Confidence-Aware Scheduled Sampling for Neural Machine Translation

Yijin Liu[1]*, Fandong Meng[2], Yufeng Chen[1] Jinan Xu[1]†, and Jie Zhou[2]
[1]Beijing Jiaotong University, China
[2]Pattern Recognition Center, WeChat AI, Tencent Inc, China
adaxry@gmail.com
{fandongmeng, withtomzhou}@tencent.com
{jaxu,chenyf}@bjtu.edu.cn

Abstract

Scheduled sampling is an effective method to alleviate the exposure bias problem of neural machine translation. It simulates the inference scene by randomly replacing ground-truth target input tokens with predicted ones during training. Despite its success, its critical schedule strategies are merely based on training steps, ignoring the real-time model competence, which limits its potential performance and convergence speed. To address this issue, we propose confidence-aware scheduled sampling. Specifically, we quantify real-time model competence by the confidence of model predictions, based on which we design fine-grained schedule strategies. In this way, the model is exactly exposed to predicted tokens for high-confidence positions and still ground-truth tokens for low-confidence positions. Moreover, we observe vanilla scheduled sampling suffers from degenerating into the original teacher forcing mode since most predicted tokens are the same as ground-truth tokens. Therefore, under the above confidence-aware strategy, we further expose more noisy tokens (*e.g.,* wordy and incorrect word order) instead of predicted ones for high-confidence token positions. We evaluate our approach on the Transformer and conduct experiments on large-scale WMT 2014 English-German, WMT 2014 English-French, and WMT 2019 Chinese-English. Results show that our approach significantly outperforms the Transformer and vanilla scheduled sampling on both translation quality and convergence speed.

1 Introduction

Neural Machine Translation (NMT) has made promising progress in recent years (Sutskever et al., 2014; Bahdanau et al., 2015; Vaswani et al., 2017). Generally, NMT models are trained to maximize the likelihood of the next token given previous golden tokens as inputs, *i.e.,* teacher forcing (Goodfellow et al., 2016). However, at the inference stage, golden tokens are unavailable. The model is exposed to an unseen data distribution generated by itself. This discrepancy between training and inference is named as the *exposure bias* problem (Ranzato et al., 2016).

Many techniques have been proposed to alleviate the exposure bias problem. To our knowledge, they mainly fall into two categories. The one is sentence-level training, which treats the sentence-level metric (*e.g.,* BLEU) as a reward, and directly maximizes the expected rewards of generated sequences (Ranzato et al., 2016; Shen et al., 2016; Rennie et al., 2017). Although intuitive, they generally suffer from slow and unstable training due to the high variance of policy gradients and the credit assignment problem (Sutton, 1984; Liu et al., 2018; Wang et al., 2018). Another category is sampling-based approaches, aiming to simulate the data distribution of reference during training. Scheduled sampling (Bengio et al., 2015) is a representative method, which samples tokens between golden references and model predictions with a scheduled probability. Zhang et al. (2019) further refine the sampling space of scheduled sampling with predictions from beam search. Mihaylova and Martins (2019) and Duckworth et al. (2019) extend scheduled sampling to the Transformer with a novel two-pass decoding architecture.

Although these sampling-based approaches have been shown effective, most of them schedule the sampling probability based on training steps. We argue this schedule strategy has two following limitations: 1) It is far from exactly reflecting the real-time model competence; 2) It is only based on training steps and equally treat all token positions, which is too coarse-grained to guide the

* This work was done when Yijin Liu was interning at Pattern Recognition Center, WeChat AI, Tencent Inc, China
† Jinan Xu is the corresponding author of the paper.

Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021, pages 2327–2337

sampling selection for each target token. These two limitations yield an inadequate and inefficient schedule strategy, which hinders the potential performance and convergence speed of vanilla scheduled sampling-based approaches.

To address these issues, we propose confidence-aware scheduled sampling. Specifically, we take the model prediction confidence as the assessment of real-time model competence, based on which we design fine-grained schedule strategies. Namely, we sample predicted tokens as target inputs for high-confidence positions and still ground-truth tokens for low-confidence positions. In this way, the NMT model is exactly exposed to corresponding tokens according to its real-time competence rather than coarse-grained predefined patterns. Additionally, we observe that most predicted tokens are the same as ground-truth tokens due to teacher forcing[1], degenerating scheduled sampling to the original teacher forcing mode. Therefore, we further expose more noisy tokens (Meng et al., 2020) (*e.g.,* wordy and incorrect word order) instead of predicted ones for high-confidence token positions. Experimentally, we evaluate our approach on the Transformer (Vaswani et al., 2017) and conduct experiments on large-scale WMT 2014 English-German (EN-DE), WMT 2014 English-French (EN-FR), and WMT 2019 Chinese-English (ZH-EN).

The main contributions of this paper can be summarized as follows[2]:

- To the best of our knowledge, we are the first to propose confidence-aware scheduled sampling for NMT, which exactly samples corresponding tokens according to the real-time model competence rather than coarse-grained predefined patterns.
- We further explore to sample more noisy tokens for high-confidence token positions, preventing scheduled sampling from degenerating into the original teacher forcing mode.
- Our approach significantly outperforms the Transformer by 1.01, 1.03, 0.98 BLEU and outperforms the stronger scheduled sampling by 0.51, 0.41, and 0.58 BLEU on EN-DE, EN-FR, and ZH-EN, respectively. Our approach speeds up model convergence about 3.0× faster than the Transformer and about 1.8× faster than vanilla scheduled sampling.
- Extensive analyses indicate the effectiveness and superiority of our approach on longer sentences. Moreover, our approach can facilitate the training of the Transformer model with deeper decoder layers.

