Bridging the Gap between Training and Inference for Neural Machine Translation

Wen Zhang[1,2] **Yang Feng**[1,2*] **Fandong Meng**[3] **Di You**[4] **Qun Liu**[5]

[1]Key Laboratory of Intelligent Information Processing
Institute of Computing Technology, Chinese Academy of Sciences (ICT/CAS)
[2]University of Chinese Academy of Sciences, Beijing, China
{zhangwen,fengyang}@ict.ac.cn
[3]Pattern Recognition Center, WeChat AI, Tencent Inc, China
fandongmeng@tencent.com
[4]Worcester Polytechnic Institute, Worcester, MA, USA
dyou@wpi.edu
[5]Huawei Noah's Ark Lab, Hong Kong, China
qun.liu@huawei.com

Abstract

Neural Machine Translation (NMT) generates target words sequentially in the way of predicting the next word conditioned on the context words. At training time, it predicts with the ground truth words as context while at inference it has to generate the entire sequence from scratch. This discrepancy of the fed context leads to error accumulation among the way. Furthermore, word-level training requires strict matching between the generated sequence and the ground truth sequence which leads to overcorrection over different but reasonable translations. In this paper, we address these issues by sampling context words not only from the ground truth sequence but also from the predicted sequence by the model during training, where the predicted sequence is selected with a sentence-level optimum. Experiment results on Chinese→English and WMT'14 English→German translation tasks demonstrate that our approach can achieve significant improvements on multiple datasets.

1 Introduction

Neural Machine Translation has shown promising results and drawn more attention recently. Most NMT models fit in the encoder-decoder framework, including the RNN-based (Sutskever et al., 2014; Bahdanau et al., 2015; Meng and Zhang, 2019), the CNN-based (Gehring et al., 2017) and the attention-based (Vaswani et al., 2017) models, which predict the next word conditioned on the previous context words, deriving a language model over target words. The scenario is at training time the ground truth words are used as context while at inference the entire sequence is generated by the resulting model on its own and hence the previous words generated by the model are fed as context. As a result, the predicted words at training and inference are drawn from different distributions, namely, from the data distribution as opposed to the model distribution. This discrepancy, called *exposure bias* (Ranzato et al., 2015), leads to a gap between training and inference. As the target sequence grows, the errors accumulate among the sequence and the model has to predict under the condition it has never met at training time.

Intuitively, to address this problem, the model should be trained to predict under the same condition it will face at inference. Inspired by DATA AS DEMONSTRATOR (DAD) (Venkatraman et al., 2015), feeding as context both ground truth words and the predicted words during training can be a solution. NMT models usually optimize the cross-entropy loss which requires a strict pairwise matching at the word level between the predicted sequence and the ground truth sequence. Once the model generates a word deviating from the ground truth sequence, the cross-entropy loss will correct the error immediately and draw the remaining generation back to the ground truth sequence. However, this causes a new problem. A sentence usually has multiple reasonable translations and it cannot be said that the model makes a mistake even if it generates a word different from the ground truth word. For example,

reference:	We should comply with the rule.
cand1:	We should abide with the rule.
cand2:	We should abide by the law.
cand3:	We should abide by the rule.

*Corresponding author.

Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 4334–4343
Florence, Italy, July 28 - August 2, 2019.

once the model generates "abide" as the third target word, the cross-entropy loss would force the model to generate "with" as the fourth word (as *cand1*) so as to produce larger sentence-level likelihood and be in line with the reference, although "by" is the right choice. Then, "with" will be fed as context to generate "the rule", as a result, the model is taught to generate "abide with the rule" which actually is wrong. The translation *cand1* can be treated as *overcorrection* phenomenon. Another potential error is that even the model predicts the right word "by" following "abide", when generating subsequent translation, it may produce "the law" improperly by feeding "by" (as *cand2*). Assume the references and the training criterion let the model memorize the pattern of the phrase "the rule" always following the word "with", to help the model recover from the two kinds of errors and create the correct translation like *cand3*, we should feed "with" as context rather than "by" even when the previous predicted phrase is "abide by". We refer to this solution as *Overcorrection Recovery* (*OR*).

In this paper, we present a method to bridge the gap between training and inference and improve the overcorrection recovery capability of NMT. Our method first selects *oracle* words from its predicted words and then samples as context from the oracle words and ground truth words. Meanwhile, the oracle words are selected not only with a word-by-word greedy search but also with a sentence-level evaluation, e.g. BLEU, which allows greater flexibility under the pairwise matching restriction of cross-entropy. At the beginning of training, the model selects as context ground truth words at a greater probability. As the model converges gradually, oracle words are chosen as context more often. In this way, the training process changes from a fully guided scheme towards a less guided scheme. Under this mechanism, the model has the chance to learn to handle the mistakes made at inference and also has the ability to recover from overcorrection over alternative translations. We verify our approach on both the RNNsearch model and the stronger Transformer model. The results show that our approach can significantly improve the performance on both models.

