Modeling Future for Neural Machine Translation by Fusing Target Information

Anonymous ACL submission

Abstract

Sequence-to-sequence Neural Machine Translation (NMT) models have achieved excellent performance. However, the NMT decoder only makes predictions based on the source and the target historical context, ignores the target future information completely, leading to a problem that NMT does not consider potential future information when making decisions. To alleviate this problem, we propose a simple and effective **Fu**ture-fused **NMT** model called FuNMT, which introduces a reverse decoder to explicitly model the target future information, then adopts an agreement mechanism to enable the forward decoder to learn this future information. Empirical studies on multiple benchmarks show that our proposed model significantly improves translation quality.

Source-1	wǒ shì jīngcháng zuò zhè chē
Reference	I often take this train .
Trans.Big	I often sit in this car .
Source-2	shípǐn ānquán jì yìngyòng yíngyǎng zhōngxīn shì měiguó wèishēng yǔ gōngzhòng fúwùbù zhīxià de zhèngfǔ jīguān .
Reference	cfsan is a government body formed under the united states department of health and human services .
Trans.Big	the centre is a government agency under the us department of health and public service .

Table 1: Translation examples showing the insufficiency of translating source sentences based only on the source context and partially-generated translations.

1 Introduction

Recently, NMT models (Sutskever et al., 2014; Bahdanau et al., 2014; Wu et al., 2016; Gehring et al., 2017; Vaswani et al., 2017) have become the most widely-used models and occupying a dominant position. NMT models treat the Machine Translation (MT) as a sequence-to-sequence task and apply an encoder-decoder framework. The decoder predicts the translation word by word from left to right, all tokens after each step in the target sequence are masked to ensure the autoregressiveness (Gu et al., 2018) of the decoder during training. However, given the bilingual parallel corpus, it is obvious that the target future information is observable for each training step. The autoregressiveness makes the NMT model totally ignore the explicit future information of the target sequence in the training set. Moreover, although the source sentence already contains all the semantic information, it may not be enough to make predictions only using the source context and the partial translation as the conditions (Zhang et al., 2019; Duan et al., 2020). Sometimes the potential future information of the target sequence also needs to be considered. To show the importance of unseen information in the target sequence, we select two examples from the training set, as shown in Table 1. Even if the entire source sentence **Source-1** in the first example is exposed to the Transformer-big [1] (denoted as **Trans.Big**), the word “zuò” is incorrectly translated as “sit”, which resulted in improper subsequent translation “sit in this car” (red part). When the model predict the third token, the unseen token “train” in the target sequence should also be considered to help generate commonly used combinations “take $\cdots$ train”. Similarly, for the second example **Source-2**, the NMT model incorrectly translates the blue part into “the centre” and lost lots of information. We claim that the above problem stems from the fact that the model lacks ability to control the global situation when making predictions at each step. It tends to select high-frequency tokens in the training set, but these tokens may have a negative impact on the generation of subsequent translations. To mitigate this problem, the most intuitive approach is to predict the translation based

[1] The model is trained on the NIST Chinese-English dataset with model average

on the context of the entire source and target sequences. But this will make the training fall into a non-autoregressive mode and lead to messy translations at inference, which is obviously unfeasible.

In this paper, we propose a simple and effective framework that can explicitly model the entire future information of the target sequence. Concretely, we introduce a reverse decoder to perform left-masked self-attention so that the representation learned by each step contains target future information. A future agreement mechanism is designed to integrate the learned future context into the Transformer decoder, so that the decoder could leverage the potential future information at inference. The effect is that although the model chooses a word with relatively small short-term benefits when making a decision at a certain step, the global benefits brought by it will be relatively large. The proposed model mainly contributes to making the Transformer encoder integrate the target future information captured by an introduced reverse decoder, without affecting the translation efficiency at inference. Compared with related works, our model can achieve a better trade-off between inference efficiency and translation performance. Empirical experiments show that our proposed model can significantly outperform the strong baseline models and related models.

2 Background

Our method can be plugged into most sequence-to-sequence frameworks. Without loss of generality, we take the Transformer model as an example to introduce our method. Assuming that one of the sentence pairs in the training set consists of the source sequence $\mathbf{x}$ and the observed translation $\mathbf{y}^*$

$$\mathbf{x} = \left\{x_1, \cdots, x_{|\mathbf{x}|}\right\}; \;\; \mathbf{y}^* = \left\{y_1^*, \cdots, y_{|\mathbf{y}^*|}^*\right\} \qquad (1)$$

Encoder In each layer of the L stacked same layers, the output of Self-Attention sub-layer ($\mathbf{SAtt}$) is fed into the feed-forward sub-layer ($\mathbf{FFN}$) [2]. The output of each sub-layer can be formatted as $\mathbf{LN}(x + \text{sublayer}(x))$, where $\mathbf{LN}(\cdot)$ is Layer Normalization (Ba et al., 2016) and $+$ means the Residual Connection.

$$h^l = \mathbf{LN}\left(h^{l-1} + \mathbf{SAtt}\left(h^{l-1}, h^{l-1}, h^{l-1}\right)\right); \qquad (2)$$

$$h^l = \mathbf{LN}\left(h^l + \mathbf{FFN}\left(h^l\right)\right) \qquad (3)$$

Note that $\mathbf{x}$ with position encoding is used as h^0 and h^l means the output of the l^{th} layer.