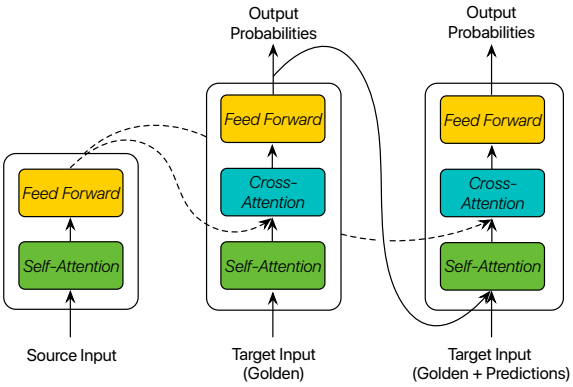

Figure 1: Scheduled sampling for the transformer with two-pass decoding (Mihaylova and Martins, 2019).

2 Background

2.1 Neural Machine Translation

Given a pair of source language $\mathbf{X} = \{x_1, x_2, \cdots, x_m\}$ with m tokens and target language $\mathbf{Y} = \{y_1, y_2, \cdots, y_n\}$ with n tokens, neural machine translation aims to model the following translation probability:

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{t=1}^{n} P(y_t|\mathbf{y}_{<t}, \mathbf{X}, \theta) = \sum_{t=1}^{n} \log P(y_t|\mathbf{y}_{<t}, \mathbf{X}, \theta) \quad (1)$$

where t is the index of target tokens, $\mathbf{y}_{<t}$ is the partial translation before y_t, and θ is model parameter. In the training stage, $\mathbf{y}_{<t}$ are ground-truth tokens, and this procedure is also known as teacher forcing. The translation model is generally trained with maximum likelihood estimation (MLE).

2.2 Scheduled Sampling for the Transformer

Scheduled sampling is initially designed for Recurrent Neural Networks (Bengio et al., 2015), and further modifications are needed when applied to the Transformer (Mihaylova and Martins, 2019;

[1]We observe that about 70% tokens are correctly predicted in WMT14 EN-DE training data.

[2]Codes are available at https://github.com/Adaxry/conf_aware_ss4nmt.

Duckworth et al., 2019). As shown in Figure 1, we follow the two-pass decoding architecture. In the first pass, the model conducts the same as a standard NMT model. Its predictions are used to simulate the inference scene[3]. In the second pass, inputs of the decoder $\widetilde{\mathbf{y}}_{<t}$ are sampled from predictions of the first pass and ground-truth tokens with a certain probability. Finally, predictions of the second pass are used to calculate the cross-entropy loss, and Equation (1) is modified as follow:

$$P(\mathbf{Y}|\mathbf{X}) = \sum_{t=1}^{n} logP(y_t|\widetilde{\mathbf{y}}_{<t}, \mathbf{X}, \theta) \quad (2)$$

Note that the two decoders are identical and share the same parameters. At inference, only the first decoder is used, that is just the standard Transformer. How to schedule the above probability of sampling tokens is the key point, which is exactly what we aim to improve in this paper.

2.3 Decay Strategies on Training Steps

Existing schedule strategies are based on training steps (Bengio et al., 2015; Zhang et al., 2019). As the number of the training step i increases, the model should be exposed to its own predictions more frequently. At the i-th training step, the probability of sampling golden tokens $f(i)$ is calculated as follow:

- Linear Decay: $f(i) = \max(\epsilon, ki + b)$, where ϵ is the minimum value, and $k < 0$ and b are respectively the slope and offset of the decay.
- Exponential Decay: $f(i) = k^i$, where $k < 1$ is the radix to adjust the sharpness of the decay.
- Inverse Sigmoid Decay: $f(i) = \frac{k}{k+e^{\frac{i}{k}}}$, where e is the mathematical constant, and $k \geq 1$ is a hyperparameter to adjust the sharpness of the decay.

We draw visible examples for different decay strategies in Figure 2.

3 Approaches

In this section, we firstly describe how to estimate model confidence at each token position. Secondly, we elaborate the fine-grained schedule strategy based on model confidence. Finally, we explore to sample more noisy tokens instead of predicted tokens for high-confidence positions.

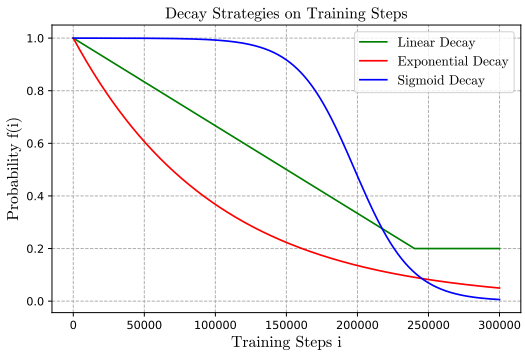

Figure 2: Examples of different decay strategies $f(i)$.

3.1 Model Confidence Estimation

We explore two approaches to estimate model confidence at each token position.