2 RNN-based NMT Model

Our method can be applied in a variety of NMT models. Without loss of generality, we take the RNN-based NMT (Bahdanau et al., 2015) as an example to introduce our method. Assume the source sequence and the observed translation are $\mathbf{x} = \{x_1, \cdots, x_{|\mathbf{x}|}\}$ and $\mathbf{y}^* = \{y_1^*, \cdots, y_{|\mathbf{y}^*|}^*\}$.

Encoder. A bidirectional Gated Recurrent Unit (GRU) (Cho et al., 2014) is used to acquire two sequences of hidden states, the annotation of x_i is $h_i = [\overrightarrow{h}_i; \overleftarrow{h}_i]$. Note that e_{x_i} is employed to represent the embedding vector of the word x_i.

$$\overrightarrow{h}_i = \mathbf{GRU}(e_{x_i}, \overrightarrow{h}_{i-1}) \quad (1)$$

$$\overleftarrow{h}_i = \mathbf{GRU}(e_{x_i}, \overleftarrow{h}_{i+1}) \quad (2)$$

Attention. The attention is designed to extract source information (called source context vector). At the j-th step, the relevance between the target word y_j^* and the i-th source word is evaluated and normalized over the source sequence

$$r_{ij} = \mathbf{v}_a^T \tanh\left(\mathbf{W}_a s_{j-1} + \mathbf{U}_a h_i\right) \quad (3)$$

$$\alpha_{ij} = \frac{\exp\left(r_{ij}\right)}{\sum_{i'=1}^{|\mathbf{x}|} \exp\left(r_{i'j}\right)} \quad (4)$$

The source context vector is the weighted sum of all source annotations and can be calculated by

$$c_j = \sum\nolimits_{i=1}^{|\mathbf{x}|} \alpha_{ij} h_i \quad (5)$$

Decoder. The decoder employs a variant of GRU to unroll the target information. At the j-th step, the target hidden state s_j is given by

$$s_j = \mathbf{GRU}(e_{y_{j-1}^*}, s_{j-1}, c_j) \quad (6)$$

The probability distribution P_j over all the words in the target vocabulary is produced conditioned on the embedding of the previous ground truth word, the source context vector and the hidden state

$$t_j = g\left(e_{y_{j-1}^*}, c_j, s_j\right) \quad (7)$$

$$o_j = \mathbf{W}_o t_j \quad (8)$$

$$P_j = \mathrm{softmax}\left(o_j\right) \quad (9)$$

where g stands for a linear transformation, $\mathbf{W}_o$ is used to map t_j to o_j so that each target word has one corresponding dimension in o_j.

3 Approach

The main framework (as shown in Figure 1) of our method is to feed as context either the ground truth words or the previous predicted words, i.e. *oracle*

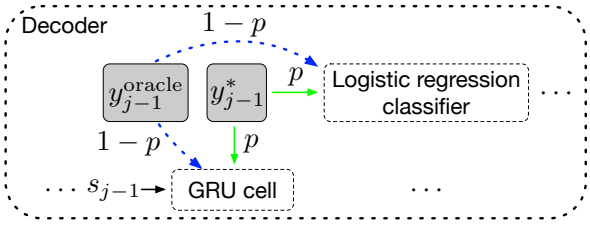

Figure 1: The architecture of our method.

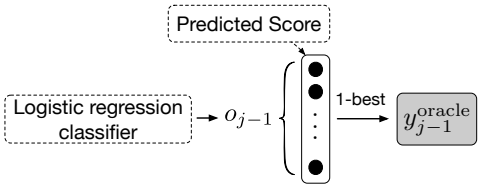

Figure 2: Word-level oracle without noise.

words, with a certain probability. This potentially can reduce the gap between training and inference by training the model to handle the situation which will appear during test time. We will introduce two methods to select the oracle words. One method is to select the oracle words at the word level with a greedy search algorithm, and another is to select a oracle sequence at the sentence-level optimum. The sentence-level oracle provides an option of n-gram matching with the ground truth sequence and hence inherently has the ability of recovering from overcorrection for the alternative context. To predict the j-th target word y_j, the following steps are involved in our approach:

1. Select an oracle word y_{j-1}^{oracle} (at word level or sentence level) at the $\{j-1\}$-th step. (Section **Oracle Word Selection**)

2. Sample from the ground truth word y_{j-1}^{*} with a probability of p or from the oracle word y_{j-1}^{oracle} with a probability of $1-p$. (Section **Sampling with Decay**)

3. Use the sampled word as y_{j-1} and replace the y_{j-1}^{*} in Equation (6) and (7) with y_{j-1}, then perform the following prediction of the attention-based NMT.

