Multi-Reward based Reinforcement Learning for Neural Machine Translation

Shuo Sun[†] , **Hongxu Hou**[*], **Nier Wu**[†] , **Ziyue Guo**[†] ,**Chaowei Zhang**[†]

[†,*]College of Computer Science-college of Software, Inner Mongolia University,China
[*]cshhx@imu.edu.cn
[†]{sunshuo07,wunier04,guoziyue08,zhangchaowei08}@126.com

Abstract

Reinforcement learning (RL) has made remarkable progress in neural machine translation (NMT). However, it exists the problems with uneven sampling distribution, sparse rewards and high variance in training phase. Therefore, we propose a multi-reward reinforcement learning training strategy to decouple action selection and value estimation. Meanwhile, our method combines with language model rewards to jointly optimize model parameters. In addition, we add Gumbel noise in sampling to obtain more effective semantic information. To verify the robustness of our method, we not only conducted experiments on large corpora, but also performed on low-resource languages. Experimental results show that our work is superior to the baselines in WMT14 English-German, LDC2014 Chinese-English and CWMT2018 Mongolian-Chinese tasks, which fully certificates the effectiveness of our method.

1 Introduction

Neural machine translation (NMT) (Bahdanau et al., 2015; Wu et al., 2018a; Yang et al., 2018) has drawn universal attention without the demand of numerous manual work. In training phase, generic NMT models employ maximum likelihood estimation (MLE) (Harris and Mandelbaum, 1985), which is the token-level objective function. However, it is inconsistent with sequence-level evaluation metrics such as BLEU (Papineni et al., 2002). Reinforcement learning (RL) are leveraged for sequence generation tasks including NMT to optimize sequence-level objectives, such as Actor-Critic (Bahdanau et al., 2017) and Minimum Risk Training (MRT) (Shen et al., 2016). In machine translation community, Wu el at. (Wu et al., 2018a) provide the first comprehensive study of different aspects of RL training, they set a single reward to mitigate the inconsistency, and combine MLE with RL to stabilize the training process. Nevertheless, NMT based on reinforcement learning (RL) is unable to guarantee that the machine-translated sentences are as natural, sufficient and accurate as reference. To obtain smoother translation results, generative adversarial network (GAN) and deep reinforcement learning (DRL) (Wu et al., 2018b) are employed to NMT. And (Yang et al., 2018) utilizes sentence-level BLEU Q as a reinforcement target based on the work of (Wu et al., 2018b) to enhance the capability of the generator.

Although this nova machine translation learning paradigm based on GAN and DRL reveals excellent manifestation, there are still some limitations: (1) when calculating rewards, the overestimation of Q value will give rise to a suboptimal strategy update. (2) during training phase, it exists the problems with uneven sampling distribution, sparse rewards and high variance. What's more, the generator uses Monte Carlo to simulate the entire sentence, but it usually requires more calculation steps, resulting in too many parameters. (3) traditional NMT usually utilizes deterministic algorithms such as Beam Search or Greedy Decoding when predicting the next token. These methods lacks randomness, which may cause the potential best solution to be discarded.

In this paper, we propose some measures to address the above problems. Foremost, we adopt a novel multi-reward reinforcement learning method. That is, we weighted sum the actual reward of the discriminator, the language model reward and the sentence BLEU to obtain the total reward. Among them,

Proceedings of the 19th China National Conference on Computational Linguistics, pages 985-994, Hainan, China, October 31 - November 1, 2020.
(c) Technical Committee on Computational Linguistics, Chinese Information Processing Society of China

we adopt reward shaping to alleviate the sparse reward when calculating sentence rewards. Nextly, our method employ Temporal-Difference Learning (TD) (Sutton, 1988) to simulate the entire sentence. It effectively speeds up training and relieves the problem of error accumulation. Finally, we adopt Gumbel-Top-K Stochastic Beam Search (Kool et al., 2019) to predict the next token. The method trains model more efficiently by adding the noise obeying Gumbel distribution to control random sampled noise. Experiments on the datasets of the English-German, Chinese-English and Mongolian-Chinese translation tasks reveal our approach outperforms the best published results. In summary, we mainly made the following contributions:

- It is the first time that duel reward has been applied to neural machine translation. This method is applicable to arbitrary end-to-end NMT system.
- Our generator to optimize reward by using Gumbel-Top-K Stochastic Beam Search to sample different samples and Temporal-Difference Learning to simulate sentences.
- In English-German and Chinese-English translation tasks, we tested two different NMT models: RNNSearch and Transformer. Experimental results reveal that our proposed method performs well.

