A Study of Improving BLEU Using RL Loss in NMT

Tianchun Huang

August 2019

1 Introduction

In NMT, there is always a gap between MLE loss and BLEU scores. Hence, reinforcement learning methods are implemented to integrate BLEU scores into the loss function. This document concludes the experimental results of training with different RL loss and different parameter settings. Experiments are conducted using IWSLT14 German-English (de-en). The model is pre-trained with MLE loss until convergence, and then RL loss is used to continue training. Experiments show that final BLEU scores can be improved with the help of RL loss.

2 Baseline Settings

Baseline architecture is transformer model in "Attention is All You Need." Code can be found on https://github.com/TianchunH97/fairseq-rl. Most of the training settings are same as in fairseq examples except that weight decay is 0.0 and loss is MLE without label smoothing. The BLEU score for the baseline model is 34.26.

3 Implementation of RL Loss

Two versions of RL loss are used for this study. One successfully improves final BLEU scores while one does not work. In both versions, GLEU score is used as reward in RL instead of BLEU score. GLEU score is similar to BLEU score except that the minimum of precision and recall is used instead of precision in BLEU. Hence, the result of GLEU computation is symmetric about reference and translation.

3.1 Version 1 RL

Each source sentence x_i in a sample is used as input and generates a translation sentence $\hat{y_i}$, then reward $r(y_i, \hat{y_i})$ which is the GLEU score can be calculated.

The total loss for a batch of samples can be calculated as:

$$L_1 = \sum_{i=1}^{N}(-logP(\hat{y_i}|x_i) \cdot r(y_i, \hat{y_i}))$$

3.2 Version 2 RL

In this version, each source sentence x_i is used to generate multiple translation sentences $\hat{y_i^1}$, ..., $\hat{y_i^M}$, and for each $\hat{y_i^j}$ there is a score $r(y_i, \hat{y_i^j})$. For each sample, there are multiple scores and the average of them is calculated as

$$\bar{r_i} = \frac{1}{M}\sum_{j=1}^{M} r(y_i, \hat{y_i^j})$$

The total loss for a batch of samples is then calculated as:

$$L_2 = \sum_{i=1}^{N}\sum_{j=1}^{M}(-logP(\hat{y_i^j}|x_i) \cdot (r(y_i, \hat{y_i^j}) - \bar{r_i}))$$

In this study, M is set to be 5. In implementation, some changes may be applied to the computation of gleu score, the generation of translation sentence, and the training setting. But the overall loss computation formula will not change.

3.3 Final Loss

The final loss is a combination of the RL loss mentioned above and the MLE loss, which can be written as:

$$L_{total} = \alpha L_{MLE} + (1 - \alpha)L_{RL},$$

where α is a hyperparameter that can be adjusted.

4 Results

4.1 Version 1

Results have shown that this implementation of RL loss does not work, but will make final BLEU drop dramatically. The larger the learning rate is, the faster the BLEU scores drop. Figure 1 shows the Epoch vs. BLEU curve when learning rate is set to be $2e-7$, which is very small.

Also we can notice that changing the weight of mle and rl loss doesn't prevent BLEU scores from falling. But the higher the weight of mle loss is, the slower the result drops. Hence, we can conclude that this version of RL loss cannot work.

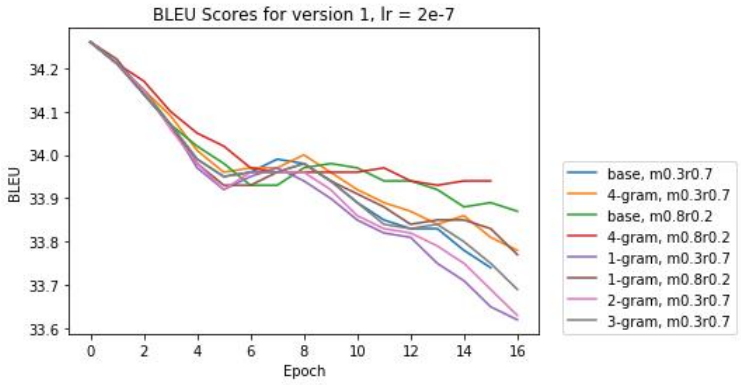

Figure 1: Results for v1 Loss

4.2 Version 2

The version 2 loss appears to be much more effective than version 1. After numbers of experiments the highest BLEU score reaches 35.09, which increases by 0.83 compared to baseline score. With this loss, the effectiveness of many other factors in the process are studied, and will be discussed in the following parts.