3.1 Oracle Word Selection

Generally, at the j-th step, the NMT model needs the ground truth word y_{j-1}^{*} as the context word to predict y_j, thus, we could select an oracle word y_{j-1}^{oracle} to simulate the context word. The oracle word should be a word similar to the ground truth or a synonym. Using different strategies will produce a different oracle word y_{j-1}^{oracle}. One option is that word-level greedy search could be employed to output the oracle word of each step, which is called *Word-level Oracle* (called WO). Besides, we can further optimize the oracle by enlarging the search space with beam search and then re-ranking the candidate translations with a sentence-level metric, e.g. BLEU (Papineni et al., 2002), GLEU (Wu et al., 2016), ROUGE (Lin, 2004), etc, the selected translation is called *oracle sentence*, the words in the translation are *Sentence-level Oracle* (denoted as SO).

Word-Level Oracle

For the $\{j-1\}$-th decoding step, the direct way to select the word-level oracle is to pick the word with the highest probability from the word distribution P_{j-1} drawn by Equation (9), which is shown in Figure 2. The predicted score in o_{j-1} is the value before the $\mathrm{softmax}$ operation. In practice, we can acquire more robust word-level oracles by introducing the *Gumbel-Max* technique (Gumbel, 1954; Maddison et al., 2014), which provides a simple and efficient way to sample from a categorical distribution.

The Gumbel noise, treated as a form of regularization, is added to o_{j-1} in Equation (8), as shown in Figure 3, then $\mathrm{softmax}$ function is performed, the word distribution of y_{j-1} is approximated by

$$\eta = -\log\left(-\log u\right) \tag{10}$$

$$\tilde{o}_{j-1} = \left(o_{j-1} + \eta\right)/\tau \tag{11}$$

$$\tilde{P}_{j-1} = \mathrm{softmax}\left(\tilde{o}_{j-1}\right) \tag{12}$$

where η is the Gumbel noise calculated from a uniform random variable $u \sim \mathcal{U}(0,1)$, τ is temperature. As τ approaches 0, the $\mathrm{softmax}$ function is similar to the argmax operation, and it becomes uniform distribution gradually when $\tau \to \infty$. Similarly, according to $\tilde{P}_{j-1}$, the 1-best word is selected as the word-level oracle word

$$y_{j-1}^{\text{oracle}} = y_{j-1}^{\text{WO}} = \mathrm{argmax}\left(\tilde{P}_{j-1}\right) \tag{13}$$

Note that the Gumbel noise is just used to select the oracle and it does not affect the loss function for training.

Sentence-Level Oracle

The sentence-level oracle is employed to allow for more flexible translation with n-gram matching required by a sentence-level metric. In this paper,

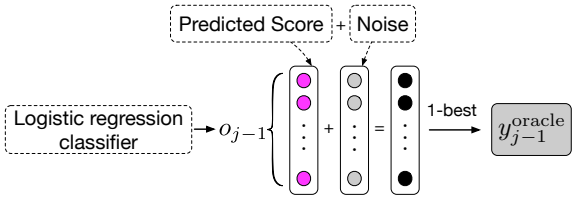

Figure 3: Word-level oracle with Gumbel noise.

we employ BLEU as the sentence-level metric. To select the sentence-level oracles, we first perform beam search for all sentences in each batch, assuming beam size is k, and get k-best candidate translations. In the process of beam search, we also could apply the Gumbel noise for each word generation. We then evaluate each translation by calculating its BLEU score with the ground truth sequence, and use the translation with the highest BLEU score as the *oracle sentence*. We denote it as $\mathbf{y}^{\text{S}} = (y_1^{\text{S}}, ..., y_{|\mathbf{y}^{\text{S}}|}^{\text{S}})$, then at the j-th decoding step, we define the sentence-level oracle word as

$$y_{j-1}^{\text{oracle}} = y_{j-1}^{\text{SO}} = y_{j-1}^{\text{S}} \tag{14}$$

But a problem comes with sentence-level oracle. As the model samples from ground truth word and the sentence-level oracle word at each step, the two sequences should have the same number of words. However we can not assure this with the naive beam search decoding algorithm. Based on the above problem, we introduce *force decoding* to make sure the two sequences have the same length.

