions.

Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop, pages 286–291
July 5 - July 10, 2020. ©2020 Association for Computational Linguistics

2 Related Work

The work of Imamura and Sumita (2017) explains the concepts of reranking and ensembling in detail. It introduces a method of bidirectional reranking in which it combines the hypothesis from l2r and r2l decoding following the works of Liu et al. (2016), which proposes an agreement model to solve unbalanced outputs of recurrent neural networks. Marie and Fujita (2018) has introduced a reranking system that uses a smorgasbord of informative features in tasks where PBSMT and NMT produce translations of different quality.

The work by Shen et al. (2004) shows how to apply perceptron-like reranking algorithms to improve the overall translation quality, and Olteanu et al. (2006) shows the usage of Language Models (LMs) for reranking on hypotheses generated by phrase-based Statistical Machine Translation systems. Wang et al. (2007) has shown linguistically motivated and computationally efficient structured language models for reranking in SMT systems.

The concept of Checkpoint ensembles is introduced by Sennrich et al. (2016) and was later on improvised to independent ensembling Sennrich et al. (2017). Vaswani et al. (2017) included a checkpoint averaging method for their model. Liu et al. (2018) has focused on decoding techniques that utilize existing models at parameter, word, and sentence level corresponding to checkpoint averaging, model ensembling, and candidate reranking and found that all of these improve the translation quality without retraining the model.

3 Checkpoint Based Reranking

In our approach, the iteration outputs are selected as the N-best list. It implies for the last K iterations; we have the corresponding K-best list for a sentence. We take our Oracle scores as the one that is having the largest BLEU Score Papineni et al. (2002) on the test reference hypothesis from this K-best list. After obtaining the oracle scores from this K-best list, we observe that this score is larger than the baseline system, and it indicates that there is scope for further improvement of translation quality. So we propose a reranking method that improves the translation quality over the baseline system without any language model or data.

We try to focus on the nature of translations that the decoder generates with and without coverage penalty. In the initial step, we keep track of the number of distinct words in the generated hypothesis, and the later ones we keep track of words that have repeated more than once.A higher score is given for sentences having a higher number of Distinct Tokens (D) and lower scores for those having more number of repetitive words (F).

For each sentence in the N-best list, these scores are sorted, and the sentence having the highest score is selected. This process is repeated for the entire test set, and the ones that are having the top most scores are chosen as the reranked output, as shown in Section 4.

4 Reranking Procedure

```
Algorithm 1 Method
  Input: Translated Target Language Sentences
$H$ = ( $h(n-k)$...,$h(n)$ ) at last $k$ epochs for given
sentence
  Output: Sentence having highest number of distinct words and lowest repetitive
words
  for each sentence $h_j$ in $H$ do
    if $h_j \leftarrow$ (w$_1$,$w_2$,$w_3$...$w_l$) then
      $D \rightarrow DISTINCT$((w$_1$,$w_2$,$w_3$...$w_l$))
      $F \rightarrow FREQ$(w$_1$)$\times FREQ$(w$_2$)...
    $score_j$ $\rightarrow$D/F
    end if
    return sentence with highest $score_j$
  end for=0
```

For a sentence, FREQ is the count of each word; DISTINCT is the total count of unique words. For each hypothesis in the K-best list we divide DISTINCT with FREQ and select the highest scorer.

5 Experiments and Results

5.1 DataSet

We used ILCI Jha (2010) corpus, which has eleven language pairs from which we chose Telugu and Hindi as our parallel data during the training process. The entire corpus is manually cleaned to remove the misalignments. Table 1 shows the split ratio of sentences followed in the process.

Data	Size
Training	45000
Validation	4000
Test	990

Table 1: Corpus Division

5.2 Experiments

We adopt the Keras implementation Álvaro Peris and Casacuberta (2018) for our experiments.We use a two-layer encoder-decoder model with 500-dimensional source and target embeddings and 500 units in each of the layers. The encoder layers are LSTM Hochreiter and Schmidhuber (1997) and decoder are ConditionalLSTM with Bahdanu's attention Bahdanau et al. (2014) and the optimizer used is Adam Kingma and Ba (2014) and the model is trained for 15 iterations with a batch size of 512 sentences. The rest of the parameters in the configuration file were set to their default values. We evaluate with coverage penalty and the absence of it for our experiments.

The hypotheses are collected for the last k=3, 5, 7 during decoding. We evaluate the generated hypotheses with BLEU Papineni et al. (2002) for our experiments.