4.3 Generation of Translation Sentence

There are two ways of generating translation sentences from the source sentence. The first is beam search and the second is sampling. In beam search, we choose words with highest probabilities at each step, while in sampling we sample words from probability distributions at each step. Sampling can make the hypotheses more diverse while beam search is more accurate. The results show that in the process of generating candidate sentences for RL loss, sampling is much more effective than beam search. With beam search, the BLEU can reach around 34.6, but with sampling the BLEU can reach above 35.0. (Figure 2)

4.4 GLEU Computation in RL Loss

The original 4-gram GLEU uses the average of log of scores of each gram as exponent, i.e. the weight for each gram is [1/4, 1/4, 1/4, 1/4]. Besides this original setting, other weight distributions for grams are used in this study.

4.4.1 Distribution of Gram Weights

The first attempt is to put all weights on one kind of gram. That is to set the weights for each gram to be [1, 0, 0, 0], [0, 1, 0, 0], [0, 0, 1, 0], [0, 0, 0, 1]. In this way, only one length of gram is used in the process of GLEU computation. The

```
[34.26, 34.41, 34.55, 34.65, 34.61]
[34.26, 34.4, 34.52, 34.57, 34.59]
[34.26, 34.4, 34.54, 34.6, 34.57, 34.53, 34.55]
[34.26, 34.4, 34.51, 34.59, 34.6]
[34.26, 34.55, 34.76, 34.88, 34.92, 34.93, 35.0, 34.98, 34.98, 35.02, 35.03, 35.04, 35.06]
[34.26, 34.56, 34.77, 34.93, 34.85, 34.97, 34.99, 35.06, 35.02, 35.06, 35.09, 35.07, 35.06]
[34.26, 34.52, 34.63, 34.57, 34.58, 34.54, 34.48]
[34.26, 34.53, 34.57, 34.59, 34.54, 34.51, 34.44]
[34.26, 34.4, 34.52, 34.59, 34.56, 34.57, 34.6]
```

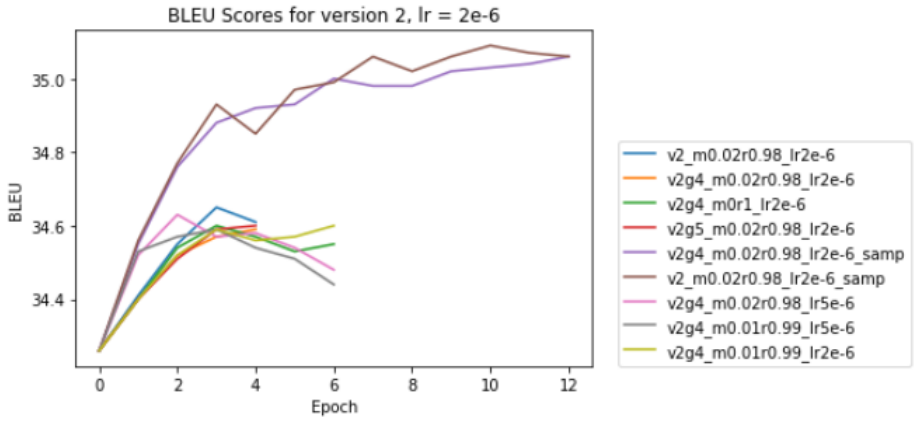

Figure 2: Beam vs sampling results comparison

results show that while all other settings are the same, using only one length of gram doesn't lead to obvious change in BLEU score. (Figure 3) Hence, we choose not to change the weight distribution for different length of grams.