Force Decoding. As the length of the ground truth sequence is $|\mathbf{y}^*|$, the goal of force decoding is to generate a sequence with $|\mathbf{y}^*|$ words followed by a special end-of-sentence (EOS) symbol. Therefore, in beam search, once a candidate translation tends to end with EOS when it is shorter or longer than $|\mathbf{y}^*|$, we will force it to generate $|\mathbf{y}^*|$ words, that is,

- If the candidate translation gets a word distribution P_j at the j-th step where $j \leqslant |\mathbf{y}^*|$ and EOS is the top first word in P_j, then we select the top second word in P_j as the j-th word of this candidate translation.
- If the candidate translation gets a word distribution $P_{|\mathbf{y}^*|+1}$ at the $\{|\mathbf{y}^*|+1\}$-th step where EOS is not the top first word in $P_{|\mathbf{y}^*|+1}$, then we select EOS as the $\{|\mathbf{y}^*|+1\}$-th word of this candidate translation.

In this way, we can make sure that all the k candidate translations have $|\mathbf{y}^*|$ words, then re-rank the k candidates according to BLEU score and select the top first as the oracle sentence. For adding Gumbel noise into the sentence-level oracle selection, we replace the P_j with $\tilde{P}_j$ at the j-th decoding step during force decoding.

3.2 Sampling with Decay

In our method, we employ a sampling mechanism to randomly select the ground truth word y_{j-1}^* or the oracle word y_{j-1}^{oracle} as y_{j-1}. At the beginning of training, as the model is not well trained, using y_{j-1}^{oracle} as y_{j-1} too often would lead to very slow convergence, even being trapped into local optimum. On the other hand, at the end of training, if the context y_{j-1} is still selected from the ground truth word y_{j-1}^* at a large probability, the model is not fully exposed to the circumstance which it has to confront at inference and hence can not know how to act in the situation at inference. In this sense, the probability p of selecting from the ground truth word can not be fixed, but has to decrease progressively as the training advances. At the beginning, p=1, which means the model is trained entirely based on the ground truth words. As the model converges gradually, the model selects from the oracle words more often.

Borrowing ideas from but being different from Bengio et al. (2015) which used a schedule to decrease p as a function of the index of mini-batch, we define p with a decay function dependent on the index of training epochs e (starting from 0)

$$p = \frac{\mu}{\mu + \exp\left(e/\mu\right)} \tag{15}$$

where μ is a hyper-parameter. The function is strictly monotone decreasing. As the training proceeds, the probability p of feeding ground truth words decreases gradually.

3.3 Training

After selecting y_{j-1} by using the above method, we can get the word distribution of y_j according to Equation (6), (7), (8) and (9). We do not add the Gumbel noise to the distribution when calculating loss for training. The objective is to maximize the probability of the ground truth sequence based on maximum likelihood estimation (MLE). Thus following loss function is minimized:

$$\mathcal{L}\left(\theta\right) = -\sum\nolimits_{n=1}^{N}\sum\nolimits_{j=1}^{|\mathbf{y}^n|}\log P_j^n\left[y_j^n\right] \tag{16}$$

where N is the number of sentence pairs in the training data, $|\mathbf{y}^n|$ indicates the length of the n-th

ground truth sentence, P_j^n refers to the predicted probability distribution at the j-th step for the n-th sentence, hence $P_j^n\left[y_j^n\right]$ is the probability of generating the ground truth word y_j^n at the j-th step.

4 Related Work

Some other researchers have noticed the problem of exposure bias in NMT and tried to solve it. Venkatraman et al. (2015) proposed DATA AS DEMONSTRATOR (DAD) which initialized the training examples as the paired two adjacent ground truth words and at each step added the predicted word paired with the next ground truth word as a new training example. Bengio et al. (2015) further developed the method by sampling as context from the previous ground truth word and the previous predicted word with a changing probability, not treating them equally in the whole training process. This is similar to our method, but they do not include the sentence-level oracle to relieve the overcorrection problem and neither the noise perturbations on the predicted distribution.

Another direction of attempts is the sentence-level training with the thinking that the sentence-level metric, e.g., BLEU, brings a certain degree of flexibility for generation and hence is more robust to mitigate the exposure bias problem. To avoid the problem of exposure bias, Ranzato et al. (2015) presented a novel algorithm Mixed Incremental Cross-Entropy Reinforce (MIXER) for sequence-level training, which directly optimized the sentence-level BLEU used at inference. Shen et al. (2016) introduced the Minimum Risk Training (MRT) into the end-to-end NMT model, which optimized model parameters by minimizing directly the expected loss with respect to arbitrary evaluation metrics, e.g., sentence-level BLEU. Shao et al. (2018) proposed to eliminate the exposure bias through a probabilistic n-gram matching objective, which trains NMT NMT under the greedy decoding strategy.

5 Experiments

We carry out experiments on the NIST Chinese→English (Zh→En) and the WMT'14 English→German (En→De) translation tasks.