2 Background & Related Work

Common NMT models are based on an encoder decoder architecture. The encoder reads and encodes the source language sequence $X = (x_1, ..., x_n)$ into the context vector representation, and the decoder generates the corresponding target language sequence $\hat{Y} = (\hat{y}_1, ..., \hat{y}_m)$. Given H training sentence pairs $\left\{x^i, y^i\right\}_{i=1}^{H}$, at each timestep t, NMT is trained by maximum likelihood estimation(MLE) and generates the target words $\hat{y}_t$ by maximum the probability of translation conditioned on the source sentence X. The training goal is to maximize:

$$L_{MLE} = \sum_{i=1}^{H} logp(\hat{y}^i|x^i) = \sum_{i=1}^{H}\sum_{t=1}^{m} logp(\hat{y}_t^i|\hat{y}_1^i...\hat{y}_{t-1}^i, x^i) \tag{1}$$

where m is the length of sentence $\hat{y}^i$.

According (Williams, 1992), reinforcement learning enables NMT to optimize evaluation during training and usually estimates the overall expectation by sampling $\hat{y}$ with policy $p\left(\hat{y}|x\right)$. The training objective of RL is to maximize the expected reward:

$$L_{RL} = \sum_{i=1}^{H} R\left(\hat{y}^i, y^i\right), \hat{y}^i \sim p\left(\hat{y}|x^i\right), \forall_i \in [H]. \tag{2}$$

where $R\left(\hat{y}, y\right)$ is the final reward calculated by BLEU after generating the complete sentence $\hat{y}$. To increase stationarity, we combine the two simple linearly:

$$L_{COM} = \mu \times L_{MLE} + (1 - \mu) \times L_{RL} \tag{3}$$

where μ is the hyperparameter to control the balance between MLE and RL. L_{COM} is the strategy to stabilize RL training progress.

(Yang et al., 2018) proposed the BLEU reinforced conditional sequence generative adversarial net (BR-CSGAN) on the basis of reinforcement learning. The specific process is that generator G generates the target sentence based on the source sentence, and discriminator D detects whether the given sentence is groundtruth. During training status, G attempts to deceive discriminator D into believing that the generated sentence is groundtruth. The D strives to improve its anti-spoofing ability to distinguish machine-translated sentences from groundtruth. When G and D reach the Nash balance, the training results achieve the optimal state, and utilize BLEU to guide the learning of the generator.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 985-994, Hainan, China, October 31 - November 1, 2020.

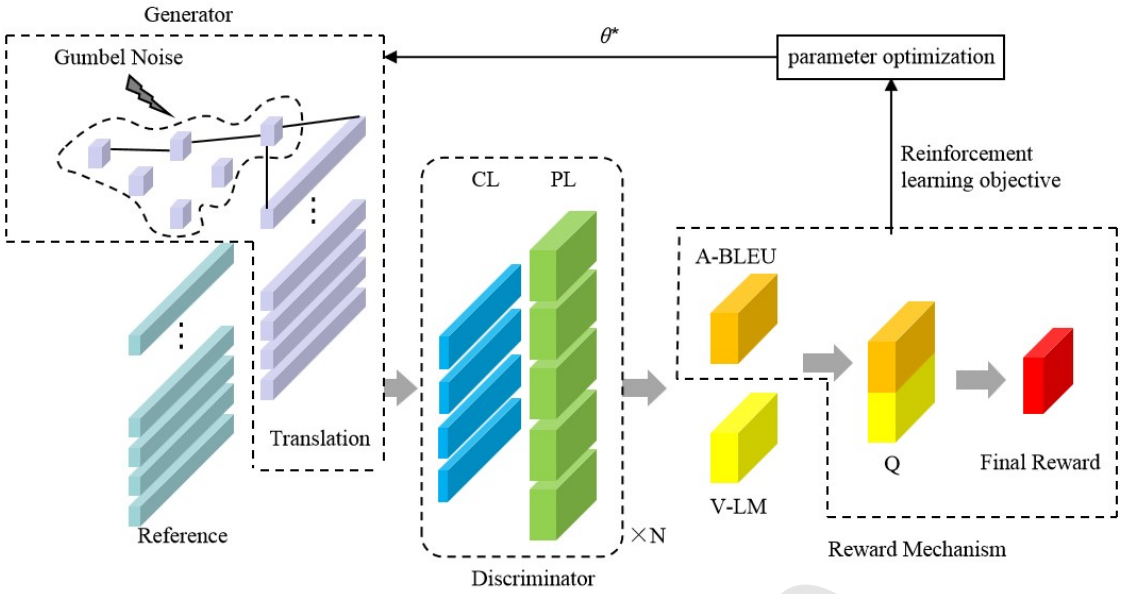

Figure 1: The Illustration of the proposed multi-reward generative adversarial net(referred to as MR-GAN). The discriminator D is trained over the reference sentence pairs translated by the human and the generated sentence pairs (sampled with Gumbel noise) by G. Extract feature information via convolution (CL) and pooling (PL) operations, and adopt global features to jointly calculate rewards. Lastly, the generator G is trained by policy gradient where the final reward R is provided by D, V and A.