Predicted Translation Probability (PTP). Current NMT models are well-calibrated with regularization techniques in the training setting (Ott et al., 2018; Müller et al., 2019; Wang et al., 2020). Namely the predicted translation probability can directly serve as the model confidence. At the t-th target token position, we calculate the model confidence $conf(t)$ as follow:

$$conf(t) = P(y_t|\mathbf{y}_{<t}, \mathbf{X}, \theta) \quad (3)$$

Since we base our approach on the Transformer with two-pass decoding (Mihaylova and Martins, 2019; Duckworth et al., 2019), above predicted translation probability can be directly obtained in the first-pass decoding (shown in Figure 1), causing no additional computation costs.

Monte Carlo Dropout Sampling. The model confidence can be quantified by Bayesian neural networks (Buntine and Weigend, 1991; Neal, 2012), which place distributions over the weights of neural networks. We adopt widely used Monte Carlo dropout sampling (Gal and Ghahramani, 2016; Wang et al., 2019b) to approximate Bayesian inference. Given a batch of training data and current NMT model parameterized by θ, we repeatedly conduct forward propagation K times[4]. On the k-th propagation, part of neurons $\hat{\theta}^{(k)}$ in network

[3]Following Goyal et al. (2017), model predictions are the weighted sum of target embeddings over output probabilities. As model predictions cause a mismatch with golden tokens, they can simulate translation errors of the inference scene.

[4]We empirically set K to 5 following Wan et al. (2020).

θ are randomly deactivated. Eventually, we obtain K sets of model parameters $\{\hat{\theta}^{(k)}\}_{k=1}^{K}$ and corresponding translation probabilities. We use the expectation or variance of translation probabilities to estimate the model confidence (Wang et al., 2019b). Intuitively, the higher expectation or, the lower variance of translation probabilities reflects higher model confidence. Formally at the t-th token position, we estimate the model confidence $conf(t)$ that calculated by the expectation of translation probabilities:

$$conf(t) = \mathbb{E}\left[P(y_t|\mathbf{y}_{<t}, \mathbf{X}, \hat{\theta}^{(k)})\right]_{k=1}^{K} \quad (4)$$

We also use the variance of translation probabilities to estimate the model confidence $conf(t)$ as an alternative:

$$conf(t) = 1 - \mathrm{Var}\left[P(y_t|\mathbf{y}_{<t}, \mathbf{X}, \theta)\right]_{k=1}^{K} \quad (5)$$

where $\mathrm{Var}[\cdot]$ denotes the variance of a distribution that calculated following the setting in (Wang et al., 2019b; Zhou et al., 2020). We will further analyze the effect of different confidence estimations in Section 4.2.

3.2 Confidence-Aware Scheduled Sampling

The confidence score $conf(t)$ quantifies whether the current NMT model is confident or hesitant on predicting the t-th target token. We take $conf(t)$ as exact and real-time information to conduct a fine-grained schedule strategy in each training iteration. Specifically, a lower $conf(t)$ indicates that the current model θ still struggles with the teacher forcing mode for the t-th target token, namely underfitting for the conditional probability $P(y_t|\mathbf{y}_{<t}, \mathbf{X}, \theta)$. Thus we should keep feeding ground-truth tokens for learning to predict the t-th target token. Conversely, a higher $conf(t)$ indicates the current model θ has learned well the basic conditional probability under teacher forcing. Thus we should empower the model with the ability to cope with the exposure bias problem. Namely, we take inevitably erroneous model predictions as target inputs for learning to predict the t-th target.

Formally, in the second-pass decoding, the above fine-grained schedule strategy is conducted at all decoding steps simultaneously:

$$y_{t-1} = \begin{cases} y_{t-1} & \text{if } conf(t) \leq t_{golden} \\ \hat{y}_{t-1} & \text{else} \end{cases} \quad (6)$$

Dataset	Size (M)	Valid / Test set
WMT14 EN-DE	4.5	newstest 2013 / 2014
WMT14 EN-FR	36	newstest 2013 / 2014
WMT19 ZH-EN	20	newstest 2018 / 2019

Table 1: Dataset statistics in our experiments.

where t_{golden} is a threshold to measure whether $conf(t)$ is high enough (*e.g.,* 0.9) to sample the predicted token $\hat{y}_{t-1}$.

3.3 Confidence-Aware Scheduled Sampling with Target Denoising

Considering predicted tokens are obtained from the teacher forcing model, most predicted tokens (*e.g.,* about 70% tokens in WMT14 EN-DE) are the same as ground-truth tokens, which degenerate the scheduled sampling to the original teacher forcing. Although previous study (Zhang et al., 2019) have proposed to address this issue by using predictions from beam search, it conducts very slowly (about 4× slower than ours) due to the autoregressive property of beam search decoding. To avoid the above degeneration problem while preserving computational efficiency, we try to add more noisy tokens instead of predicted tokens for high-confidence positions. Inspired by Meng et al. (2020), we replace ground-truth y_{t-1} with a random token y_{rand} of the current target sentence, which can simulate wordy and incorrect word order phenomena that occur at inference. Considering y_{rand} is more difficult[5] to learn than $\hat{y}_{t-1}$, we only adopt the noisy y_{rand} for higher confidence positions. Therefore, the fine-grained schedule strategy in Equation 6 is extended to:

$$y_{t-1} = \begin{cases} y_{t-1} & \text{if } conf(t) \leq t_{golden} \\ \hat{y}_{t-1} & \text{if } t_{golden} < conf(t) \leq t_{rand} \\ y_{rand} & \text{if } conf(t) > t_{rand} \end{cases} \quad (7)$$

where t_{rand} is a threshold to measure whether $conf(t)$ is high enough (*e.g.,* 0.95) to sample the random target token y_{rand}. We provide detailed selections about t_{golden} and t_{rand} in Section 4.2.