Decoder In each layer of another L stacked same layers, besides the two sub-layers applied in each encoder layer, a Cross-Attention sub-layer ($\mathbf{CAtt}$) is employed to extract source information (called *context vector*). Assuming that s_j^l represents the encoding of the j^{th} word in the l^{th} layer.

$$s_j^l = \mathbf{LN}\left(s_{<j}^{l-1} + \mathbf{SAtt}\left(s_{<j}^{l-1}, s_{<j}^{l-1}, s_{<j}^{l-1}\right)\right); \qquad (4)$$

$$s_j^l = \mathbf{LN}\left(s_j^l + \mathbf{FFN}\left(s_j^l\right)\right); \qquad (5)$$

$$s_j^l = \mathbf{LN}\left(s_j^l + \mathbf{CAtt}\left(s_j^l, h^L, h^L\right)\right) \qquad (6)$$

Due to the autoregressive property, at the j^{th} step, the Self-Attention sub-layer only attends to all previous positions. For convenience, we express the calculation of the multilayer decoder as

$$s_j^{l_2} = Dec^{l_1 \to l_2}\left(s_{<j}^{l_1}, h^L, h^L\right) \qquad (7)$$

where $l_2 > l_1$, the calculation for each layer is the same as Eq. 5$\sim$6 and $s_j^{l_2}$ represents the hidden state output by the l_2^{th} layer at the j^{th} step. The word probability distribution P_j over all the words in the target vocabulary is estimated conditioned on s_j^L.

$$P_j = \text{softmax}\left(\mathbf{W}_s s_j^L + \mathbf{b}_s\right) \qquad (8)$$

where the trainable parameters $\mathbf{W}_s$ and $\mathbf{b}_s$ map s_j^L to a vector with the size of vocabulary.

Training The objective is to maximize the probability of the ground truth sequence by Maximum Likelihood Estimation (MLE)

$$\mathcal{L}(\theta) = -\sum\nolimits_{j=1}^{|\mathbf{y}^*|} \log P_j[y_j^*] \qquad (9)$$

where $|\mathbf{y}^*|$ indicates the length of the ground truth translation $\mathbf{y}^*$, $P_j[y_j^*]$ is the predicted probability of generating the golden word y_j^* at the j^{th} step, θ represents all trainable parameters related to the naive translation model.

3 Proposed Model

The proposed FuNMT consists of three modules: **F**uture-**m**atched **Decoder** (**FmDecoder**), **R**everse **Decoder** (**RDecoder**) and Future-**f**used **Decoder** (**FfDecoder**). The Transformer decoder is composed of **FmDecoder** and **FfDecoder**.

[2] Refer to Vaswani et al. (2017) for the details about $\mathbf{SAtt}$ and $\mathbf{FFN}$, we omit dropout for convenience.

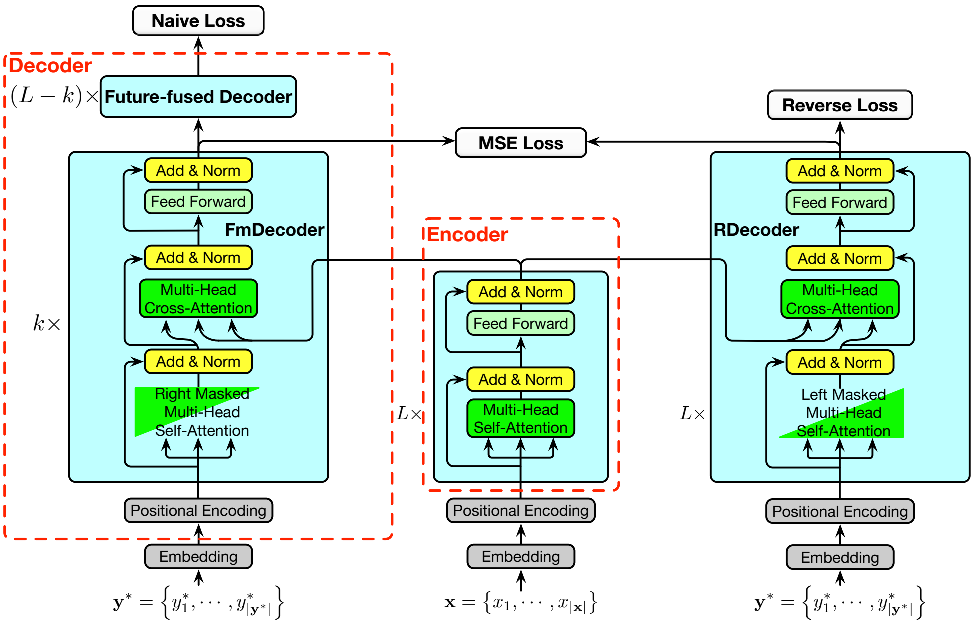

Figure 1: The architecture diagram of our proposed FUNMT model. "Add" and "Norm" represent Residual Connection and Layer Normalization respectively, and other terms are referred in the model section.

3.1 Future-matched Decoder

We delicately design the FmDecoder so that the Transformer model has the ability to perceive the future context in the target sequence, while ensuring the autoregressiveness and decoding efficiency of the decoder. Assuming that both naive Transformer encoder and decoder [3] have L same layers, we use the first k layers as the FmDecoder.