5.1 Settings

For Zh→En, the training dataset consists of 1.25M sentence pairs extracted from LDC corpora[1]. We choose the NIST 2002 (MT02) dataset as the validation set, which has 878 sentences, and the NIST 2003 (MT03), NIST 2004 (MT04), NIST 2005 (MT05) and NIST 2006 (MT06) datasets as the test sets, which contain 919, 1788, 1082 and 1664 sentences respectively. For En→De, we perform our experiments on the corpus provided by WMT'14, which contains 4.5M sentence pairs[2]. We use the newstest2013 as the validation set, and the newstest2014 as the test sets, which containing 3003 and 2737 sentences respectively. We measure the translation quality with BLEU scores (Papineni et al., 2002). For Zh→En, case-insensitive BLEU score is calculated by using the *mteval-v11b.pl* script. For En→De, we tokenize the references and evaluate the performance with case-sensitive BLEU score by the *multi-bleu.pl* script. The metrics are exactly the same as in previous work. Besides, we make statistical significance test according to the method of Collins et al. (2005).

In training the NMT model, we limit the source and target vocabulary to the most frequent 30K words for both sides in the Zh→En translation task, covering approximately 97.7% and 99.3% words of two corpus respectively. For the En→De translation task, sentences are encoded using byte-pair encoding (BPE) (Sennrich et al., 2016) with $37k$ merging operations for both source and target languages, which have vocabularies of 39418 and 40274 tokens respectively. We limit the length of sentences in the training datasets to 50 words for Zh→En and 128 subwords for En→De. For RNNSearch model, the dimension of word embedding and hidden layer is 512, and the beam size in testing is 10. All parameters are initialized by the uniform distribution over $[-0.1, 0.1]$. The mini-batch stochastic gradient descent (SGD) algorithm is employed to train the model parameters with batch size setting to 80. Moreover, the learning rate is adjusted by adadelta optimizer (Zeiler, 2012) with ρ=0.95 and ϵ=1e-6. Dropout is applied on the output layer with dropout rate being 0.5. For Transformer model, we train base model with

[1]These sentence pairs are mainly extracted from LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06

[2]http://www.statmt.org/wmt14/translation-task.html

Systems	Architecture	MT03	MT04	MT05	MT06	Average
Existing end-to-end NMT systems						
Tu et al. (2016)	Coverage	33.69	38.05	35.01	34.83	35.40
Shen et al. (2016)	MRT	37.41	39.87	37.45	36.80	37.88
Zhang et al. (2017)	Distortion	37.93	40.40	36.81	35.77	37.73
Our end-to-end NMT systems						
this work	RNNsearch	37.93	40.53	36.65	35.80	37.73
	+ SS-NMT	38.82	41.68	37.28	37.98	38.94
	+ MIXER	38.70	40.81	37.59	38.38	38.87
	+ OR-NMT	**40.40**‡†*	**42.63**‡†*	**38.87**‡†*	**38.44**‡	**40.09**
	Transformer	46.89	47.88	47.40	46.66	47.21
	+ word oracle	47.42	48.34	47.89	47.34	47.75
	+ sentence oracle	**48.31***	**49.40***	**48.72***	**48.45***	**48.72**

Table 1: Case-insensitive BLEU scores (%) on Zh→En translation task. "‡", "†", "⋆" and "∗" indicate statistically significant difference (p<0.01) from RNNsearch, SS-NMT, MIXER and Transformer, respectively.

default settings (fairseq[3]).

5.2 Systems

The following systems are involved:

RNNsearch: Our implementation of an improved model as described in Section 2, where the decoder employs two GRUs and an attention. Specifically, Equation 6 is substituted with:

$$\tilde{s}_j = \mathbf{GRU}_1(e_{y^*_{j-1}}, s_{j-1}) \quad (17)$$

$$s_j = \mathbf{GRU}_2(c_j, \tilde{s}_j) \quad (18)$$

Besides, in Equation 3, s_{j-1} is replaced with $\tilde{s}_{j-1}$.

SS-NMT: Our implementation of the scheduled sampling (SS) method (Bengio et al., 2015) on the basis of the RNNsearch. The decay scheme is the same as Equation 15 in our approach.

MIXER: Our implementation of the mixed incremental cross-entropy reinforce (Ranzato et al., 2015), where the sentence-level metric is BLEU and the average reward is acquired according to its offline method with a 1-layer linear regressor.

OR-NMT: Based on the RNNsearch, we introduced the word-level oracles, sentence-level oracles and the Gumbel noises to enhance the overcorrection recovery capacity. For the sentence-level oracle selection, we set the beam size to be 3, set τ=0.5 in Equation (11) and μ=12 for the decay function in Equation (15). OR-NMT is the abbreviation of NMT with Overcorrection Recovery.