3 Approach

In this section, we describe the multi-reward of reinforcement learning evaluation paradigm based on GAN model. The overall architecture is shown in Figure 1. We introduce the generator G, discriminator D, sampling with Gumbel-Top-K Stochastic Beam Search, calculating final reward and training the entire model in detail.

3.1 GAN-based NMT

The Generative Adversarial Net comprises of two adversarial sub models, a generator and a discriminator. The generator G is similar to the NMT model. Based on the source language sentence X, G aims to generate a target sentence $\hat{Y}$ which is indistinguishable from the reference Y. We take two different architectures for the generator, the traditional RNNSearch (Bahdanau et al., 2015) and the state-of-the-art Transformer (Vaswani et al., 2017).

We utilize CNN (Yin et al., 2016) that performs better in classification tasks to construct the discriminator D. It aims to identify machine-generated sentences from a set of sentences containing machine translation $\hat{Y}$ and reference Y. To be specific, the generator's output $\hat{Y}$ or reference Y is spliced with the source language sentence X to form a two-dimensional matrix, and the similarity between Y and X is measured by a convolution network. The optimization goal of the discriminator is minimize the coss-entropy loss of the binary classification:

$$L = -\left[a * log(p) + (1-a) * log(1-p)\right], p = \delta\left(V\left[r_X; r_{\hat{Y}}\right]\right) \tag{4}$$

where p is the probability that the target-language sentence is being real. r_X is the sentence representation of source language, which consists of extracting different features through different numbers of kernels with different window sizes. Similarly, $r_{\hat{Y}}$ is the target language sentence representation extracted from the target matrix $\hat{Y}_{1:T} = \hat{y}_1; \hat{y}_2; ...; \hat{y}_T$. V indicates the matrix which is used to merge r_X and $r_{\hat{Y}}$ into a low dimensional vector space. δ denote as the logistic function and a is a variable, which is correctly 1, otherwise 0.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 985-994, Hainan, China, October 31 - November 1, 2020.

3.2 Sampling

General NMT adopt Beam Search to generate the next token to reduce search space and speed up decoding. However, in many training methods such as RL or MRT, it is necessary to randomly collect multiple different samples from the model to calculate the sentence-level loss when decoding, but traditional methods can only produce similar results and loss of randomness. For this purpose, this work adopts an efficient and stable sampling method based on Gumbel-Top-K Stochastic Beam Search (Kool et al., 2019) to predict next token.

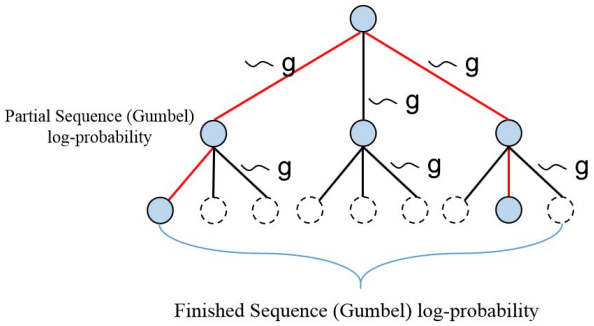

Figure 2: Gumbel-Top-K Stochastic Beam Search. $\sim g$ indicate add Gumbel noise when sampling.

This algorithm uses Top-Down sampling (Maddison et al., 2014) and performs Beam Search on the log probability of random perturbations. The structure is shown in Figure 2 with $k = 2$. We first perturb the log probability of the root node, then perturb and correct the log probability of all candidate sequences, and only keep the two nodes with the highest log perturbation probability to expand. Finally we get two samples with more randomness as well as each sample is subject to the original distribution.

Specifically, for a category distribution I with N categories $I \sim Categorical\left(\frac{exp\phi_i}{\sum_{j\in N} exp\phi_i}\right)$, where ϕ_i is the log-probability of the i-th category and $i \in N$. If we take the logarithm of each category of I and add the noise g that obeys the Gumbel distribution, then take the Top-K from this slightly disturbed sentence with the Top-K probability (ie.largest K categories after logarithmic calculation). The equations are as follows:

$$G \sim Gumbel\left(\phi\right) = \phi - log\left(-logU\right) = Gumbel\left(0\right) + \phi \tag{5}$$

$$I_{1,\ldots,k} = argtopK_{i\in N} Gumbel\left(\phi_i\right) \tag{6}$$

where $U \sim Uniform\left(0,1\right)$ and $G_i \sim Gumbel\left(0\right)$. It can be guaranteed that these K categories are subject to I and are different simultaneously, meanwhile the noise is controlled by the Gumbel distribution. With this method, we can train the model more efficiently and alleviate the problems of overtranslation and undertranslation in NMT.