[5]Given a pre-trained Transformer$_{base}$ model, we respectively replace ground-truth tokens with predicted tokens $\hat{y}$ or random tokens y_{rand} with the same rate, and measure such difficulty by the increment of model perplexity. We observe that y_{rand} yields about 15% higher model perplexity than $\hat{y}$.

4 Experiments

We conduct experiments on three large-scale WMT 2014 English-German (EN-DE), WMT 2014 English-French (EN-FR), and WMT 2019 Chinese-English (ZH-EN) translation tasks. We respectively build a shared source-target vocabulary for the EN-DE and EN-FR datasets, and unshared vocabularies for the ZH-EN dataset. We apply byte-pair encoding (Sennrich et al., 2016) with 32k merge operations for all datasets. More datasets statistics are listed in Table 1.

4.1 Implementation Details

Training Setup. We train the Transformer$_{base}$ and Transformer$_{big}$ models (Vaswani et al., 2017) with the open-source THUMT (Zhang et al., 2017). All Transformer models are first trained by teacher forcing with 100k steps, and then trained with different training objects or scheduled sampling approaches for 300k steps. All experiments are conducted on 8 NVIDIA Tesla V100 GPUs, where each is allocated with a batch size of approximately 4096 tokens. We use Adam optimizer (Kingma and Ba, 2014) with 4000 warmup steps. During training and the Monte Carlo Dropout process, we set dropout (Srivastava et al., 2014) rate to 0.1 for the Transformer$_{base}$ and 0.3 for the Transformer$_{big}$.

Evaluation. We set the beam size to 4 and the length penalty to 0.6 during inference. We use *multibleu.perl* to calculate case-sensitive BLEU scores for WMT14 EN-DE and EN-FR, and use *mteval-v13a.pl* to calculate case-sensitive BLEU scores for WMT19 ZH-EN. We use the paired bootstrap resampling methods (Koehn, 2004) to compute the statistical significance of test results.

4.2 Hyperparameter Experiments

In this section, we elaborate hyperparameters settings involved in our approaches according to the performance on the validation set of WMT14 EN-DE, and share these settings for all WMT tasks.

Different Confidence Estimations. In this section, we analyze effects of different estimations for model confidence described in Section 3.1. As shown in Table 2, we observe that Monte Carlo dropout sampling based approaches (*i.e.,* expectation and variance of translation probabilities) achieve comparable or marginally better translation quality than PTP. However, since Monte Carlo dropout sampling based approaches need additional passes for forward propagation, which yields about 2.7× computation costs than the Transformer$_{base}$. On the contrary, PTP only causes marginal additional computation costs (1.3×) than the Transformer$_{base}$, as PTP can be directly obtained in the first pass decoding. Considering the trade-off between training efficiency and final performance, we use PTP to estimate model confidence by default in the following experiments.

Methods	Training Cost	BLEU	Δ
Transformer$_{base}$	ref.	27.10	ref.
+ PTP	1.3×	28.15	+1.05
+ Expectation	2.7×	28.15	+1.05
+ Variance	2.7×	28.20	+1.10

Table 2: BLUE scores (%) on the validation set of WMT14 EN-DE with different confidence estimations. 'Training Cost' is calculated by the total training time until models convergence on 8 NVIDIA V100 GPUs. 'PTP' refers to PTP-based confidence estimation in Equation (3). 'Expectation' and 'Variance' refers to Monte Carlo dropout sampling-based confidence estimation in Equation (4) and (5), respectively. 'ref.' is short for the reference baseline.

Thresholds Settings. There are two important hyperparameters in our approaches, namely the two threshold t_{golden} and t_{rand} that determine token selections in Equation (7). In our preliminary experiments, we observe our approach is relatively not sensitive to t_{golden}, thus we firstly fix t_{golden} to a modest value, i.e., 0.5 and analyze effects when t_{rand} ranging from 0.5 to 0.95. As the red line is shown in Figure 3, we observe that a rapid improvement in performance with the growth of t_{rand}. Therefore, we decide to set t_{rand} to 0.95 and then analyze effects when t_{golden} ranging from 0.5 to 0.95. As the blue line is shown in Figure 3, the model performance gently rises with the growth of t_{golden} and finally achieves its peak when $t_{golden} = 0.9$. Thus we finally set t_{golden} to 0.9.

4.3 Systems

Mixer. A sequence-level training algorithm for text generations by combining both REINFORCE and cross-entropy (Ranzato et al., 2016).

Minimal Risk Training. Minimal Risk Training (MRT) (Shen et al., 2016) introduces evaluation metrics (*e.g.,* BLEU) as loss functions and aims to minimize expected loss on the training data.