According to the Eq. 7, the hidden state output by FmDecoder at the j^{th} step can be calculated as:

$$f_j^k = Dec^{0\to k}\left(f_{\leq j}^0, h^L, h^L\right) \tag{10}$$

Consistent with the naive Transformer decoder, at the j^{th} step, FmDecoder performs Right Masked Multi-Head Self-Attention with all subsequent tokens ($\mathbf{y}_{>j}^*$) masked, as shown in the green upper triangular matrix in the left box of Figure 1. The input $f_{\leq j}^0$ of the FmDecoder is the embedding of $\mathbf{y}^*$ with positional encoding. FmDecoder aims to fit the future context information explicitly modelled by Reverse Decoder through a Future Agreement mechanism. Next, we elaborate on the Reverse Decoder and the Future Agreement mechanism.

3.2 Reverse Decoder

Different from FmDecoder, RDecoder executes Left Masked Multi-Head Self-Attention and masks all previous tokens ($\mathbf{y}_{<j}^*$) out, as shown in the green lower triangular matrix in the right box of Figure 1. The j^{th} token is only associated with all subsequent tokens ($\mathbf{y}_{>j}^*$), which therefore represents the contextual information of the future.

$$r_j^L = \mathbf{LN}\left(r_{\geq j}^{L-1} + \mathbf{SAtt}\left(r_{>j}^{L-1}, r_{>j}^{L-1}, r_{>j}^{L-1}\right)\right) \tag{11}$$

$$r_j^L = \mathbf{LN}\left(r_j^L + \mathbf{FNN}(r_j^L)\right) \tag{12}$$

$$r_j^L = \mathbf{LN}\left(r_j^L + \mathbf{CAtt}\left(r_j^L, h^L, h^L\right)\right) \tag{13}$$

where L denotes the layers number of the RDecoder. r_j^L, the hidden state produced by the last layer, can be reasonably regarded as the *future context* including information of $\mathbf{y}_{>j}^*$.

RDecoder is also optimized to predict the ground truth target sequence. The calculation method of Reverse Loss is similar to Eq. 8 and Eq. 9:

$$P_j^r = \text{softmax}\left(\mathbf{W}_r r_j^L + \mathbf{b}_r\right) \tag{14}$$

$$\mathcal{L}\left(\theta^r\right) = -\sum_{j=1}^{|\mathbf{y}^*|} \log P_j^r[y_j^*] \tag{15}$$

where the trainable parameters $\mathbf{W}_r$ and $\mathbf{b}_r$ map r_j^L to a vector with the size of vocabulary. P_j^r is the probability distribution of the j^{th} token predicted by RDecoder.

3.3 Future Agreement

Inspired by the work of Liu et al. (2017) which minimizes the Mean Squared Error (MSE) between the hidden vectors produced by the decoder and the future vectors on the Language Model (LM) task to make the decoder consider the future information,

[3] Here we refer to the original Transformer encoder and decoder as naive Transformer encoder and decoder respectively.

we analogously minimize the MSE between the hidden state f_j^k output by FmDecoder and the *future context* r_j^L produced by RDecoder to let FmDecoder fuse the future information. The MSE loss is defined as:

$$\mathcal{L}\left(\theta^a\right) = \frac{1}{m} \sum\nolimits_{j=1}^{|\mathbf{y}^*|} \left\| f_j^k - r_j^L \right\|^2 \qquad (16)$$

where m is the dimension of *future context* vector.

3.4 Future-fused Decoder

Except for the input and the number of layers, the FfDecoder module is exactly the same as the naive Transformer decoder. Both autoregressively encode the target sequence from left to right, which means that only the information before each token is considered. In our model, we only use the last $L-k$ layers of the naive Transformer decoder as FfDecoder. We define the input of FfDecoder as $\mathbf{s}^k$. Most directly, we take the output of the FmDecoder as the input of the FfDecoder ($\mathbf{s}^k = \mathbf{f}^k$). As mentioned earlier, because we minimize the MSE optimization objective to make the output of the FmDecoder match the *future context*, although the *future context* information is actually inaccessible at inference, our FmDecoder has approximately learned the potential future information in the training set.

3.5 Training Objective

We finally minimize the following loss function:

$$\mathcal{L}(\theta) = (1-\lambda_r) * \mathcal{L}\left(\theta^f\right) + \lambda_r * \mathcal{L}\left(\theta^r\right) + \lambda_a * \mathcal{L}\left(\theta^a\right) \quad (17)$$

where λ_r and λ_a are used to balance the three losses. In our experiments, we explore the hyper-parameters λ_r on the validation set. Besides, we design an on-the-fly phased training strategy to make the model fully leverage future information. In detail, the Transformer decoder and RDecoder are synchronously optimized from scratch in the first stage. When the training reaches the E^{th} epoch [4], the MSE constraint starts to take effect and affects the training, making FmDecoder start learning the future context. We design λ_a as follows:

$$\lambda_a = \begin{cases} 0 & if\ e_{idx} < \mathrm{E} \\ 1 & if\ e_{idx} \geq \mathrm{E} \end{cases}$$

where e_{idx} denotes the index of epoch.

4 Related Works

Lots of related works conduct different strategies to leverage future information, which are listed:

Future Modeling Liu et al. (2017) embedded the rest of the sequence into future vectors and incorporated these future vectors with the LSTM-based LM. Serdyuk et al. (2018) proposed to encourage generative RNNs to plan ahead and ease modeling of long-term dependencies by using twin networks. These works considered the future information on the tasks of LM, speech recognition and image capture respectively, and are not applicable to the NMT model. Zheng et al. (2018) and Zheng et al. (2019) modeled translated past contents and untranslated future contents on the source side for NMT model. Comparing to them, we introduce a different method to model the future information on the target side. Duan et al. (2020) estimated the future cost based on the current generated target word by previewing the translation cost of next target word at the current time-step. Our proposed method can model the entire target future information, not just the next one word.