5.3 Results

Hypothesis	BLEU
Checkpoint-1	0.62
Checkpoint-2	3.55
Checkpoint-3	8.83
Checkpoint-4	13.53
Checkpoint-5	17.01
Checkpoint-6	19.20
Checkpoint-7	20.72
Checkpoint-8	21.09
Checkpoint-9	21.38
Checkpoint-10	21.87
Checkpoint-11	22.39
Checkpoint-12	22.37
Checkpoint-13	22.57
Checkpoint-14	22.71
Checkpoint-15	22.92

Table 2: BLEU Scores with Baseline System

The scores obtained after each iteration are shown in Table 2. After this, we apply our proposed reranking method to the last few iteration outputs, which are selected as the N-best list. The proposed reranking method leads to an overall improvement of translation quality by +0.07, +0.15, +0.16 BLEU score compared to the baseline with oracle improvements up to +0.55, +0.90, +1.01 on the three systems. The scores obtained for each of them are shown in Tables 3, 4, 5.

System	BLEU
Baseline	22.92
Reranking	22.99 **(+0.07)**
Oracle	23.47 **(+0.55)**

Table 3: Last 3 Iterations

System	BLEU
Baseline	22.92
Reranking	23.07 **(+0.15)**
Oracle	23.82 **(+0.90)**

Table 4: Last 5 Iterations

System	BLEU
Baseline	22.92
Reranking	23.08 **(+0.16)**
Oracle	23.93 **(+1.01)**

Table 5: Last 7 Iterations

5.4 With Coverage Penalty

We also evaluate our work by adding coverage penalty Wu et al. (2016) in the training process to ensure that this algorithm works when both the under translations and over translations are addressed adequately. All the hyperparameters are kept the same as the baseline system except for the coverage penalty.

Hypothesis	0.1 penalty
Checkpoint-1	1.22
Checkpoint-2	5.59
Checkpoint-3	11.92
Checkpoint-4	16.59
Checkpoint-5	19.18
Checkpoint-6	20.51
Checkpoint-7	21.26
Checkpoint-8	21.40
Checkpoint-9	21.80
Checkpoint-10	21.72
Checkpoint-11	22.27
Checkpoint-12	22.57
Checkpoint-13	23.11
Checkpoint-14	22.93
Checkpoint-15	23.35

Table 6: BLEU Scores With 0.1 Coverage Penalty

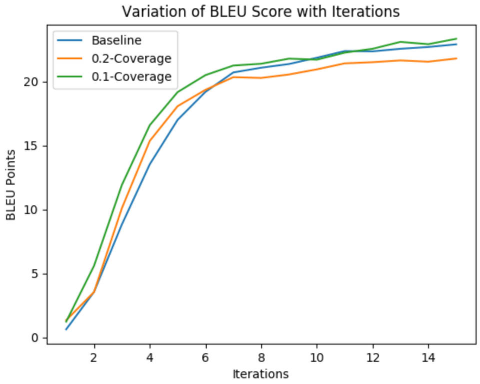

Figure 1: Comparison with Baseline System

System	0.1 penalty
Baseline	23.35
Reranking	23.40 **(+0.05)**
Oracle	23.86 **(+0.51)**

Table 7: Last 3 Iterations with 0.1 coverage penalty

System	0.1 penalty
Baseline	23.35
Reranking	23.50 **(+0.15)**
Oracle	24.17 **(+0.82)**

Table 8: Last 5 Iterations with 0.1 coverage penalty

System	0.1 penalty
Baseline	23.35
Reranking	23.52 **(+0.17)**
Oracle	24.34 **(+0.99)**

Table 9: Last 7 Iterations with 0.1 coverage penalty

From Tables 7, 8, 9 it can be inferred that there is an improvement of +0.05, +0.15, +0.17 and oracle improvements up to +0.51, +0.82, +0.99 for 0.1 coverage penalty.

With excess coverage penalty, there is a decline in translation quality compared to the baseline system without coverage penalty, as shown in Tables 2 and 10. Still, the proposed method gives an increase of +0.12, +0.15, +0.15 over baseline with oracle improvements up to +0.91, +1.30, +1.30 for the last 3, 5 and 7 checkpoints respectively as shown in Tables 11, 12, 13.

One can also observe that the improvements and the oracle scores increase correspondingly with the size of the N-best list.The variation with the baseline can be obtained as shown in Figure 1.

Hypothesis	0.2 penalty
Checkpoint-1	1.33
Checkpoint-2	3.54
Checkpoint-3	10.10
Checkpoint-4	15.36
Checkpoint-5	18.08
Checkpoint-6	19.36
Checkpoint-7	20.35
Checkpoint-8	20.29
Checkpoint-9	20.56
Checkpoint-10	20.96
Checkpoint-11	21.43
Checkpoint-12	21.52
Checkpoint-13	21.66
Checkpoint-14	21.56
Checkpoint-15	21.81

Table 10: BLEU Scores With 0.2 Coverage Penalty

System	0.2 penalty
Baseline	21.81
Reranking	21.93 **(+0.12)**
Oracle	22.72 **(+0.91)**

Table 11: Last 3 Iterations with 0.2 coverage penalty

System	0.2 penalty
Baseline	21.81
Reranking	21.96 **(+0.15)**
Oracle	23.11 **(+1.30)**

Table 12: Last 5 Iterations with 0.2 coverage penalty

System	0.2 penalty
Baseline	21.81
Reranking	21.96 **(+0.15)**
Oracle	23.11 **(+1.30)**

Table 13: Last 7 Iterations with 0.2 coverage penalty

6 Conclusions and Future Work

In this paper, we introduce a method of selecting an N-best list for NMT systems and propose a way of reranking to the generated hypotheses from the system. We observe that our approach is giving better results over the baseline model by following the proposed reranking method and is also evaluated with the coverage penalty.