[3] https://github.com/pytorch/fairseq

5.3 Results on Zh→En Translation

We verify our method on two baseline models with the NIST Zh→En datasets in this section.

Results on the RNNsearch

As shown in Table 1, Tu et al. (2016) propose to model coverage in RNN-based NMT to improve the adequacy of translations. Shen et al. (2016) propose minimum risk training (MRT) for NMT to directly optimize model parameters with respect to BLEU scores. Zhang et al. (2017) model distortion to enhance the attention model. Compared with them, our baseline system RNNsearch 1) outperforms previous shallow RNN-based NMT system equipped with the coverage model (Tu et al., 2016); and 2) achieves competitive performance with the MRT (Shen et al., 2016) and the Distortion (Zhang et al., 2017) on the same datasets. We hope that the strong shallow baseline system used in this work makes the evaluation convincing.

We also compare with the other two related methods that aim at solving the exposure bias problem, including the scheduled sampling (Bengio et al., 2015) (SS-NMT) and the sentence-level training (Ranzato et al., 2015) (MIXER). From Table 1, we can see that both SS-NMT and MIXER can achieve improvements by taking measures to mitigate the exposure bias. While our approach OR-NMT can outperform the baseline system RNNsearch and the competitive comparison systems by directly incorporate the sentence-level oracle and noise perturbations for relieving the overcorrection problem. Particularly, our OR-NMT significantly outperforms the RNNsearch by +2.36 BLEU points averagely on four test datasets. Comparing with the two related models,

Systems	Average
RNNsearch	37.73
+ word oracle	38.94
+ noise	39.50
+ sentence oracle	39.56
+ noise	**40.09**

Table 2: Factor analysis on Zh→En translation, the results are average BLEU scores on MT03∼06 datasets.

our approach further gives a significant improvements on most test sets and achieves improvement by about +1.2 BLEU points on average.

Results on the Transformer

The methods we propose can also be adapted to the stronger Transformer model. The evaluated results are listed in Table 1. Our word-level method can improve the base model by +0.54 BLEU points on average, and the sentence-level method can further bring in +1.0 BLEU points improvement.

5.4 Factor Analysis

We propose several strategies to improve the performance of approach on relieving the overcorrection problem, including utilizing the word-level oracle, the sentence-level oracle, and incorporating the Gumbel noise for oracle selection. To investigate the influence of these factors, we conduct the experiments and list the results in Table 2.

When only employing the word-level oracle, the translation performance was improved by +1.21 BLEU points, this indicates that feeding predicted words as context can mitigate exposure bias. When employing the sentence-level oracle, we can further achieve +0.62 BLEU points improvement. It shows that the sentence-level oracle performs better than the word-level oracle in terms of BLEU. We conjecture that the superiority may come from a greater flexibility for word generation which can mitigate the problem of overcorrection. By incorporating the Gumbel noise during the generation of the word-level and sentence-level oracle words, the BLEU score are further improved by 0.56 and 0.53 respectively. This indicates Gumbel noise can help the selection of each oracle word, which is consistent with our claim that Gumbel-Max provides a efficient and robust way to sample from a categorical distribution.

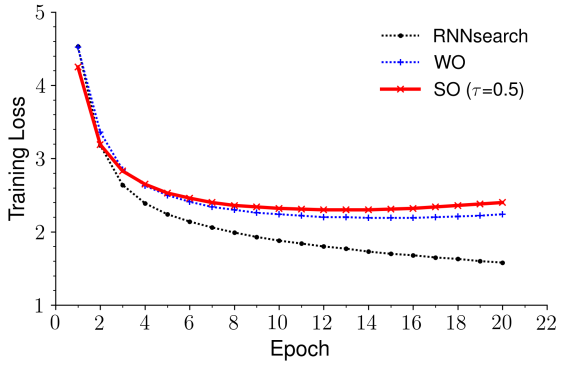

Figure 4: Training loss curves on Zh→En translation with different factors. The black, blue and red colors represent the RNNsearch, RNNsearch with word-level oracle and RNNsearch with sentence-level oracle systems respectively.

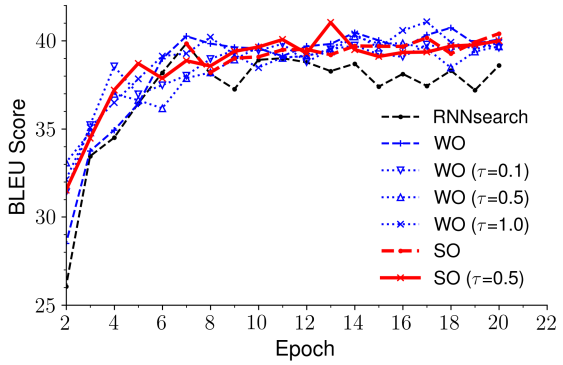

Figure 5: Trends of BLEU scores on the validation set with different factors on the Zh→En translation task.