Besides experiments with gram length 1,2,3,4, 5-gram is also attempted. That is to set the weights to be [0, 0, 0, 0, 1]. The result is that 5-gram will have a negative effect on final BLEU scores. (Figure 4)

4.4.2 Max order for GLEU Computation

The second attempt is to change the max order of GLEU. For $max\_order = N$, the weight distribution for different length of grams is then [1/N, ..., 1/N]. In this experiment we use $max\_order = 2, 3, 4, 5$. For $max\_order = 5$, a modified version is added. In some cases both reference sentence and translation sentence may be shorter than 5 words. In original version, 5-gram score for this sentence will be 0 and it will count in the final score. In the modified version, for sentences shorter than 5 words, the max order will be set to be the maximum length of reference sentence and translation sentence. The results (Figure 5) show that the change of max order can slightly affect the final results. The higher the max order is, the higher the BLEU. However, there is only little difference. Hence, using the default max order 4 is enough. For this group of experiments, sampling is applied, learning rate is $1e-5$, and no MLE weight is

```
[34.26, 34.81, 34.98, 35.04, 35.06, 35.05, 35.02, 35.01, 34.98, 35.01]
[34.26, 34.9, 34.95, 34.99, 34.95, 34.91, 35.0, 34.98, 34.98, 34.96, 35.01]
[34.26, 34.87, 34.95, 35.0, 35.03, 35.07, 35.06, 35.02, 35.03, 35.03, 35.0]
[34.26, 34.81, 34.97, 35.0, 35.0, 35.04, 35.04, 35.02, 35.03, 35.0, 35.0]
```

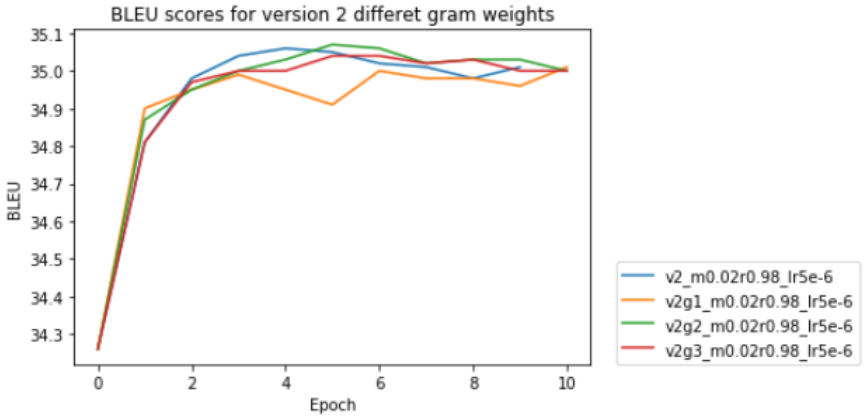

Figure 3: Results for v2 (sampling) with different gram choice

```
[34.26, 34.41, 34.55, 34.65]
[34.26, 34.53, 34.48, 34.45, 34.48]
[34.26, 33.14, 33.08]
[34.26, 34.4, 34.51, 34.59, 34.6]
[34.26, 34.46, 34.48, 34.27, 34.22]
```