3.3 Multi-Reward of Reinforcement Learning

As shown in Figure 1. Distinct with (Yang et al., 2018), which directly apply smoothed sentence-level BLEU as the specific objective Q for the generator. We aim to alleviate the overestimation of the reward, meanwhile, consider the fluency of machine translation and loyalty to the real translation. Therefore, our method is inspired by (Wang et al., 2016), given the generated sentence $\hat{Y}_{1:t}$ and the reference Y, the objective Q calculates a reward $Q(\hat{Y}_{1:t}, Y)$, which measures the fluency and loyalty of the generated sentence $\hat{Y}_{1:t}$, the equation is computed as:

$$Q(\hat{Y}_{1:t}, Y) = \lambda V\left(\hat{Y}_{1:t}\right) + (1-\lambda)\, A\left(\hat{Y}_{1:t}, Y\right) \tag{7}$$

where we set the independently generated language model reward as the value function $V\left(\hat{Y}_{1:t}\right)$, and the sentence reward as the advantage function $A\left(\hat{Y}_{1:t}, Y\right)$. λ is a hyper-parameter.

Value function For the sake of receiving smoother translation, we utilize the language model scores to participate in the calculation of rewards in reinforcement learning so that the NMT can consider the contextual and positional information of the corpus when translating. $V\left(\hat{Y}_{1:t}\right)$ represents the fluency score of sequence $\hat{y}_{1:t}$ including the current word, which guide the NMT to translate a sufficiently accurate and smooth result.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 985-994, Hainan, China, October 31 - November 1, 2020.

Typical language models have problems such as zero probability or statistical inadequacies. Good and Turing (Gale and Sampson, 1995) proposed a new probabilistic formula to ease the "unsmoothness" problem. As shown in Figure 3. They solved the zero-probability problem by down-regulating the frequency of words below the threshold and giving the out-of-vocabulary (OOV) a small non-zero value, where the sum of the down-regulated frequencies equals the probability of the OOV. The equation for 3-gram is as follow:

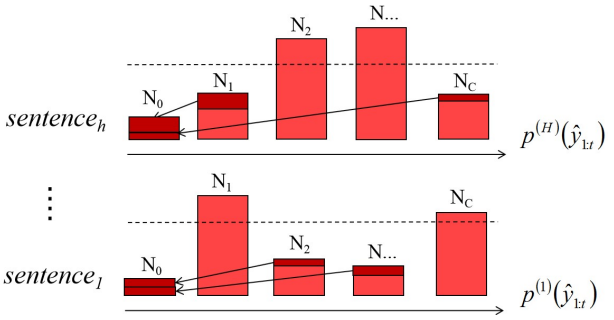

Figure 3: Good-Turing smoothing algorithm. "...." represent threshold, decrease the frequency of words which number of occurrences is lower than the threshold, and give the sum of the resulting frequencies to the words that do not appear.

$$P_{GT}(\omega_i|\omega_{i-2},\omega_{i-1}) = \begin{cases} f(\omega_i|\omega_{i-2},\omega_{i-1}), c(\omega_{i-2},\omega_{i-1},\omega_i) \geq U \\ f_{gt}(\omega_i|\omega_{i-2},\omega_{i-1}), 0 < c(\omega_{i-2},\omega_{i-1},\omega_i) < U \\ Q(\omega_{i-2},\omega_{i-1}) \cdot P(\omega_i|\omega_{i-1}), otherwise \end{cases} \tag{8}$$

where we set $U = 9$, which is a threshold, and the function $f_{gt}(.)$ represents the relative frequency after Good-Turing estimation. Therefore, the probability is normalized to get $V\left(\hat{Y}_{1:t}\right)$, the equation is as follow:

$$V\left(\hat{Y}_{1:t}\right) = \sum_{s=1}^{S} c_s P_{GT}^s \tag{9}$$

where c_s represents the number of words that occur s times, and P_{GT}^s represents the probability of s occurrences obtained from Good-Turing smooth algorithm.

Advantage function From Equation(2), the reward $R\left(\hat{y}, y\right)$ is only obtained after generate a complete sentence $\hat{y}$, it indicate only one reward is available for all actions(sample $\hat{y}_1 ... \hat{y}_T$). Consequently, RL training is inefficient due to the sparsity of rewards, and the model updates each token in the training sentence with the same reward without distinction. Following (Wu et al., 2018a), we employ reward shaping to overcome the shortcoming. The current reward with reward shaping is defined as:

$$r_t\left(\hat{y}_t, y\right) = R\left(\hat{y}_{1...t}, y\right) - R\left(\hat{y}_{1...t-1}, y\right) \tag{10}$$

where $R\left(\hat{y}_{1...t}, y\right)$ is defined as the BLEU score of $\hat{y}_{1...t}$ respect to y. Reinforce algorithm has high variance because it use a single sample $\hat{y}$ to estimate the expectation. To improve the stability of the algorithm, we add an estimate of the average reward at each step t, and then subtract it from future cumulative reward. The cumulative reward are obtained from (11):

$$R\left(\hat{y}, y\right) = \sum_{i=1}^{m} r_t\left(\hat{y}, y\right), R\left(\hat{y}, y\right) - \hat{r}_t \tag{11}$$

