Guiding Teacher Forcing with Seer Forcing for Neural Machine Translation

Yang Feng[1,2] **Shuhao Gu**[1,2] **Dengji Guo**[1,2] **Zhengxin Yang**[1,2] **Chenze Shao**[1,2] *
[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
{fengyang, gushuhao19b, guodengji19s}@ ict.ac.cn
{yangzhengxin17z, shaochenze18z}@ ict.ac.cn

Abstract

Although teacher forcing has become the main training paradigm for neural machine translation, it usually makes predictions only conditioned on past information, and hence lacks global planning for the future. To address this problem, we introduce another decoder, called seer decoder, into the encoder-decoder framework during training, which involves future information in target predictions. Meanwhile, we force the conventional decoder to simulate the behaviors of the seer decoder via knowledge distillation. In this way, at test the conventional decoder can perform like the seer decoder without the attendance of it. Experiment results on the Chinese-English, English-German and English-Romanian translation tasks show our method can outperform competitive baselines significantly and achieves greater improvements on the bigger data sets. Besides, the experiments also prove knowledge distillation the best way to transfer knowledge from the seer decoder to the conventional decoder compared to adversarial learning and L2 regularization.

1 Introduction

Neural machine translation (NMT) (Kalchbrenner and Blunsom, 2013; Sutskever et al., 2014; Bahdanau et al., 2014; Gehring et al., 2017; Vaswani et al., 2017) has achieved great success and is drawing larger attention recently. Most NMT models are under the attention-based encoder-decoder framework which assumes there is a common semantic space between the source and target languages. The encoder encodes the source sentence to the common space to get its meaning, and the decoder projects the source meaning to the target space to generate corresponding target words. Whenever generating a target word at a time step, the decoder needs to retrieve the attended source information and then decodes into a target word. The underline principle which makes sure the framework works is that the information hold by the source sentence and its target counterpart is equivalent. Thus the translation procedure can be considered to decompose source information into different pieces and then to convert each piece to a proper target word according to bilingual context. When all the information encoded in the source sentence is throughly processed, the whole translation has been generated.

Neural machine translation models are usually trained via maximum likelihood estimation (MLE) (Johansen and Juselius, 1990) and the operation form is known as *teacher forcing* (Williams and Zipser, 1989). The teacher forcing strategy performs one-step-ahead predictions with the past ground truth words fed as context and forces the distribution of the next prediction to approach a 0-1 distribution where the probability of the next ground truth word corresponds to 1 and others to 0. In this way, the predicted sequence is trained to be close to the ground truth sequence. From the perspective of information division, the function of teacher forcing is to teach the translation model how to segment source information and derive the ground truth word from the source information at a maximum probability.

However, teacher forcing can only provide up-to-now ground truth words for one-step-ahead predictions and hence lacks global planning for the future. This will result in local optimization especially when the next prediction is highly related to the future. Besides, as the translation grows, the previous prediction errors will be accumulated and affect later predictions (Zhang et al., 2019c). This is the important reason why NMT models cannot always produce the ground truth sequence during training. Therefore, it is more possible to achieve

The code: https://github.com/ictnlp/SeerForcingNMT

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 2862–2872
August 1–6, 2021.