Knowledge distillation Zhang et al. (2019) distilled the future information produced by R2L decoder through KL divergences and alleviated the error propagation problem during generation. Zhang et al. (2019) presented a future-aware knowledge distillation framework which enables the unidirectional decoder to explore the future context for word prediction. They also use the entire future information, but either the model is not robust enough, or two decoders need to be retained at inference, resulting in a worse trade-off between translation performance and efficiency.

Bidirectional Decoding Zhang et al. (2018) equipped RNN-based encoder-decoder NMT framework with a backward decoder and fully leveraged the target-side context. Zhou et al. (2019) proposed a synchronous bidirectional NMT model that adopts one decoder to generate outputs with left-to-right and right-to-left directions simultaneously and interactively. Although these works improved translation performance, they need two-pass decoding directly or indirectly. While our method can achieve better translation quality without affecting translation efficiency.

Reinforcement Learning There are also a line of works to estimate a score representing future information through reinforcement learning to guide the prediction of the current step. Li et al. (2017)

[4] Additional experiments show that different E values have a weak effect on the translation results. We empirically set E as $1/7$ of the maximum epoch. Thus, strictly speaking, E is not a hyper-parameters.

Model	NIST Zh⇒En					
	MT03	MT04	MT05	MT06	MT08	Avg.
Transformer-base	45.29	45.31	45.18	44.31	35.39	43.10
+ FuNMT	**46.17***	**47.19**†	**46.58**†	**45.55**†	**36.35**	**44.37**
Transformer-base + ensemble	45.83	46.41	46.72	45.86	36.75	44.31
+ FuNMT	**47.21***	**48.14**†	**47.77**	**46.95***	**38.06**†	**45.63**
Transformer-big	46.88	46.63	46.60	45.69	37.36	44.63
+ FuNMT	**48.46**†	**48.31**†	**48.95**†	**46.91**†	**38.46**†	**46.22**
Transformer-big + ensemble	46.82	47.61	47.94	46.73	38.18	45.46
+ FuNMT	**49.17**†	**48.65**†	**49.14***	**47.89**	**39.39**†	**46.85**

Table 2: Translation performance of different models on the NIST Zh⇒En translation task. "$*$" and "$\dagger$" indicate statistically significant difference with $p<0.05$ and $p<0.01$ from Transformer respectively.

introduced a simple actor-critic model, where the actor employed the MLE-based token generation policy and the critic acted as a value function that estimates the future value of the desired property for decision making. Bahdanau et al. (2016) introduced a critic network that is trained to predict the value of an output token, given the policy of an actor network. He et al. (2017) developed a new decoding scheme for NMT, which considers not only the local conditional probability of a candidate word, but also its long-term reward for future decoding predicted by a proposed value network. Generally, equipped with RL, the RNN-based NMT model is difficult to optimize and the performance improvement is limited. In addition, it is impractical to apply RL to attention-based Transformer, but our method is not limited to model architecture.

5 Experiments

Datasets For the small-scale scenario, we choose IWSLT'14 German⇒English (De⇒En) and ASPEC Chinese⇒Japanese (Zh⇒Jp) translation task, which contain 160K and 672K sentence pairs. We follow Edunov et al. (2018) and Nakazawa et al. (2016) to do data splitting. We employ Byte Pair Encoding (BPE) (Sennrich et al., 2016) model jointly learned using 10k and 30k merging operations for the two language pairs.

For the middle-scale scenario, we use NIST Chinese⇒English (Zh⇒En), WMT'14 English⇒German (En⇒De) and WMT'17 English⇒German (En⇒De), containing 1.25M, 3.9M and 5.2M training samples respectively. Sentences are encoded using BPE with 32k, 37k and 40k joint merging operations respectively.†

For the large-scale scenario, we use the WMT'14 English⇒French (En⇒Fr) dataset with 35.8M training samples. BPE model is jointly learned using 40k merging operations to generate subwords. All datasets except the Zh⇒En are publicly available. For Zh⇒En, the training set is mainly extracted from LDC corpora, and we use the NIST 2002 (MT02) test set as the validation set.

Setting Our implementation is based on *fairseq*. The setting *transformer_iwslt_de_en* is used for both De⇒En and Zh⇒Jp tasks. For Zh⇒En and WMT'14 En⇒De tasks, *transformer_wmt_en_de* and *transformer_vaswani_wmt_en_de_big* are used for base and big settings. For En⇒Fr, *transformer_vaswani_wmt_en_fr_big* setting is applied. More details about data and model are described in Appendix 6

Evaluation Metrics We measure the translation quality with 4-gram BLEU scores (Papineni et al., 2002). For IWSLT'14 De⇒En, case-insensitive BLEU score is calculated by *multi-bleu.pl*. For NIST Zh⇒En, we employ four raw references and compute the case-insensitive BLEU with SacreBLEU [5] (Post, 2018). We compute the case-sensitive tokenized BLEU for WMT'14 En⇒De and En⇒Fr. For WMT'17 En⇒De translation task, we do not tokenize the references and calculate the case-sensitive BLEU with SacreBLEU [6]. To ensure comparability, we keep the evaluation metrics consistent with the previous works.