Combined with reward shaping, at each step t the Advantage function is computed as:

$$A\left(\hat{Y}_{1:t}, Y\right) = \sum_{T=t}^{m} r_T\left(\hat{y}_T, y\right) - \hat{r}_t \tag{12}$$

Final Reward According to the objective of the generator model (policy) $G_{\theta^*}(\hat{y}_t|\hat{Y}_{1:t-1})$ (Yu et al., 2017), to estimate $R_{D,V,A}^{G_{\theta^*}}$,which is the action-value function of a target sentence. Following Equation

(6), we consider the actual estimated probability of the discriminator D, the language model scores V and the sentence reward A as the final reward that update and optimize the generator G:

$$R_{D,V,A}^{G_{\theta^*}}\left(\hat{Y}_{1:T-1}, X, \hat{y}_T, Y\right) = \alpha\left(D\left(X, \hat{Y}_{1:T}\right) - b\left(X, \hat{Y}_{1:T}\right)\right) + \beta V\left(\hat{Y}_{1:t}\right) + \gamma A\left(\hat{Y}_{1:t}, Y\right) \quad (13)$$

where $b\left(X, \hat{Y}\right)$ represents the baseline value for reducing the variance estimation of rewards. We set $b\left(X, \hat{Y}\right) = 0.5$ based on experience. $\hat{Y}_{1:T}$ represents the generated target sentence and Y indicates the reference. α,β,γ are hyper-parameters.

However, D only provides a reward value for a entire generated target sequence. If $\hat{Y}_{1:T}$ is not the completed target sequence, the value of $D\left(X, \hat{Y}_{1:T}\right)$ is meaningless. Therefore, we cannot obtain the action-value of the intermediate state directly. Due to the large variance and parameters of Monte Carlo search, our work utilize Temporal-Difference (TD)[0] to sample the last $T-t$ tokens, it does not stop until the end of the sentence is sampled or the sampled sentence attains the maximum length. We implement the H TD emulation process as:

$$(\hat{Y}_{1:T_1}^1, ..., \hat{Y}_{1:T_H}^H) = TD^{G_{\theta^*}}\left(\left(\hat{Y}_{1:t}, X\right), H\right) \quad (14)$$

where $\left(\hat{Y}_{1:t}, X\right) = (\hat{y}_1 ... \hat{y}_t, X)$ is the current state, and $\hat{Y}_{t+1:T_H}^H$ is sampling based on G_{θ^*}. The discriminator rewards the sampled sentences separately and the discriminator output is calculated as the average of the H rewards. Therefore, for a target sentence of length T, we calculate the reward for $\hat{y}_t$ as:

$$R_{D,V,A}^{G_{\theta^*}}\left(\hat{Y}_{1:t-1}, X, \hat{y}_T, Y\right) = \begin{cases} \frac{1}{H}\sum_{j=1}^{H} \alpha\left(D\left(X, \hat{Y}_{1:T_h}^h\right) - b\left(X, \hat{Y}_{1:T_h}^h\right)\right) + \beta V\left(\hat{Y}_{1:T_h}\right) + \gamma A\left(\hat{Y}_{1:T_h}, Y\right) & t < T \\ \alpha D((X, \hat{Y}_{1:t}) - b(X, \hat{Y}_{1:t})) + \beta V\left(\hat{Y}_{1:t}\right) + \gamma A\left(\hat{Y}_{1:t}, Y\right) & t = T \end{cases} \quad (15)$$

3.4 Training

The training goal is to train G from the initial state to achieve maximum expectations end rewards. The objective equation is as follows:

$$J(\theta^*) = \sum_{\hat{Y}_{1:T}} G_{\theta^*}\left(\hat{Y}_{1:T}|X\right) \cdot R_{D,V,A}^{G_{\theta^*}}\left(\hat{Y}_{1:T-1}, X, \hat{y}_T, Y\right) \quad (16)$$

where R is Equation (16). Using sentence overall rewards to dynamically update the discriminator and then the generator.

$$min - E_{X,\hat{Y}\in P_{data}}\left[logD\left(X, \hat{Y}\right)\right] - E_{X,\hat{Y}\in G}\left[log\left(1 - D\left(X, \hat{Y}\right)\right)\right] \quad (17)$$

After completing the above operations, we adopt gradient descent to retrain the generator:

$$\bigtriangledown J(\theta^*) = \frac{1}{T}\sum_{t=1}^{T} E_{\hat{y}_t\in G_{\theta^*}}\left[R_{D,V,A}^{G_{\theta^*}}\left(\hat{Y}_{1:t-1}, X, \hat{y}_T, Y\right) \cdot \bigtriangledown_{\theta^*} logp\left(\hat{y}_t|\hat{Y}_{1:t-1}, X\right)\right] \quad (18)$$

4 Experiment and Analysis

We evaluate Chinese-English (Zh-En), English-German (En-De) and Mongolian-Chinese(Mo-Zh) tasks to verify the effectiveness of our MR-GAN.