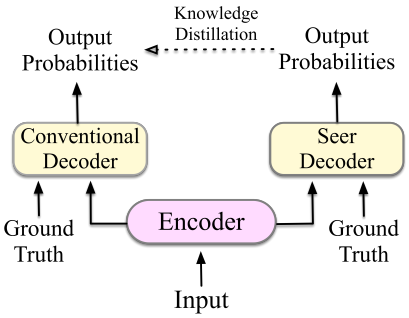

Figure 1: The architecture of the proposed method

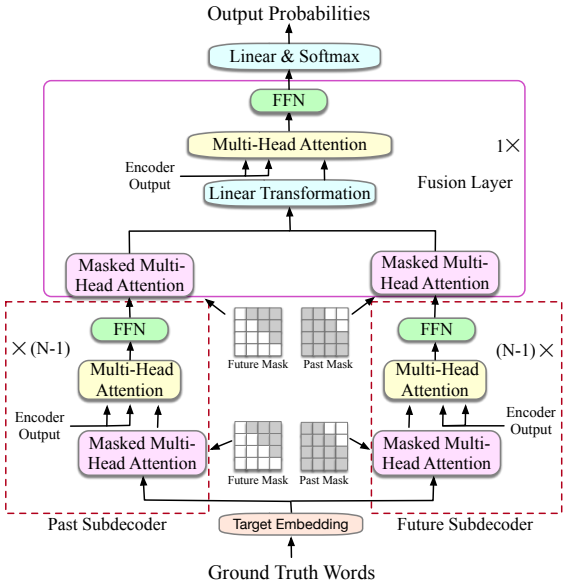

Figure 2: The architecture of the seer decoder

global optimization by getting to know the future ground truth words. This can lead to better cross-attention to the source sentence and thus better information devision. But unfortunately, ground truth can be only obtained during training and we cannot inference with future ground truth at test.

To address this problem, we introduce an additional seer decoder into the encoder-decoder framework to integrate future information. During training, the seer decoder is used to guide the behaviors of the conventional decoder while at test the translation model only inferences with the conventional decoder without introducing any extra parameters and calculation cost. Specifically, the conventional decoder only gets past information participating in the next prediction, while the seer decoder has both the past and future ground truth words engaged in the next prediction. Both decoders are trained to generate ground truth via MLE and meanwhile the conventional decoder is forced to simulate the behaviors of the seer decoder via knowledge distillation (Buciluǎ et al., 2006; Hinton et al., 2015). In this way, at test the conventional decoder can perform like the seer decoder as if it knew the future translation.

We conducted experiments on two small data sets (Chinese-English and English-Romanian) and two big data sets (Chinese-English and English-German) and the experiment results show that our method can outperform strong baselines on all the data sets. In addition, we also compared different mechanisms of transferring knowledge and found that knowledge distillation is more effective than adversarial learning and L2 regularization. To the best of our knowledge, this paper is the first to explore the effects of the three mechanisms simultaneously in machine translation.

2 The Proposed Method

We introduce our method on the basis of *Transformer* which is under the encoder-decoder framework (Vaswani et al., 2017). Our model consists of three components: the encoder, the conventional decoder and the seer decoder. The architecture is shown in Figure 1. The encoder and the conventional decoder work in the same way as the corresponding components of Transformer do. The seer decoder integrates future ground truth information into its self-attention representation and calculates cross-attention over source hidden states with the self-attention representation as the query. During training, the encoder is shared by the two decoders and both decoders perform predictions to generate ground truth. The behaviors of the conventional decoder are guided by the seer decoder via knowledge distillation. If the conventional decoder can predict a similar distribution as the seer decoder, we think the conventional decoder performs like the seer decoder. Then we can only use the conventional decoder for test.

The details of the encoder and the conventional decoder can be got from Vaswani et al. (2017). Assume the input sequence is $\mathbf{x} = (x_1, ..., x_J)$, the ground truth sequence is $\mathbf{y}^* = (y_1^*, ..., y_I^*)$ and the generated translation is $\mathbf{y} = (y_1, ..., y_I)$. We will give more description to the seer decoder and the training in what follows.

2.1 The Seer Decoder

Although we feed the future ground truth words to the seer decoder, we will not tell it the next ground truth word to be generated, in case it will only learn a copy operation, not how to derive a word. Considering efficiency, the seer decoder does not integrate the past and future ground truth information with a unique decoder , but two separate subdecoders. As a result, the seer decoder consists of three components: the past subdecoder, the future subdecoder and the fusion layer. The architecture of the seer decoder is given in Figure 2. The past and future subdecoders are employed to decode the past and future ground truth information into hidden states respectively and the fusion layer is used to fuse the output of the past and future subdecoders and calculate the final hidden state for the next prediction.

The past subdecoder is composed of $N-1$ layers and each layer has three sublayers which are the multi-head sublayer, the cross-attention sublayer and the feed-forward network (FNN) sublayer, the same as Transformer. The multi-attention sublayer accepts the whole ground truth sequence as the input and applies a mask matrix $\mathbf{M}_p$ to make sure only the past ground truth words attend the self-attention. Specifically, to generate the i-th target word, its corresponding mask vector in the mask matrix $\mathbf{M}_p$ is set to mask the words $y_i^*, y_{i+1}^*, ..., y_I^*$. Then after the cross-attention sublayer and the FFN sublayer, the past subdecoder output a sequence of past hidden states, the packed matrix of which is denoted as $\mathbf{H}_p$.