5.1 Translation Performance

Different Model Architectures We verified the effect of our model on the Transformer base and big settings on the Zh⇒En datasets. To futher make the conclusion convincing, we also explore the impact of the single models and averaged models (+ ensemble) for both settings. As shown in Table 2,

[5] `BLEU+case.mixed+lang.zh-en+numrefs.4+smooth.exp+tok.13a+version.1.4.4`

[6] `BLEU+case.mixed+lang.en-de+numrefs.1+smooth.exp+tok.13a+version.1.4.4`

Model	WMT14		WMT17	Speed	
	En⇒De	*En⇒Fr*	*En⇒De*	*Train*	*Inference*
Reproduced Transformer (Vaswani et al., 2017)	28.68	43.35	27.88	-	-
+asynchronous bidirectional (Zhang et al., 2018)	28.22	-	-	-43.0%	-63.9%
+future-aware KD (Zhang et al., 2019)	29.42		28.80	-48.8%	-52.4%
+synchronous bidirectional (Zhou et al., 2019)	29.21	-	-	-39.1%	-10.5%
+next word prediction (Duan et al., 2020)	29.12	42.02	-	-7.9%	-0.0%
+our work	**29.73**†	**43.65**	**28.92**†	-45.4%	-0.0%

Table 3: Comparison with existing works of future modeling on the benchmarking datasets. The value in the speed column represents the percentage of the drop in training and infer speed compared to the Transformer-big model. "†" indicates statistically significant difference with p<0.01 from Transformer.

Model	De⇒En	Zh⇒Jp
Transformer	34.32	49.40
+ FuNMT	**35.32**	**50.06**

Table 4: Translation performance on small-scale IWSLT14 De⇒En and ASPEC Zh⇒Jp datasets.

for Transformer-base, in both cases of using only single model and model average, our model can bring an average improvement of 1.3 BLEU scores to the baseline model on all test sets.

A similar situation occurs on the Transformer-big setting. Our proposed single model brings an average improvement of 1.6 BLEU scores on all Zh⇒En test sets compared with the baseline model. When equipped with the model average technique, FuNMT can outperform the baseline system by an average of 1.4 BLEU points on all test sets. It can also be observed from the Table 2 that in any case, our proposed model can significantly and steadily improve the baseline model on most test sets.

Small-Scale Datasets In order to further prove the effectiveness of our proposed method, we conduct experiments on two other small-scale datasets. Our proposed method also improves the baseline model by 1.0 BLEU scores on the IWSLT'14 De⇒En test set, as shown in Table 4.

Most languages have a subject-verb-object (SVO) syntactic structure, while the most significant feature of Japanese is the post-predicate, which is the syntactic structure of the subject-object-verb (SOV). In view of this, we assume that Japanese has a strong long-distance dependence, and the generation of Japanese is more dependent on future information. In order to verify whether our method is helpful for the translation whose target language has the SOV syntactic structure, we adopt the ASPEC Zh⇒Jp translation task whose target language is Japanese. As shown in Table 4, our method has an improvement of 0.7 BLEU scores on the baseline model.

Comparison with Existing Work In order to make a fair comparison with the other two related works, we also trained Transformer-base model on the WMT'17 En⇒De dataset, which is exactly the same as that reported in the two related works. All results are listed in Table 3. It can be seen from Table 3 that our proposed FuNMT improves the baselines by 1.04 and 0.3 BLEU points on the WMT'14 En⇒De and En⇒Fr test sets, respectively.

Our work has similar training efficiency to the asynchronous bidirectional work (Zhang et al., 2018). However, the two-way decoding results in a decrease in translation speed [7] of 63.9%. Our proposed approach has significant advantages in translation performance and efficiency.

Although the method of future-aware KL also makes full use of the target future information and has a significant improvement in translation effect over the baseline system on the WMT'17 test set, their proposed method requires two decoders at inference, so the efficiency of training and decoding is about half reduced. The training efficiency of future-aware KL and the performance on the WMT'17 test set are both comparable to our method, while our method has obvious advantages in inference efficiency compared with it.

Although the training efficiency has slightly decreased, the translation quality of our proposed FuNMT is 0.72 BLEU higher than the synchronous bidirectional NMT model on the WMT'14 En⇒De test set, and the translation efficiency is also superior to synchronous bidirectional NMT model.

Compared with the method of next word prediction, our training efficiency has no advantage. But Duan et al. (2020) only considers the next one word when predicting the translation, so their work

[7] We compare the reduction of the training/testing speed of the methods relative to the baseline systems, so even if the computing environment is different, we claim that the comparison is fair.

Models	**#mistranslated**	**#missed**
Transformer-big	27	46
FUNMT	15	32

Table 5: Statistics of mistranslated and under-translated words on all NIST Zh⇒En test sets.

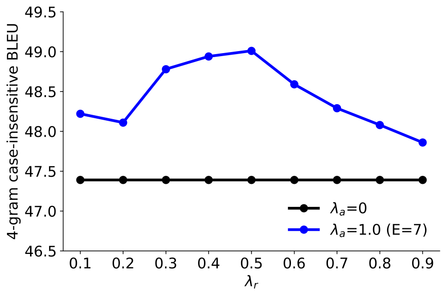

Figure 2: Transformer vs. FUNMT on NIST Zh⇒En validation set for different λ_r values.

is not able to utilize the future information fully. Although it does not affect the efficiency of inference, the performance of translation is 0.71 and 1.63 BLEU points lower than FUNMT on WMT'14 En⇒De and En⇒Fr respectively.