5.5 About Convergence

In this section, we analyze the influence of different factors for the convergence. Figure 4 gives the training loss curves of the RNNsearch, word-level oracle (WO) without noise and sentence-level oracle (SO) with noise. In training, BLEU score on the validation set is used to select the best model, a detailed comparison among the BLEU score curves under different factors is shown in Figure 5. RNNsearch converges fast and achieves the best result at the 7-th epoch, while the training loss continues to decline after the 7-th epoch until the end. Thus, the training of RNNsearch may encounter the overfitting problem. Figure 4 and 5 also reveal that, integrating the oracle sampling and the Gumbel noise leads to a little slower convergence and the training loss does not keep decreasing after the best results appear on the validation set. This is consistent with our intuition that oracle sampling and noises can avoid overfit-

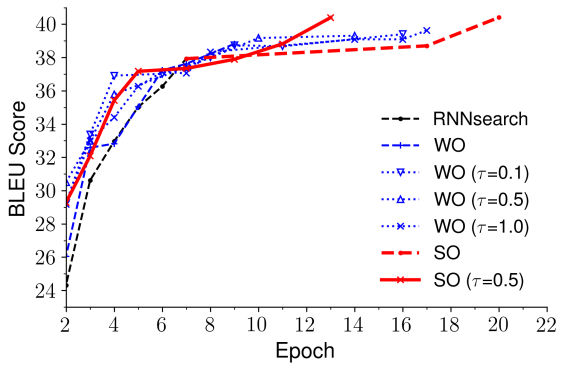

Figure 6: Trends of BLEU scores on the MT03 test set with different factors on the Zh→En translation task.

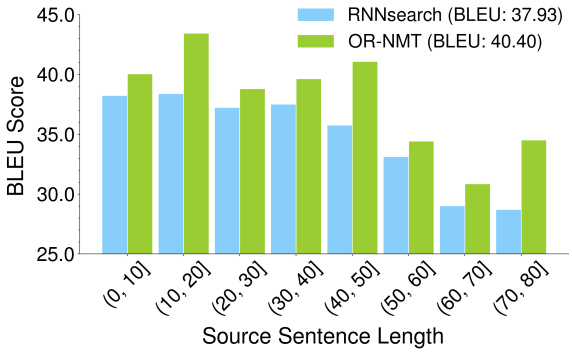

Figure 7: Performance comparison on the MT03 test set with respect to the different lengths of source sentences on the Zh→En translation task.

ting despite needs a longer time to converge.

Figure 6 shows the BLEU scores curves on the MT03 test set under different factors[4]. When sampling oracles with noise (τ=0.5) on the sentence level, we obtain the best model. Without noise, our system converges to a lower BLEU score. This can be understood easily that using its own results repeatedly during training without any regularization will lead to overfitting and quick convergence. In this sense, our method benefits from the sentence-level sampling and Gumbel noise.

5.6 About Length

Figure 7 shows the BLEU scores of generated translations on the MT03 test set with respect to the lengths of the source sentences. In particular, we split the translations for the MT03 test set into different bins according to the length of source sentences, then test the BLEU scores for translations in each bin separately with the results reported in Figure 7. Our approach can achieve big improvements over the baseline system in all bins, especially in the bins (10,20], (40,50] and (70,80] of the super-long sentences. The cross-entropy loss requires that the predicted sequence is exactly the same as the ground truth sequence which is more difficult to achieve for long sentences, while our sentence-level oracle can help recover from this kind of overcorrection.

5.7 Effect on Exposure Bias

To validate whether the improvements is mainly obtained by addressing the exposure bias problem, we randomly select 1K sentence pairs from the Zh→En training data, and use the pre-trained RNNSearch model and proposed model to decode the source sentences. The BLEU score of RNNSearch model was 24.87, while our model produced +2.18 points. We then count the ground truth words whose probabilities in the predicted distributions produced by our model are greater than those produced by the baseline model, and mark the number as $\mathcal{N}$. There are totally $28,266$ gold words in the references, and $\mathcal{N}$=$18,391$. The proportion is $18,391/28,266$=65.06%, which could verify the improvements are mainly obtained by addressing the exposure bias problem.