Model	BLEU		
	EN-DE	ZH-EN	EN-FR
Transformer$_{base}$ (Vaswani et al., 2017)	27.30	–	38.10
Transformer$_{base}$ (Vaswani et al., 2017) †	27.90	24.97	40.30
+ Mixer (Ranzato et al., 2016) †	28.54	25.28	40.57
+ Minimal Risk Training (Shen et al., 2016) †	28.55	25.23	40.82
+ TeaForN (Goodman et al., 2020)	27.90	–	40.84
+ TeaForN (Goodman et al., 2020) †	28.60	25.45	40.94
+ Self-paced learning (Wan et al., 2020) †	28.85	25.56	41.12
+ Vanilla scheduled sampling (Bengio et al., 2015) †	28.40	25.43	40.87
+ Target denoising (Meng et al., 2020) †	28.55	25.58	40.57
+ Sampling with sentence oracles (Zhang et al., 2019)	28.65	–	–
+ Sampling with sentence oracles (Zhang et al., 2019) †	28.65	25.50	40.85
+ Confidence-aware scheduled sampling (ours) †	28.80$*$	25.95$**$	41.19$**$
+ Confidence-aware scheduled sampling with target denoising (ours) †	**28.91**$**$	**26.00**$**$	**41.28**$**$
Transformer$_{big}$ (Vaswani et al., 2017)	28.40	–	41.80
Transformer$_{big}$ (Vaswani et al., 2017) †	28.90	25.22	41.89
+ Mixer (Ranzato et al., 2016) †	29.27	25.58	42.37
+ Minimal Risk Training (Shen et al., 2016) †	29.35	25.65	42.46
+ TeaForN (Goodman et al., 2020)	29.30	–	42.73
+ TeaForN (Goodman et al., 2020) †	29.32	25.48	42.62
+ Error correction (Song et al., 2020)	29.20	–	–
+ Self-paced learning (Wan et al., 2020) †	29.68	25.56	42.32
+ Vanilla scheduled sampling (Bengio et al., 2015) †	29.62	25.60	42.55
+ Target denoising (Meng et al., 2020) †	29.18	25.56	42.32
+ Scheduled sampling with sentence oracles (Zhang et al., 2019) †	29.57	25.78	42.65
+ Confidence-aware scheduled sampling (ours) †	29.95$**$	26.00$**$	42.90$**$
+ Confidence-aware scheduled sampling with target denoising (ours) †	**30.09**$**$	**26.27**$**$	**42.97**$**$

Table 3: Translation performance on each WMT dataset. '†' is our implementations under unified settings. The original TeaForN (Goodman et al., 2020) reports SacreBLEU scores. For fair comparison, we re-implement it and report BLEU scores. '$*$/$**$': significantly (Koehn, 2004) better than 'Vanilla Scheduled Sampling' with $p < 0.05$ and $p < 0.01$.

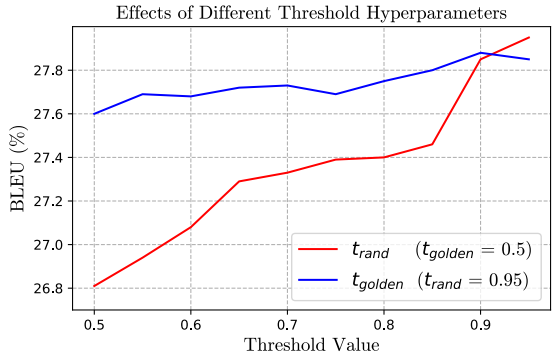

Figure 3: BLUE scores (%) on the validation set of WMT14 EN-DE with different t_{golden} and t_{rand}.

TeaForN. Teacher forcing with n-grams (Goodman et al., 2020) enable the standard teacher forcing with a broader view by n-grams optimization.

Self-paced learning. Wan et al. (2020) assign confidence scores for each input to weight its loss.

Vanilla schedule sampling. Scheduled sampling on training steps with the inverse sigmoid decay (Bengio et al., 2015; Zhang et al., 2019).

Sampling with sentence oracles. Zhang et al. (2019) refine the sampling space of scheduled sampling with sentence oracles, *i.e.,* predictions from beam search. Note that its sampling strategy is still based on training steps with the sigmoid decay.

Target denoising. Meng et al. (2020) add noisy perturbations into decoder inputs when training, which yields a more robust translation model against prediction errors by target denoising.

Confidence-aware scheduled sampling. Our fine-grained schedule strategy described in Equation (6) with $t_{golden} = 0.9$.

Confidence-aware scheduled sampling with target denoising. Our fine-grained schedule strategy described in Equation (7) with $t_{golden} = 0.9$

Schedule Strategy	BLEU	Δ
Transformer$_{base}$	27.10	ref.
+ Linear decay	27.56$*$	+0.46
+ Exponential decay	27.60$*$	+0.50
+ Inverse sigmoid decay	27.65$*$	+0.55
+ Confidence (ours)	**28.15**$**$	**+1.05**

Table 4: BLUE scores (%) on the validation set of WMT14 EN-DE with different schedule strategies. 'Confidence' refers to the confidence-aware strategy in Equation (6). 'ref.' is short for the reference baseline. '$*$ / $**$': significantly (Koehn, 2004) better than the Transformer$_{base}$ with $p < 0.05$ and $p < 0.01$.

Model	BLEU	Δ
Our approach	28.15	ref.
− Confidence	27.75	-0.40
− Denoising	28.00	-0.15
− Confidence & Denoising	27.64	-0.51

Table 5: BLUE scores (%) on the validation set of WMT14 EN-DE for ablation experiments. 'Our approach' is 'confidence-aware scheduled sampling with target denoising' in Equation (7). 'Confidence' refers to the confidence-aware strategy in Equation (7). 'Denoising' refers to the target random noise y_{rand} in Equation (7). 'ref.' is short for the reference baseline.

and $t_{rand} = 0.95$.