[0]Monte Carlo search is updated after sampled the complete sentence $\hat{y}$. It causes too many parameters and slower update speed when sentence length is longer. Temporal-Difference (TD) algorithm is an iterative way of calculating value function, which is updated once per sampling, accelerates the convergence speed and reduces variance.

Model	Zh-En				En-De
	MT14	MT15	MT16	AVE	Newstest2014
Representative end-to-end NMT systems					
RNNSearch (Bahdanau et al., 2015)	33.76	34.08	33.98	33.94	21.20
RNNSearch+BR-CSGAN (Yang et al., 2018)	35.47	35.71	36.14	35.77	22.89
Transformer (Vaswani et al., 2017)	41.82	41.67	41.92	41.80	27.30
Transformer+RL (Wu et al., 2018a)	41.96	42.13	41.97	42.02	27.25
Transformer+BR-CSGAN (Yang et al., 2018)	42.46	42.54	42.83	42.61	27.92
Our work					
RNNSearch+MR-GAN	36.93	37.04	36.89	**36.95**	**24.61**
Transformer+MR-GAN	43.23	43.66	43.98	**43.62**	**28.69**

Table 1: BLEU scores of different NMT systems on Chinese-English and English-German.

4.1 Datasets and preprocessing

For En-De translation, we conduct our experiments on WMT14 En-De dataset, which contains 4.5 million bilingual pairs. Sentences are encoded using byte-pair encoding(BPE) (Sennrich et al., 2016). Newstest2012/2013 are chosen for development set, Newstest2014 as the test set. For the Zh-En translation, LDC2014 corpus as training set with a total of 1.6 million bilingual pairs. Both the source and target sentences are encoded with BPE. MT2013 is used as a development set and MT2014/2015/2016 as a test set. For Mo-Zh translation, the dataset adopts 261643 sentence pair Mongolian-Chinese bilingual aligned corpus provided by CWMT2018, we utilize 220000 sentence pairs as training set, 20822 as validation set, and the rest as test set. We perform word segmentation processing on the Chinese. On the Mongolian end, due to its own natural separator, so we encode it with BPE.

4.2 Setting

For Transformer-Big, following (Vaswani et al., 2017), we set $dropout = 0.1$ and set the dimension of the word embedding as 1024. We employ the Gumbel-Top-K Stochastic Beam Search to sample the target token with beam size $K = 4$. A single model obtained by averaging the last 20 checkpoints and we use adaptive methods to adjust the learning rate. For RNNSearch (Bahdanau et al., 2015), it is an RNN-based encoder decoder framework with attention mechanism. We set the hidden layer nodes and word embedding dimensions of the encoder and decoder to 512 and $dropout = 0$. The learning rate and checkpoint settings are consistent with Transformer-Big.

For D, CNN consists of one input layer, three convolution + pool layer pairs, one MLP layer and softmax layer. When the model is down-sampling, we use a 3×3 convolution window to perform convolution calculations on the internal corpus, and the output size is 2×2 pooling window. In addition, we set the feature map and MLP hidden layer size as 20. The word embedding dimension and the number of nodes are consistent with G.

Considering the computational complexity of model and the hardware environment of experiment, we adopt ELMO[1] (Peters et al., 2018) to construct and train the language model, which fully consider contextual information in semantic learning. Furthermore, we adopt BLEU (Papineni et al., 2002) to evaluate these tasks. All models are implemented in T2T tool and trained on two Titan XP GPUs. We stop training when the model does not improve on the tenth evaluation of the development set.

4.3 The pre-training of model

When the generator and discriminator achieve the synchronization and coordination effect, the performance of the model will be optimal. Therefore, we need to pre-train the model. The first step is pre-train

[1] http://allennlp.org/elmo/
ELMO, which fully consider contextual information has shown certain potential in semantic learning. It has strong modeling capabilities, meanwhile, the parameters and complexity are relatively small, which is convenient for model construction and training.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 985-994, Hainan, China, October 31 - November 1, 2020.