Briefly, our proposed FUNMT can outperform all related works on all medium-scale and large-scale datasets. Although FUNMT incorporates a additional RDecoder during the training process, which reduces the training efficiency by 45.4% , RDecoder is not needed at inference, so FUNMT has no effect on the inference speed.

Human Evaluation We conduct human evaluation on the NIST Zh⇒En test set. We first merge all the test sets MT03-08, and then select 50 sentences from them to form 50 triples (S, T_a, T_b), where T_a and T_b represent the translation generated by the baseline and FUNMT respectively. We let people who are proficient in English count the number of mistranslated and under-translated words in source according to T_a and T_b, as shown in Table 5. It can be seen that there are fewer words incorrectly translated or ignored by FUNMT, which means that the fusion of future information can alleviate the phenomenon of mistranslations and under-translations.

5.2 Ablation Study

We perform all ablation experiments on the NIST Zh⇒En validation set.

Hyper-parameters λ_r We first investigate the impact of different values of λ_r on the validation set. As shown in Table 2, when λ_r is equal to 0.5, the BLEU score on the validation set reaches the maximum. The results are intuitive. When the forward decoder and the backward decoder are trained in a balanced manner, FmDecoder can make full use of the future information of the target sequence.

RDecoder	**DropNet**	**Constraint**	**BLEU**
✓	×	MSE	49.01
✓	✓	MSE	48.57
✓	×	KL	48.98

Table 6: Ablation study of the RDecoder, DropNet and Constraint on NIST Zh⇒En MT02 validation set.

DropNet Since the number of parameters of the Transformer decoder remains invariant, we suspect that the decoder's representation ability is not enough that the representation capacity of the decoder is not enough to learn the target-side historical and future information simultaneously. In view of this, we fuse $\mathbf{y}$ and $\mathbf{f}^k$ based on DropNet (Zhu et al., 2020) to verify whether the model can be further improved. It can be observed from Table 6 that, unfortunately, DropNet does not work in our scenario, but instead reduces the translation performance by about 0.5 BLEU points. In our experiments, we do not conduct the DropNet-based fusion strategy.

KL vs MSE KL divergence has been proven effective as a measure of the similarity between two probability distributions (Zhang et al., 2019; Feng et al., 2020). We also try to replace the MSE constraint in the Future Agreement module with KL divergence, which means the Eq. 16 is updated to:

$$\begin{aligned}\mathcal{L}\left(\theta^{KL}\right) = \sum\nolimits_{j=1}^{|\mathbf{y}^*|} KL\Big\{&\text{softmax}\left(\mathbf{W}_f f_j^k\right) \\ &|| \text{ softmax}\left(\mathbf{W}_r r_j^L\right)\Big\}\end{aligned} \quad (18)$$

where the trainable parameters $\mathbf{W}_f$ and $\mathbf{W}_r$ are used to map f_j^k and r_j^L to vectors with the size of vocabulary. After that, we observe the impact of different constraints on translation performance. Comparing the first and third row in Table 6, we observe that in our scenario, KL divergence does not bring benefits, and the results obtained are very close to the MSE constraints. For all experiments reported in our work, we use MSE constraints.

k	$L-k$	**BLEU**
3	3	46.57
4	2	46.56
5	1	46.51

Table 7: Comparison between different layers of FmDecoder on the average BLEU of all Zh⇒En test sets.

About k Value We also conduct experiments to explore the effect of different layers of FmDecoder

Source-1	tàiguó dāngjú pàijī cóng jiǎnpǔzhài chèlí tàiqiáo
Reference	thai authorities sent planes to evacuate thai nationals from cambodia
Trans.Big	thai authorities evacuate thai nationals from cambodia
FuNMT	thai authorities [**send plane to**] evacuate thai from cambodia
Source-2	Zài nóngcūn xiǎng gǎo diǎn wénhuà huódòng , zhǎo diǎn “ lè” zi tài nánle,
Reference	It is too difficult to organize some cultural activities , to find some fun in rural areas .
Trans.Big	It is too difficult to find some " music " ; in rural areas .
FuNMT	It is too difficult to [**carry out some cultural activities**] in the rural areas .
Source-3	nàijílìyà zhèngfǔ zhèng jiāqiáng gōngzuò , zǔzhǐ zài fēizhōu dàlù de bìngdú chuánrǎn gěi rénlèi .
Reference	the nigerian government is stepping up efforts to prevent the virus on the african continent from spreading to humans .
Trans.Big	the nigerian government is working harder to prevent the virus from spreading to human beings in africa .
FuNMT	the nigerian government is stepping up efforts to stop the spread of [**the virus across the african continent**] to humans .
Source-4	cháoxiǎn bàndǎo yú yījiǔsìbānián fēnliè chéngwéi shíháng zīběnzhǔyì de nánhán yǔ gòngchǎnzhǔyì de běihán , shuāngfāng céng zài yījiǔwǔ língnián zhì yījiǔwǔsānnián de hánzhàn shíqī xiānghù díduì .
Reference	in 1948 , the korean peninsula was split into capitalist south korea and communist north korea. the two sides engaged in hostile conflict during the 1950-1953 korean war .
Trans.Big	in 1948 , the korean peninsula split into a capitalist north korea , where the two sides were hostile to each other during the korean war from 1950 to 1953 .
FuNMT	the korean peninsula [**was split into a capitalist south korea and communist north korea**] in 1948 , and the two sides hostile each other during the korean war from 1950 to 1953 .