4.4 Main Results

We list translation qualities in Table 3. For the Transformer$_{base}$ baseline, our 'Confidence-aware scheduled sampling' shows consistent improvements by 0.90, 0.98, 0.89 BLEU points on EN-DE, ZH-EN, and EN-FR, respectively. Moreover, after applying the more fine-grained strategy with target denoising, our 'Confidence-aware scheduled sampling with target denoising' achieves further improvements which are 1.01, 1.03, 0.98 BLEU points on EN-DE, ZH-EN, and EN-FR, respectively. When comparing with the stronger vanilla scheduled sampling method, 'Confidence-aware scheduled sampling with target denoising' still yields improvements by 0.51, 0.57, and 0.41 BLEU points on the above three tasks, respectively. For the more powerful Transformers$_{big}$, we also observe similar experimental conclusions as above. Specifically, 'Confidence-aware scheduled sampling with target denoising' outperforms vanilla scheduled sampling by 0.47, 0.67, and 0.42 BLEU points, respectively. In summary, experiments on strong baselines and various tasks verify the effectiveness and superiority of our approaches.

5 Analysis and Discussion

We analyze our proposals on WMT 2014 EN-DE with the Transformer$_{base}$ model.

5.1 Effects of Confidence-Aware Strategies

In this section, we rigorously validate the effectiveness of confidence-aware strategies by univariate experiments with the only difference at schedule strategy. As shown in Table 4, existing heuristic functions, *i.e.,* linear, exponential, and inverse sigmoid decay, moderately bring improvements over the Transformer$_{base}$ baseline by 0.46, 0.50, and 0.55 BLEU points, respectively. While our confidence-aware strategy that described in Equation (6) can significantly outperform the baseline by 1.05 BLEU points. We attribute the effectiveness of the confidence-aware strategy to its exact and suitable token assignments according to the real-time model competence rather than predefined patterns.

5.2 Ablation Experiments

We conduct ablation experiments to investigate the impacts of various components in our 'Confidence-aware scheduled sampling with target denoising' (described in Equation (7)) and list results in Table 5. Separately removing the confidence-aware strategy degenerates our approach into the vanilla target denoising with a uniform strategy (Meng et al., 2020), which causes a noticeable drop (0.4 BLEU), indicating the confidence-aware strategy plays a leading role for performance. On the other hand, we only observe a drop (0.15 BLEU) when removing 'Target denoising', revealing the additional noise plays a secondary role for performance. Finally, ablating both the confidence-aware strategy and 'Target denoising' degenerates our approach into the vanilla scheduled sampling. It yields a further decrease (0.51 BLEU), suggesting the confidence-aware strategy and 'Target denoising' are complementary with each other.

5.3 Different Numbers of Decoder Layers

As known in existing studies (Domhan, 2018; Wang et al., 2019a), there exists a performance bottleneck at the decoder side of NMT models. Namely, the increase in the number of decoder layers can not bring corresponding improvements for performance. He et al. (2019) attribute this bottle-

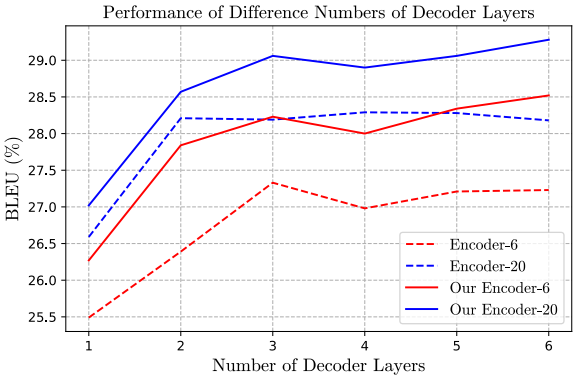

Figure 4: BLUE scores (%) on the validation set of WMT14 EN-DE with different numbers of decoder layers. Solid lines refer to our confidence-aware schedule strategy. Dashed lines refer to the Transformer$_{base}$.

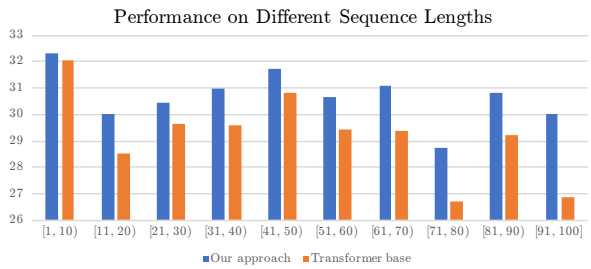

Figure 5: BLUE scores (%) on the randomly sampled WMT14 EN-DE training data with different lengths.

neck to the fact that decoders learn an easier task than encoders.

In this paper, our fine-grained schedule strategy in Equation (7) assigns a more difficult task to the decoder. We can not help wondering whether our strategy is able to alleviate the above performance bottleneck. Firstly, we keep the number of encoders fixed to 6 (*i.e.,* Encoder-6), then apply our confidence-aware schedule strategy on the Encoder-6 Transformer$_{base}$ with the number of decoder layers ranging from 1 to 6. As shown in Figure 4, our approach (solid red line) consistently outperforms the Encoder-6 Transformer$_{base}$ (dashed red line). More importantly, the improvement of Encoder-6 Transformer$_{base}$ stops (*i.e.,* performance bottleneck) once the number of decoder exceeds 4. Despite this, we observe continuous improvement with the growth of decoder layers in our approach. Moreover, we repeat the above experiments with more powerful deep encoders (Encoder-20). We observe that the performance bottleneck for Encoder-20 Transformer$_{base}$ becomes more evident (dashed blue line). Despite this, our approaches (solid blue line) still keep improving performance with the growth of decoder layers on the stronger Encoder-20 Transformer$_{base}$.