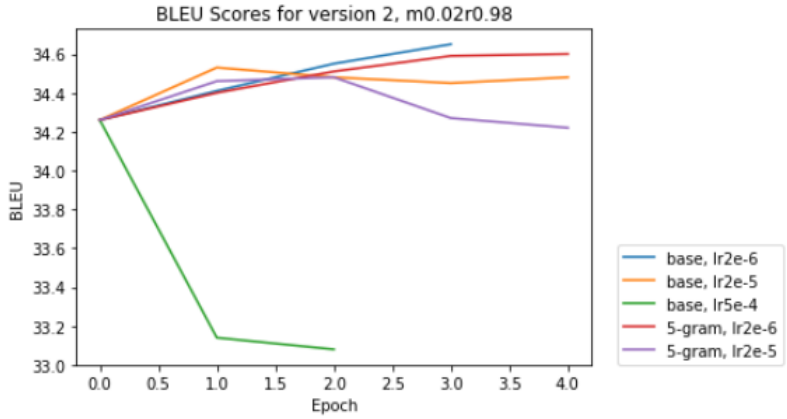

Figure 4: Results for v2 (beam) with 5-gram and different lr

added. Additionally, attention dropout and relu dropout are both set to be 0.1, while in other experiments they are not added.

```
[34.26, 34.9, 34.95, 34.89, 34.99, 34.92, 34.94, 34.81, 34.71]
[34.26, 34.9, 34.94, 35.0, 34.98, 34.91, 34.92, 34.92, 34.82, 34.69]
[34.26, 34.95, 35.0, 35.02, 34.99, 34.94, 34.89, 34.9, 34.78, 34.7]
[34.26, 34.95, 35.0, 35.03, 34.97, 34.97, 34.88, 34.92, 34.91, 34.73]
[34.26, 34.92, 35.01, 35.03, 34.94, 34.92, 34.89, 34.94, 34.88, 34.77]
```

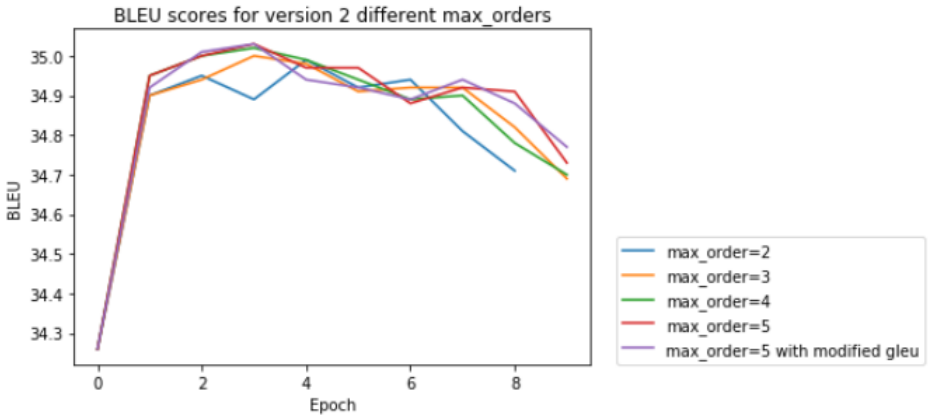

Figure 5: Results for v2 (sampling) with different max order

4.4.3 Learning Rate

Learning rate for training must be smaller than that used for MLE loss. The "lr" in figures represent the maximum learning rate, and the learning rate scheduler does not change compared to the MLE version. As shown in Figure 4, if the maximum learning rate is not changed, the BLEU score will dramatically drop from the beginning. Hence, we need lower the maximum learning rate. Results have shown that both $lr = 2e-6$ (Figure 2) and $lr = 1e-5$ (Figure 6) lead to the best BLEU 35.09, while larger learning rate reaches the best score faster. But with $lr = 2e-7$, BLEU score cannot go as high because the learning rate is too small. (Figure 7) Therefore, maximum learning rate between $2e-6$ and $1e-5$ is recommended.

4.4.4 MLE Weight

MLE weight is added to make the training stable. As shown in Figure 6 above, without MLE weight the BLEU will become unstable and drop fast after it reaches its peak. Also, the maximum BLEU score will be slightly lower without mle weight. Since MLE loss is much larger than RL loss, mle weight should be small compared to rl weight. Results show that when $mle\_weight = 0.2$ and $rl\_weight = 0.8$, the maximum BLEU score cannot reach 35.0 (Figure 6), while both $mle\_weight = 0.02$ (Figure 2) and $mle\_weight = 0.1$ (Figure 6) lead to best BLEU score 35.09. Therefore, mle weight between 0.02 and 0.1 is recommended and should be adjusted according to different conditions.

```
[34.26, 34.82, 34.98, 35.03, 35.04, 34.94, 34.92, 34.93, 34.85, 34.73]
[34.26, 34.99, 35.02, 35.03, 34.73, 34.62, 33.96]
[34.26, 34.95, 34.95, 35.07, 35.07, 35.09, 35.05, 35.03, 35.04, 35.01]
[34.26, 34.95, 35.05, 35.06, 35.02, 35.01, 34.97, 34.68]
[34.26, 34.84, 34.83, 34.91, 34.91, 34.94, 34.97, 34.94, 34.94]
[34.26, 34.95, 35.05, 35.06, 35.05, 35.05, 35.01, 34.86]
```