The future subdecoder has the same structure as the past subdecoder except for the mask matrix. The future subdecoder also has the whole ground truth sequence as the input but employs a different mask matrix $\mathbf{M}_f$ to only remain the future ground truth information. To generate the i-th target word, the corresponding mask vector in $\mathbf{M}_f$ masks the words $y_1^*, ..., y_{i-1}^*, y_i^*$. The packed matrix of the future hidden states generated by the future subdecoder is denoted as $\mathbf{H}_f$.

The fusion layer is composed of four sublayers: the multi-head sublayer, the linear sublayer, the cross-attention sublayer and the FFN sublayer. Except the linear sublayer, the rest three sublayers works in the same way as Transformer does. The multi-head sublayer encodes the outputs of the past and future subdecoders separately with the mask matrix $\mathbf{M}_p$ and $\mathbf{M}_f$, and the packed matrix of their output are denoted as $\mathbf{H}'_p$ and $\mathbf{H}'_f$ respectively. Then we reverse the order of the vectors in $\mathbf{H}'_f$ to get $\mathbf{H}''_f$, so that the same index in $\mathbf{H}'_p$ and $\mathbf{H}''_f$ can correspond to the past and future representation needed for the same prediction. Assume $\mathbf{H}'_f = [\mathbf{h}'_{f1}; \mathbf{h}'_{f2}; ...; \mathbf{h}'_{fI}]$, then its reversed matrix is $\mathbf{H}''_f = [\mathbf{h}'_{fI}; ...; \mathbf{h}'_{f2}; \mathbf{h}'_{f1}]$. The linear sublayer fuses $\mathbf{H}'_p$ and $\mathbf{H}''_f$ via a linear transformation as

$$\mathbf{A} = \mathbf{W}_p\mathbf{H}'_p + \mathbf{W}_f\mathbf{H}''_f \quad (1)$$

Now we can think each representation in the matrix $\mathbf{A}$ incorporates the past and future information for its corresponding prediction. Then after the cross-attention sublayer over the outputs of the encoder and then the FFN sublayer, we can get the target hidden states produced by the seer decoder as $\mathbf{S}_s = [\mathbf{s}_{s_1}; ...; \mathbf{s}_{s_I}]^T$. Then the probability to generate the target word y_i is

$$p_s(y_i|\mathbf{y}^*_{>i}, \mathbf{y}^*_{<i}, \mathbf{x}) \propto \exp\left(\mathbf{W}_o\mathbf{s}_{si}\right) \quad (2)$$

Note that the past and the future subdecoders share the same set of parameters, and the same linear transformation matrix $\mathbf{W}_o$ is applied to the outputs of the conventional and seer decoders.

3 Training

In our method, only the conventional decoder is employed for test and the seer decoder is only used to guide the conventional decoder during training. Given a sentence pair $\langle \mathbf{x}, \mathbf{y}^* \rangle$ in the training set, the conventional decoder and the seer decoder can predict a distribution for target position i as $p_c(y_i|\mathbf{y}^*_{<i}, \mathbf{x})$ and $p_s(y_i|\mathbf{y}^*_{>i}, \mathbf{y}^*_{<i}, \mathbf{x})$, respectively. The two decoders are both trained by comparing its predicted distribution with the 0-1 distribution of the ground truth word by minimizing the cross entropy, that is to maximize the likelihood of the corresponding ground truth word. As the two decoders involve different information for next prediction, we call the training strategy *teacher forcing* and *seer forcing*, respectively. The cross-entropy loss for the conventional decoder is

$$\mathcal{L}_c = -\sum_{k=1}^{\mathrm{K}}\sum_{i=1}^{I_k} \log p_c(y_i^*|\mathbf{y}^*_{<i}, \mathbf{x}), \quad (3)$$

and the cross-entropy loss for the seer decoder is

$$\mathcal{L}_s = -\sum_{k=1}^{\mathrm{K}}\sum_{i=1}^{I_k} \log p_s(y_i^*|\mathbf{y}^*_{>i}, \mathbf{y}^*_{<i}, \mathbf{x}). \quad (4)$$

where K is the size of the training set and I_k is the length of the k-th target sentence.