In summary, our confidence-aware schedule strategy brings a meaningful increase in the difficulty of decoders, and the bottleneck at the decoder side is alleviated to a certain extend.

5.4 Effects on Different Sequence Lengths

Due to error accumulations, the exposure bias problem becomes more problematic with the growth of sequence lengths (Zhou et al., 2019; Zhang et al., 2020). Thus it is intuitive to verify the effectiveness of our approach over different sequence lengths. Considering the validation set of WMT14 EN-DE (3k) is too small to cover scenarios with various sentence lengths, we randomly select 10k training data with lengths from 10 to 100. As shown in Figure 5, our approach consistently outperform the Transformer$_{base}$ model at different sequence lengths. Moreover, the improvements of our approach over the Transformer$_{base}$ is gradually increasing with sentence lengths. Specifically, we observe more than 1.0 BLEU improvements when sentence lengths in $[80, 100]$.

5.5 Model Convergence

As aforementioned, our confidence-aware scheduled sampling learns to deal with the exposure bias problem in an efficient manner, thus speeding up the model convergence. As shown in Figure 6, it costs the Transformer$_{base}$ 245k steps to converge to a local optimum (about 27.1 BLEU). To achieve the same performance, it only costs our confidence-aware scheduled sampling 80k step, namely about 3.0$\times$ speed up over the Transformer$_{base}$ and 1.8$\times$ speed up over the vanilla scheduled sampling. Since vanilla scheduled sampling randomly exposes more difficult predicted tokens for each token position, regardless of the actual model competence, its convergence speed is restricted to a certain extent. On the contrary, our approach samples predicted tokens only if the current model is capable of dealing with these more difficult inputs, mimicking the learning process of humans. Therefore, our approach is trained more efficiently.

6 Related Work

Confidence-aware Learning for NMT. As to confidence estimations for NMT, Zoph et al. (2015) frame translation as a compression game and measure the amount of information added by transla-

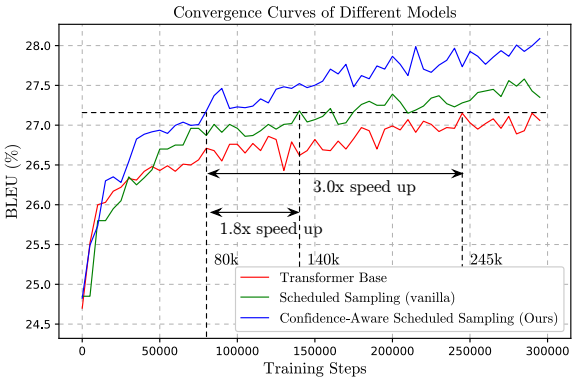

Figure 6: Convergence curves for different models. BLUE scores (%) are calculated on the validation set of WMT14 EN-DE. Our approach can achieve the same performance as the Transformer$_{base}$ with about 3.0× speed up.

tors. Wang et al. (2019b) propose to quantify the confidence of NMT model predictions based on model uncertainty, which is widely extend to select training samples (Jiao et al., 2020; Dou et al., 2020), to design confidence-aware curriculum learning (Zhou et al., 2020; Wan et al., 2020), and to augment synthetic corpora (Wei et al., 2020). Model confidence is also served as a useful metric for analyze NMT model from the perspective of fitting and search (Ott et al., 2018), visualization (Rikters et al., 2017) and calibration (Kumar and Sarawagi, 2019; Wang et al., 2020). Different from existing studies, we are the first to propose confidence-aware scheduled sampling for alleviating the exposure bias problem in NMT.

7 Conclusion

In this paper, we propose confidence-aware scheduled sampling for NMT, which exactly samples corresponding tokens according to the real-time model competence rather than human intuitions. We further explore to sample more noisy tokens for high-confidence token positions, preventing scheduled sampling from degenerating into the original teacher forcing mode. Experiments on three large-scale WMT translation tasks suggest that our approach improves vanilla scheduled sampling both translation quality and convergence speed. We elaborately analyze the effectiveness and efficiency of our approach from multiple aspects. As a result, we further observe our approaches: 1) can alleviate the performance bottleneck of decoders for NMT to a certain extend; 2) improve the translation quality of long sequences.

Acknowledgments

The research work descried in this paper has been supported by the National Key R&D Program of China (2020AAA0108001) and the National Nature Science Foundation of China (No. 61976015, 61976016, 61876198 and 61370130). The authors would like to thank the anonymous reviewers for their valuable comments and suggestions to improve this paper.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *Advances in neural information processing systems*.

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.

Wray L. Buntine and A. Weigend. 1991. Bayesian back-propagation. *Complex Syst.*, 5.

Tobias Domhan. 2018. How much attention do you need? a granular analysis of neural machine translation architectures. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1799–1808.

Zi-Yi Dou, Antonios Anastasopoulos, and Graham Neubig. 2020. Dynamic data selection and weighting for iterative back-translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5894–5904.

Daniel Duckworth, Arvind Neelakantan, Ben Goodrich, Lukasz Kaiser, and Samy Bengio. 2019. Parallel scheduled sampling. *arXiv preprint arXiv:1906.04331*.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *international conference on machine learning*, pages 1050–1059. PMLR.

Ian Goodfellow, Yoshua Bengio, and Aaron Courville. 2016. *Deep Learning*. MIT Press. http://www.deeplearningbook.org.