5.8 Results on En→De Translation

Systems	newstest2014
RNNsearch	25.82
+ SS-NMT	26.50
+ MIXER	26.76
+ OR-NMT	**27.41**‡
Transformer (base)	27.34
+ SS-NMT	28.05
+ MIXER	27.98
+ OR-NMT	**28.65**‡

Table 3: Case-sensitive BLEU scores (%) on En→De task. The "‡" indicates the results are significantly better ($p<0.01$) than RNNsearch and Transformer.

We also evaluate our approach on the WMT'14 benchmarks on the En→De translation task. From the results listed in Table 3, we conclude that the proposed method significantly outperforms the competitive baseline model as well as related approaches. Similar with results on the Zh→En task, both scheduled sampling and MIXER could improve the two baseline systems. Our method im-

[4]Note that the "SO" model without noise is trained based on the pre-trained RNNsearch model (as shown by the red dashed lines in Figure 5 and 6).

proves the RNNSearch and Transformer baseline models by +1.59 and +1.31 BLEU points respectively. These results demonstrate that our model works well across different language pairs.

6 Conclusion

The end-to-end NMT model generates a translation word by word with the ground truth words as context at training time as opposed to the previous words generated by the model as context at inference. To mitigate the discrepancy between training and inference, when predicting one word, we feed as context either the ground truth word or the previous predicted word with a sampling scheme. The predicted words, referred to as oracle words, can be generated with the word-level or sentence-level optimization. Compared to word-level oracle, sentence-level oracle can further equip the model with the ability of overcorrection recovery. To make the model fully exposed to the circumstance at reference, we sample the context word with decay from the ground truth words. We verified the effectiveness of our method with two strong baseline models and related works on the real translation tasks, achieved significant improvement on all the datasets. We also conclude that the sentence-level oracle show superiority over the word-level oracle.

Acknowledgments

We thank the three anonymous reviewers for their valuable suggestions. This work was supported by National Natural Science Foundation of China (NO. 61662077, NO. 61876174) and National Key R&D Program of China (NO. YS2017YFGH001428).

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. *ICLR 2015*.

Samy Bengio, Oriol Vinyals, Navdeep Jaitly, and Noam Shazeer. 2015. Scheduled sampling for sequence prediction with recurrent neural networks. In C. Cortes, N. D. Lawrence, D. D. Lee, M. Sugiyama, and R. Garnett, editors, *Advances in Neural Information Processing Systems 28*, pages 1171–1179. Curran Associates, Inc.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using rnn encoder–decoder for statistical machine translation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar. Association for Computational Linguistics.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 531–540, Ann Arbor, Michigan. Association for Computational Linguistics.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann N. Dauphin. 2017. Convolutional sequence to sequence learning. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1243–1252, International Convention Centre, Sydney, Australia. PMLR.

Emil Julius Gumbel. 1954. Statistical theory of extreme valuse and some practical applications. *Nat. Bur. Standards Appl. Math. Ser. 33*.

Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text Summarization Branches Out: Proceedings of the ACL-04 Workshop*, pages 74–81, Barcelona, Spain. Association for Computational Linguistics.

Chris J Maddison, Daniel Tarlow, and Tom Minka. 2014. A* sampling. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3086–3094. Curran Associates, Inc.

Fandong Meng and Jinchao Zhang. 2019. Dtmt: A novel deep transition architecture for neural machine translation. Proceedings of AAAI.

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 on association for computational linguistics*, pages 311–318. Association for Computational Linguistics.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2015. Sequence level training with recurrent neural networks. *arXiv preprint arXiv:1511.06732*.

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 (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Chenze Shao, Xilin Chen, and Yang Feng. 2018. Greedy search with probabilistic n-gram matching for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 4778–4784.

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 (Volume 1: Long Papers)*, volume 1, pages 1683–1692.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems 27*, pages 3104–3112. Curran Associates, Inc.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, editors, *Advances in Neural Information Processing Systems 30*, pages 5998–6008. Curran Associates, Inc.

Arun Venkatraman, Martial Hebert, and J. Andrew Bagnell. 2015. Improving multi-step prediction of learned time series models. In *Proceedings of the Twenty-Ninth AAAI Conference on Artificial Intelligence*, AAAI'15, pages 3024–3030. AAAI Press.

Yonghui Wu, Mike Schuster, Zhifeng Chen, Quoc V Le, Mohammad Norouzi, Wolfgang Macherey, Maxim Krikun, Yuan Cao, Qin Gao, Klaus Macherey, et al. 2016. Google's neural machine translation system: Bridging the gap between human and machine translation. *arXiv preprint arXiv:1609.08144*.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Jinchao Zhang, Mingxuan Wang, Qun Liu, and Jie Zhou. 2017. Incorporating word reordering knowledge into attention-based neural machine translation. In *Proceedings of ACL*.