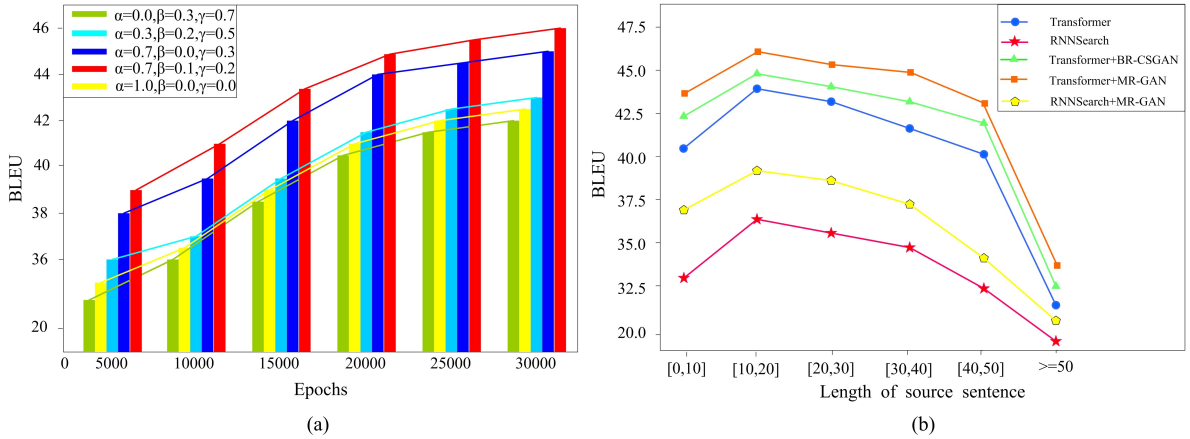

Figure 4: (a):Training line charts with different hyper-parameters weights. (b):BLEU scores on test set of LDC2014 Zh→En over different length of source sentences.

the generator G on bilingual training set until the best translation performance is achieved and we employ the traditional maximum likelihood estimation during the process. Then, generate the sentences(machine translations) by using the generator to decode the training data. The next step is pre-trian the discriminator on the combination of true bilingual data and machine transaltion data until the classification accuracy achieves at ξ. Finally, according to the study of Yang et al. (Yang et al., 2018), the method of jointly training the generator and discriminator and using the policy gradient to train the generator will lead to unstableness. Therefore, following (Yang et al., 2018), we adopt the teacher forcing approach to solve this problem. The parameter setting is exactly similar to (Yang et al., 2018), but the difference is that we employ the Temporal-Difference instead of Monte Carlo.

4.4 Main results and Analysis

For RNNSearch, it is optimized with the mini-batch of 64 examples. For Transformer, each training batch contains a set of sentence pairs contains approximately 25000 source tokens and 25000 target tokens. Table 1 shows the comparison between existing NMT system and our work. It can be seen that on Transformer, our approach outperforms the beat performance model and achieves improvement up to +1.01 BLEU points averagely on Chinese-English test sets an +0.77 BLEU points on English-German test set. It is profit from the novel method we have adopted to calculate rewards. Compared with traditional reinforcement learning, the scope of reward calculation is wider and making translation results more accurate and fluent. Furthermore, our method adds Gumbel noise when sampling, which makes the sampling more random and alleviates the problem of overtranslation and undertranslation. Experiments on the RNNSearch model shows the same trends, our approach still achieves 36.95 and 24.61 BLEU points on Chinese-English and English-German translations respectively.

4.5 Effect of Hyper-parameters and sentence length

We conduct a set of typical experiments using Transformer on the Chinese-English task to verify the influence of hyper-parameters (Equation 15) on experimental results. As shown in Figure $4(a)$, the worst result is obtained when $\alpha = 0$. The effect of the model continues to improve as the value of α increases. In the case of $\alpha = 0.7$, $\beta = 0.1$. and $\gamma = 0.2$, it achieves the best performance in several groups of experiments, and when $\alpha = 0.7$, $\beta = 0$. and $\gamma = 0.3$, the effect is not satisfactory. It indicates that the integration of language model rewards to evaluate the fluency of translation can effectively improve the quality of the model. When $\alpha = 1.0$, the effect is very poor but better than $\alpha = 0$, which explains that the multiple rewards proposed in this paper are effectively.

To verify the performance of this method on long sentences, following (Bahdanau et al., 2015), we divided the development set data and test set data of the Chinese-English task according to the sentence length. Figure $4(b)$ shows the BLEU scores for different sentence lengths. No matter on RNNSearch

Proceedings of the 19th China National Conference on Computational Linguistics, pages 985-994, Hainan, China, October 31 - November 1, 2020.