Table 8: Translation examples.

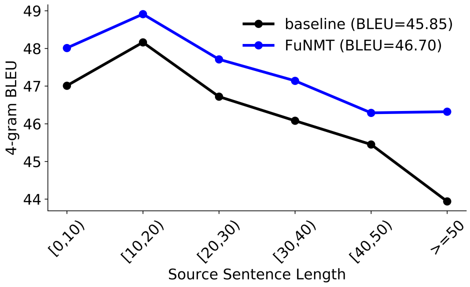

Figure 3: Comparison of the translation performance of Transformer (black lines) and FuNMT (blue lines) on the NIST Zh⇒En translation tasks according to the length of different source sentences.

and FfDecoder on the results. The comprison results are shown in Table 7. Considering that FmDecoder needs to fit the information learnt by RDecoder, we increase the number of layers of FmDecoder and find that it has almost no effect on the translation performance.

5.3 Analysis

Sentence Length Intuitively, the generation of the translation is more sensitive to future information as the length of the source sentence increases. To explore the model's ability to translate long sentences, we conduct comparative experiments on the test set of the NIST Zh⇒En tasks. First, we merge all test sets of MT03-08, then divide the merged test set into different groups at intervals of length 10 according to the length of source sentences. Then Transformer and FuNMT translate each group separately with corresponding BLEU scores shown in Figure 3. It can be seen that FuNMT surpasses the baseline system in all length intervals, especially for long sentences. Since FuNMT has a “global view” when generating translations, it will try to make choices that maximize the benefits of the entire translation at each step, and long sentence translation benefits more from this.

Case Study We list four translation examples in Table 8. Compared with the baseline model, our model may either generate some seemingly incorrect translations in the early stages of translating a sentence, such as “send plane” in the first example, or generate some relatively uncommon expressions, such as “carry out” and “stepping up” in the second and third examples, or miss some translations, such as “in 1948” in the fourth example. But from a global perspective, translations generated by FuNMT are more faithful to the source sentence. This can be explained as that when generating implausible translations, our model takes into account the potential future information through the representation r_j^L output by RDecoder.

6 Conclusion

We propose a simple and effective model FuNMT that enables the NMT model to fuse potential future information when making decisions without loss of decoding efficiency. Experiments on multiple translation tasks show that FuNMT brings a significant improvement in translation quality.

References

Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E Hinton. 2016. Layer normalization. *arXiv preprint arXiv:1607.06450*.

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2016. An actor-critic algorithm for sequence prediction. *arXiv preprint arXiv:1607.07086*.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*.

Chaoqun Duan, Kehai Chen, Rui Wang, Masao Utiyama, Eiichiro Sumita, Conghui Zhu, and Tiejun Zhao. 2020. Modeling future cost for neural machine translation. *arXiv preprint arXiv:2002.12558*.

Sergey Edunov, Myle Ott, Michael Auli, David Grangier, and Marc'Aurelio Ranzato. 2018. Classical structured prediction losses for sequence to sequence learning. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 355–364, New Orleans, Louisiana. Association for Computational Linguistics.

Yang Feng, Wanying Xie, Shuhao Gu, Chenze Shao, Wen Zhang, Zhengxin Yang, and Dong Yu. 2020. Modeling fluency and faithfulness for diverse neural machine translation. In *Proceedings of the Thirty-Fourth AAAI Conference on Artificial Intelligence*, volume 34 of *AAAI'20*, pages 59–66. AAAI Press.

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.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. 2018. Non-autoregressive neural machine translation. In *International Conference on Learning Representations*.

Di He, Hanqing Lu, Yingce Xia, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2017. Decoding with value networks for neural machine translation. In *Advances in Neural Information Processing Systems*, volume 30, pages 178–187. Curran Associates, Inc.

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

Jiwei Li, Will Monroe, and Dan Jurafsky. 2017. Learning to decode for future success. *arXiv preprint arXiv:1701.06549*.

Q. Liu, Y. Qian, and K. Yu. 2017. Future vector enhanced lstm language model for lvcsr. In *2017 IEEE Automatic Speech Recognition and Understanding Workshop (ASRU)*, pages 104–110.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. Aspec: Asian scientific paper excerpt corpus. In *Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC 2016)*, pages 2204–2208, Portorož, Slovenia. European Language Resources Association (ELRA).

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

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers*, pages 186–191, Brussels, Belgium. Association for Computational Linguistics.

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

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.

Dmitriy Serdyuk, Nan Rosemary Ke, Alessandro Sordoni, Adam Trischler, Chris Pal, and Yoshua Bengio. 2018. Twin networks: Matching the future for sequence generation. In *International Conference on Learning Representations*.

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.

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.

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*.

B. Zhang, D. Xiong, J. Su, and J. Luo. 2019. Future-aware knowledge distillation for neural machine translation. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(12):2278–2287.

Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous bidirectional decoding for neural machine translation.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019. Regularizing neural machine translation by target-bidirectional agreement. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence*, AAAI'19, pages 443–450. AAAI Press.

Zaixiang Zheng, Shujian Huang, Zhaopeng Tu, Xin-Yu Dai, and Jiajun Chen. 2019. Dynamic past and future for neural machine translation. 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 931–941, Hong Kong, China. Association for Computational Linguistics.