Sebastian Goodman, Nan Ding, and Radu Soricut. 2020. TeaForN: Teacher-forcing with n-grams. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8704–8717, Online. Association for Computational Linguistics.

Kartik Goyal, Chris Dyer, and Taylor Berg-Kirkpatrick. 2017. Differentiable scheduled sampling for credit assignment. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 366–371.

Tianyu He, Xu Tan, and Tao Qin. 2019. Hard but robust, easy but sensitive: How encoder and decoder perform in neural machine translation. *arXiv preprint arXiv:1908.06259*.

Wenxiang Jiao, Xing Wang, Shilin He, Irwin King, Michael Lyu, and Zhaopeng Tu. 2020. Data rejuvenation: Exploiting inactive training examples for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2255–2266.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Aviral Kumar and Sunita Sarawagi. 2019. Calibration of encoder decoder models for neural machine translation. *arXiv preprint arXiv:1903.00802*.

Hao Liu, Yihao Feng, Yi Mao, Dengyong Zhou, Jian Peng, and Qiang Liu. 2018. Action-dependent control variates for policy optimization via stein identity. In *International Conference on Learning Representations*.

Fandong Meng, Jianhao Yan, Yijin Liu, Yuan Gao, Xianfeng Zeng, Qinsong Zeng, Peng Li, Ming Chen, Jie Zhou, Sifan Liu, and Hao Zhou. 2020. Wechat neural machine translation systems for wmt20. In *Proceedings of the Fifth Conference on Machine Translation*, pages 238–246, Online. Association for Computational Linguistics.

Tsvetomila Mihaylova and André F. T. Martins. 2019. Scheduled sampling for transformers. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics: Student Research Workshop*, pages 351–356, Florence, Italy. Association for Computational Linguistics.

R. Müller, Simon Kornblith, and Geoffrey E. Hinton. 2019. When does label smoothing help? In *NeurIPS*.

Radford M Neal. 2012. *Bayesian learning for neural networks*, volume 118. Springer Science & Business Media.

Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Analyzing uncertainty in neural machine translation. In *ICML*.

Marc'Aurelio Ranzato, Sumit Chopra, Michael Auli, and Wojciech Zaremba. 2016. Sequence level training with recurrent neural networks. In *ICLR*.

Steven J Rennie, Etienne Marcheret, Youssef Mroueh, Jerret Ross, and Vaibhava Goel. 2017. Self-critical sequence training for image captioning. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 7008–7024.

Matīss Rikters, Mark Fishel, and Ondřej Bojar. 2017. Visualizing neural machine translation attention and confidence. *The Prague Bulletin of Mathematical Linguistics*, 109(1):39–50.

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.

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)*, pages 1683–1692, Berlin, Germany. Association for Computational Linguistics.

Kaitao Song, Xu Tan, and Jianfeng Lu. 2020. Neural machine translation with error correction. In *IJCAI*.

Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. 2014. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

Richard Stuart Sutton. 1984. Temporal credit assignment in reinforcement learning.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008.

Yu Wan, Baosong Yang, Derek F. Wong, Yikai Zhou, Lidia S. Chao, Haibo Zhang, and Boxing Chen. 2020. Self-paced learning for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Qiang Wang, Bei Li, Tong Xiao, Jingbo Zhu, Changliang Li, Derek F Wong, and Lidia S Chao. 2019a. Learning deep transformer models for machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1810–1822.

Shuo Wang, Yang Liu, Chao Wang, Huanbo Luan, and Maosong Sun. 2019b. Improving back-translation with uncertainty-based confidence estimation. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 791–802, Hong Kong, China. Association for Computational Linguistics.

Shuo Wang, Zhaopeng Tu, Shuming Shi, and Yang Liu. 2020. On the inference calibration of neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 3070–3079, Online. Association for Computational Linguistics.

Xin Wang, Wenhu Chen, Yuan-Fang Wang, and William Yang Wang. 2018. No metrics are perfect: Adversarial reward learning for visual storytelling. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 899–909.

Xiangpeng Wei, Heng Yu, Yue Hu, Rongxiang Weng, Luxi Xing, and Weihua Luo. 2020. Uncertainty-aware semantic augmentation for neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2724–2735.

Jiacheng Zhang, Yanzhuo Ding, Shiqi Shen, Yong Cheng, Maosong Sun, Huanbo Luan, and Yang Liu. 2017. Thumt: An open source toolkit for neural machine translation. *arXiv preprint arXiv:1706.06415*.

Jiajun Zhang, Long Zhou, Yang Zhao, and Chengqing Zong. 2020. Synchronous bidirectional inference for neural sequence generation. *Artificial Intelligence*, 281:103234.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019. Bridging the gap between training and inference for neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4334–4343, Florence, Italy. Association for Computational Linguistics.

Long Zhou, Jiajun Zhang, and Chengqing Zong. 2019. Synchronous bidirectional neural machine translation. *Transactions of the Association for Computational Linguistics*, 7:91–105.

Yikai Zhou, Baosong Yang, Derek F Wong, Yu Wan, and Lidia S Chao. 2020. Uncertainty-aware curriculum learning for neural machine translation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 6934–6944.

Barret Zoph, Marjan Ghazvininejad, and Kevin Knight. 2015. How much information does a human translator add to the original? In *Proceedings of the 2015 Conference on Empirical Methods in Natural Language Processing*, pages 889–898, Lisbon, Portugal. Association for Computational Linguistics.