ID	V-LM	A-BLEU	G-N	ZH-EN	EN-DE
1	×	×	×	42.61	27.92
2	×	√	×	42.82	27.97
3	√	√	×	43.07	28.23
4	√	×	√	43.21	28.42
5	√	√	√	**43.62**	**28.69**

(a)

Model	MO-ZH	Promote
Transformer	34.56	—
Transformer+RL	35.02	0.46
Transformer+BR-CSGAN	35.63	1.07
Transformer+Our method	**36.82**	**2.26**

(b)

Figure 5: (a): Ablation study on Zh→En and En→De tasks. "○" means utilize this module and "×" means not utilize. "G-N" indicate sample with Gumbel noise and Line 1 represent the result of BR-CSGAN. (b): BLEU scores on test set of CWMT2018 MO→ZH over different length of source sentences.

or Transformer, compared with baseline and the best performing BR-CSGANS (Yang et al., 2018), our work have outstanding behaviors continuously. It is due to our method not only calculates the single-step reward, but also adds a smoothing restriction, which makes our method perform better on both long and short sentences.

4.6 Ablation Study

Figure $5(a)$ shows the results of ablation study. Line 1 represent the result of BR-CSGAN and line 2 represent that reward shaping is used to calculate BLEU on the basis of BR-CSGAN. It is clear that language model reward plays a critical role since removing it impairs translation performance (line 3). As shown in line 4, sampling with Gumbel noise is also an essential part of our approach. The sentence reward with each token is also shown to be benefit for improving performance (line 2) but seem to have relatively smaller contributions than the above two parts.

4.7 Result of Mongolian-Chinese

To verify the robustness of the proposed method, we conducted a low-resource language Mongolian-Chinese experiment on Transformer. The experimental results are shown in Figure $5(b)$. Compared with the traditional Transformer, our approach improves 2.26 BLEU scores, meanwhile, it also increases 1.09 on the current best performance BR-CSGAN. It is fully proved that our method is also helpful for low-resource translation tasks.

5 Conclusion

In this paper, we propose a novel multi-reward reinforcement learning training paradigm to guide the optimization of model parameters, which makes the reward calculation more extensive. In addition, we employ Gumbel method instead of traditional beam search to selectively sample more random datas in the target space, and combining TD to calculate real-time reward. We validate the effectiveness of our method on the RNNSearch and the Transformer. A large number of experiments clearly show that our approach achieves significant improvements.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron C. Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings*.

William A. Gale and Geoffrey Sampson. 1995. Good-turing frequency estimation without tears. *J. Quant. Linguistics*, 2(3):217–237.

Carl M. Harris and Jay Mandelbaum. 1985. A note on convergence requirements for nonlinear maximum-likelihood estimation of parameters from mixture models. *Computers & OR*, 12(2):237–240.

Wouter Kool, Herke van Hoof, and Max Welling. 2019. Stochastic beams and where to find them: The gumbel-top-k trick for sampling sequences without replacement. In *Proceedings of the 36th International Conference on Machine Learning, ICML 2019, 9-15 June 2019, Long Beach, California, USA*, pages 3499–3508.

Chris J. Maddison, Daniel Tarlow, and Tom Minka. 2014. A* sampling. In *Advances in Neural Information Processing Systems 27: Annual Conference on Neural Information Processing Systems 2014, December 8-13 2014, Montreal, Quebec, Canada*, pages 3086–3094.

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, July 6-12, 2002, Philadelphia, PA, USA*, pages 311–318.

Matthew E. Peters, Mark Neumann, Mohit Iyyer, Matt Gardner, Christopher Clark, Kenton Lee, and Luke Zettlemoyer. 2018. Deep contextualized word representations. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 2227–2237.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*.

Richard S. Sutton. 1988. Learning to predict by the methods of temporal differences. *Machine Learning*, 3:9–44.

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 *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4-9 December 2017, Long Beach, CA, USA*, pages 5998–6008.

Ziyu Wang, Tom Schaul, Matteo Hessel, Hado van Hasselt, Marc Lanctot, and Nando de Freitas. 2016. Dueling network architectures for deep reinforcement learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, pages 1995–2003.

Ronald J. Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine Learning*, 8:229–256.

Lijun Wu, Fei Tian, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018a. A study of reinforcement learning for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, pages 3612–3621.

Lijun Wu, Yingce Xia, Fei Tian, Li Zhao, Tao Qin, Jianhuang Lai, and Tie-Yan Liu. 2018b. Adversarial neural machine translation. In *Proceedings of The 10th Asian Conference on Machine Learning, ACML 2018, Beijing, China, November 14-16, 2018*, pages 534–549.

Zhen Yang, Wei Chen, Feng Wang, and Bo Xu. 2018. Improving neural machine translation with conditional sequence generative adversarial nets. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, pages 1346–1355.

Wenpeng Yin, Hinrich Schütze, Bing Xiang, and Bowen Zhou. 2016. ABCNN: attention-based convolutional neural network for modeling sentence pairs. *TACL*, 4:259–272.

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence, February 4-9, 2017, San Francisco, California, USA*, pages 2852–2858.

Proceedings of the 19th China National Conference on Computational Linguistics, pages 985-994, Hainan, China, October 31 - Novermber 1, 2020.