One can investigate our approach with varying beam sizes and analyzing the effect of length penalty Wu et al. (2016) and comparing it with methods such as Yang et al. (2018). We also look forward to coming up with better reranking ways that are closer to the oracle scores and investigate the efficacy of the approach in low-resourced data conditions.

Language models are used for getting the likelihood of sentences and is a widely used concept for reranking hypotheses. Introducing Language Models during reranking could establish a tradeoff between perplexity and the scores to the hypotheses generated. We also plan to explore the work by Çaglar Gülçehre et al. (2017) and Çaglar Gülçehre et al. (2015) that introduces language models into the existing neural architecture with methods such as Shallow Fusion and Deep Fusion. It is another promising area to be looked upon for reranking.

Acknowledgments

The authors would like to thank some of the Anonymous reviewers for providing critical suggestions. We acknowledge Kenton Murray for the post-acceptance mentoring, Manish Shrivastava for the guidance, Vandan Mujadia for reviewing the drafts and Grammarly for their generous donation towards drafting the final manuscript.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *CoRR*, abs/1409.0473.

Kyunghyun Cho, Bart van Merrienboer, Çaglar Gülçehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *EMNLP*.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *ArXiv*, abs/1503.03535.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, and Yoshua Bengio. 2017. On integrating a language model into neural machine translation. *Computer Speech Language*, 45:137–148.

Lars Kai Hansen and Peter Salamon. 1990. Neural network ensembles. *IEEE Trans. Pattern Anal. Mach. Intell.*, 12:993–1001.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural Computation*, 9:1735–1780.

Kenji Imamura and Eiichiro Sumita. 2017. Ensemble and reranking: Using multiple models in the nict-2 neural machine translation system at wat2017. In *WAT@IJCNLP*.

Girish Nath Jha. 2010. The tdil program and the indian langauge corpora intitiative (ilci). In *Proceedings of the Seventh International Conference on Language Resources and Evaluation (LREC'10)*, Valletta, Malta. European Language Resources Association (ELRA).

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *CoRR*, abs/1412.6980.

Lemao Liu, Masao Utiyama, Andrew M. Finch, and Eiichiro Sumita. 2016. Agreement on target-bidirectional neural machine translation. In *HLT-NAACL*.

Yuchen Liu, Long Zhou, Yining Wang, Yang Zhao, Jiajun Zhang, and Chengqing Zong. 2018. A comparable study on model averaging, ensembling and reranking in nmt. In *NLPCC*.

Benjamin Marie and Atsushi Fujita. 2018. A smorgasbord of features to combine phrase-based and neural machine translation. In *AMTA*.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016a. A coverage embedding model for neural machine translation. *ArXiv*, abs/1605.03148.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016b. Supervised attentions for neural machine translation. In *EMNLP*.

Marian Olteanu, Pasin Suriyentrakorn, and Dan I. Moldovan. 2006. Language models and reranking for machine translation. In *WMT@HLT-NAACL*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Álvaro Peris and Francisco Casacuberta. 2018. NMT-Keras: a Very Flexible Toolkit with a Focus on Interactive NMT and Online Learning. *The Prague Bulletin of Mathematical Linguistics*, 111:113–124.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017. The university of edinburgh's neural mt systems for wmt17. In *WMT*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Edinburgh neural machine translation systems for wmt 16. In *WMT*.

Libin Shen, Anoop Sarkar, and Franz Josef Och. 2004. Discriminative reranking for machine translation. In *HLT-NAACL*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS*.

Zhaopeng Tu, Zhengdong Lu, Yang P. Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *ACL*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *NIPS*.

Weiqi Wang, Andreas Stolcke, and Jing Zheng. 2007. Reranking machine translation hypotheses with structured and web-based language models. *2007 IEEE Workshop on Automatic Speech Recognition Understanding (ASRU)*, pages 159–164.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V. Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, Jeff Klingner, Apurva Shah, Melvin Johnson, Xiaobing Liu, Lukasz Kaiser, Stephan Gouws, Yoshikiyo Kato, Taku Kudo, Hideto Kazawa, Keith Stevens, George Kurian, Nishant Patil, Wei Wang, Cliff Young, Jason Smith, Jason Riesa, Alex Rudnick, Oriol Vinyals, Gregory S. Corrado, Macduff Hughes, and Jeffrey Dean. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *ArXiv*, abs/1609.08144.

Yilin Yang, Liang Huang, and Mingbo Ma. 2018. Breaking the beam search curse: A study of (re-)scoring methods and stopping criteria for neural machine translation. In *EMNLP*.