Zaixiang Zheng, Hao Zhou, Shujian Huang, Lili Mou, Xinyu Dai, Jiajun Chen, and Zhaopeng Tu. 2018. Modeling past and future for neural machine translation. *Transactions of the Association for Computational Linguistics*, 6:145–157.

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

Jinhua Zhu, Yingce Xia, Lijun Wu, Di He, Tao Qin, Wengang Zhou, Houqiang Li, and Tieyan Liu. 2020. Incorporating bert into neural machine translation. In *International Conference on Learning Representations*.

We elaborate from three aspects.

.1 datasets

IWSLT'14 De⇒En The training set consists of 160K sentence pairs and we randomly select 7,283 samples from the training set as the validation set. We concatenated dev2010, dev2012, tst2010, tst2011 and tst2012 as the test set, which contain 6,750 sentences[8]. Byte-Pair Encodings (BPE) (Sennrich et al., 2016) model is jointly learned using 10K merging operations to encode the source and target sentences, generating a vocabulary of 10,151 tokens.

NIST Zh⇒En The training set consists of 1.25M sentence pairs extracted from LDC corpora[9]. BPE model is jointly learned using 32K merging operations to generate subwords, producing a vocabulary of 42,679 subwords. We tokenize Chinese and English by Stanford and Moses tokenizer respectively.

WMT'14 En⇒De The training set contains 3.9M sentence pairs[10]. newstest2013 and newstest2014 are used as the validation and test set, which contains 3,000 and 3,003 sentences respectively. Sentences are encoded using BPE with 37K joint merging operations. The vocabulary contains 40,727 tokens.

WMT'14 En⇒Fr The training set contains 35.8M sentence pairs[11]. 26,854 sentences are extracted from the training set as the development set to select the model. newstest2014 with 3,003 sentences are used as the test set. BPE model is jointly learned using 40K merging operations to encode the English and French sentences, producing a vocabulary of 44,511 subwords.

ASPEC Zh⇒Jp The training set of ASPEC-JC [12] (Nakazawa et al., 2016) is composed of 672,315 sentence pairs, the development set and test set contains 2,090 and 2,107 sentence pairs respectively. We use jieba [13] and MeCab [14] to segment Chinese and Japanese. Sentences are further segmented using BPE model with 30K merging operations for source and target languages separately. Data preprocessing produces vocabularies of 25,063 subwords for Chinese and 25,103 subwords for Japanese.

WMT'17 En⇒De The training set is also acquired by *prepare-wmt14en2de.sh* without parameter "--icml17", containing about 5.2M sentence pairs. newstest2013 and newstest2017 are used as the validation and test set.

Task	#GPUs	T	F	lr	M
IWSLT'14 De⇒En	4(P40)	15K	1	$5e$-4	150
ASPEC Zh⇒Jp	4(P40)	15K	1	$5e$-4	150
NIST Zh⇒En(base)	4(P40)	6144	2	$7e$-4	30
NIST Zh⇒En(big)	8(P40)	4096	3	$5e$-4	30
WMT'14 En⇒De(base)	8(V100)	6144	2	$7e$-4	80K
WMT'14 En⇒De(big)	8(V100)	6144	2	$5e$-4	200K
WMT'14 En⇒Fr(big)	8(V100)	6827	3	$5e$-4	150K
WMT'17 En⇒De(base)	8(V100)	12288	4	$1e$-3	150K

Table 9: Model settings on different translation tasks. "T" means batch size on single GPU, "F" means gradient accumulation times. "M" represents the maximum number of training epochs (150) or updates (80K).

.2 Model Settings

All other settings are default, except the settings listed in Table 9. Adam optimizer (Kingma and Ba, 2014) with β_1=0.9, β_2=0.98 and ϵ=1e-6 is employed. The learning rate is controlled based on the inverse square root of the update number. The learning rate is initialized to 1e-07, linearly increases to lr in the first 4000 updates, and then is decayed proportional to the number of updates. For De⇒En and Zh⇒Jp, we decode with a beam size of 5 and length penalty $\alpha = 0.6$, for WMT'17 En⇒De, beam size is set to 12 and $\alpha = 0.4$, and for all other tasks, beam size is 4 and $\alpha = 0.6$. We keep the latest 10 checkpoints, average the latest 5 and 10 checkpoints respectively, and then select the model with the largest BLEU score on the development set from the 12 checkpoints as our best model.

[8] We adopt the script https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-iwslt14.sh to download and preprocess the dataset, and follow previous works (Ranzato et al., 2016; Edunov et al., 2018) for data splitting.

[9] The sentence pairs are mainly extracted from LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06, we use the NIST 2002 (MT02) test set as the validation set, which has 878 sentences, and the NIST 2003 (MT03), NIST 2004 (MT04), NIST 2005 (MT05), NIST 2006 (MT06) and NIST 2008 (MT08) as the test sets, containing 919, 1,788, 1,082, 1,664 and 1,357 sentences respectively.

[10] We obtain the dataset by https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-wmt14en2de.sh

[11] We obtain the dataset by https://github.com/pytorch/fairseq/blob/master/examples/translation/prepare-wmt14en2fr.sh

[12] The dataset is described in http://orchid.kuee.kyoto-u.ac.jp/ASPEC/

[13] https://github.com/fxsjy/jieba

[14] https://pypi.org/project/mecab-python3/