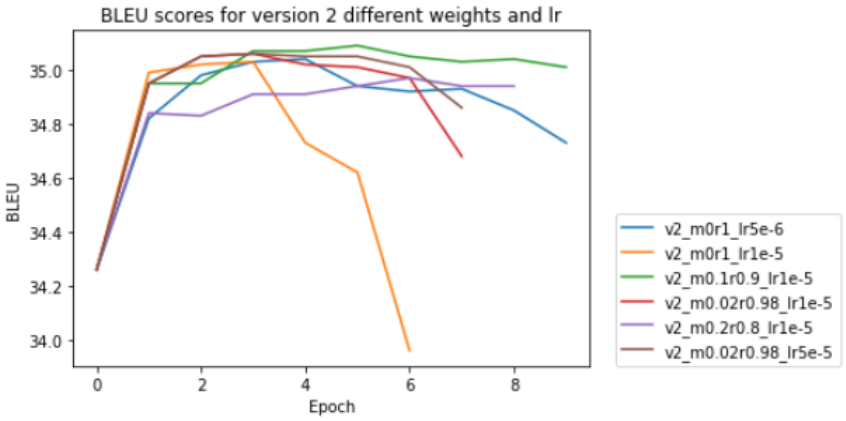

Figure 6: Results for v2 (sampling) with different mle weight and lr

5 Conclusion

This document studies the effect of RL loss in NMT from implementation to training settings with IWSLT14 DE-EN as dataset and transformer as model. The results show that RL is able to improve the BLEU score from 34.26 which is the baseline score to 35.09. In implementation, the results show that generating multiple hypos for each source sentence in a sample and using the difference between GLEU score and average GLEU per sample as reward is an effective way to make the RL loss work. Besides, generating hypos by sampling is much more effective than beam search. Other factors such as weight of MLE loss and learning rate also affects the final result, while changing the max order or the grams used in computing GLEU score in reward can hardly make a difference.

```
[34.26, 34.29, 34.3, 34.3, 34.31, 34.32, 34.34, 34.33]
[34.26, 34.24, 34.23, 34.23, 34.2, 34.23, 34.2, 34.21]
[34.26, 34.29, 34.29, 34.3, 34.31, 34.32, 34.32, 34.34]
[34.26, 34.32, 34.32, 34.35, 34.36, 34.38, 34.38, 34.42]
[34.26, 34.31, 34.33]
[34.26, 34.33, 34.35]
[34.26, 34.31, 34.33]
[34.26, 34.32, 34.33]
[34.26, 34.31, 34.34]
[34.26, 34.32, 34.34]
```

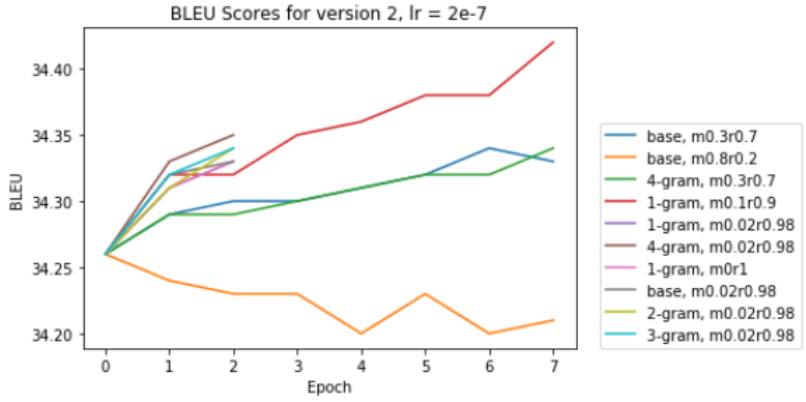

Figure 7: Results for v2 (beam) with lr=2e-7