The conventional decoder is further trained to get close to the distribution of the seer decoder via knowledge distillation. In knowledge distillation, the conventional decoder (*the student*) has to not only match the one-hot ground truth word, but fit the distribution over the target vocabulary V drawn by the seer decoder (*the teacher*). The knowledge distillation loss can be formalized as

$$\mathcal{L}_{kd} = -\sum_{k=1}^{\mathrm{K}}\sum_{i=1}^{I_k}\sum_{l=1}^{|\mathrm{V}|} p_s(y_i = l|\mathbf{y}^*_{>i}, \mathbf{y}^*_{<i}, \mathbf{x}) \times \log p_c(y_i = l|\mathbf{y}^*_{<i}, \mathbf{x}) \quad (5)$$

where $|\mathrm{V}|$ is the size of the target vocabulary.

The final training loss is

$$\mathcal{L} = \mathcal{L}_s + \lambda\mathcal{L}_c + (1-\lambda)\mathcal{L}_{kd}\ . \quad (6)$$

Different from the conventional knowledge distillation which first trains the teacher via cross entropy against ground truth, then fixes the teacher and only trains the student, we train all the parameters from the scratch, but we still follow the above rule to keep the teacher (i.e. the seer decoder) unchanged in the process of distillation. To do this, we do not update the parameters of the seer decoder through the loss $\mathcal{L}_{kd}$, that is, we only back propagate gradients to the seer decoder through $\mathcal{L}_s$, but not through $\mathcal{L}_{kd}$.

4 Related Work

Reinforcement-learning-based methods also encode future information in the rewards to supervise fine-tuning of the translation model. The rewards are worked out either by sampling future translation with the REINFORCE algorithm (Williams, 1992; Yu et al., 2017; Yang et al., 2018; Shao et al., 2019), or by directly calculating a value with the actor-critic algorithm (Bahdanau et al., 2016; Li et al., 2017). This set of methods only give a weak supervision to the NMT model through rewards and suffer from unstable training. In contrast, Shao et al. (2018) propose to train autoregressive NMT with the probabilistic n-gram based GLEU (Wu et al., 2016) and Shao et al. (2020) propose to minimize the bag-of-ngrams difference for non-autoregressive NMT so that the two methods can abandon reinforcement learning and perform training directly by gradient descent.

Another set of methods introduce future information into inference with additional pass of decoding or extra components at test. Niehues et al. (2016), Xia et al. (2017), Hassan et al. (2018) and Zhang et al. (2018) proposed a two-pass decoding algorithm to first generate a draft translation and then generate final translation referring to the draft. Geng et al. (2018) expand this line of methods by performing an adaptive multi-pass decoding where the number of decoding passes is determined by a policy network. Liu et al. (2016a), Liu et al. (2016b), Hoang et al. (2017), Zhang et al. (2019d) and He et al. (2019) perform bidirectional decoding simultaneously and the two decoders correlate to each other via an agreement term or a regularization term in the loss. Zhou et al. (2019a) , Zhou et al. (2019b) and Zhang et al. (2019b) also maintain a forward decoder and a backward decoder to decode simultaneously but they interact to each other when making predictions. Zhang et al. (2019a) introduce a future-aware vector at test which is learned via the knowledge distillation framework during training. The difference between this set of methods and our method is that our method does not require any other cost at test and is easy to use.

There are some other works which integrate future information during training while only perform one-pass decoding. Serdyuk et al. (2018) introduce a twin network to perform bidirectional decoding simultaneously during training and force the hidden states generated by the two decoders to be consistent, then at inference it can only use the forward decoder. But in this method the two decoders act as a counterpart to each other and no decoder plays a role of teacher, which determines that it can only be trained via L_2 regularization, not knowledge distillation which has proven in the experiments more effective than L_2 regularization. Feng et al. (2020) introduce an evaluation module to give each translation more reasonable evaluation when it cannot match the ground truth. The evaluation is conducted from the perspective of fluency and faithfulness which both need the participation of past and future information. The difference from the method proposed in this paper is their method uses self-generated translation as past information and does not train with knowledge distillation.

Some researchers work in another perspective by introducing future information. Zhang et al. (2020b) propose to employ future source information to guide simultaneous machine translation with knowledge distillation, so that the incompleteness of source can be mitigated. Zheng et al. (2018) and

Zheng et al. (2019) propose to model past and future information for the source to help the decoder focus on untranslated source information.

5 Experiments

5.1 Settings

5.1.1 Data Preparation

We conducted experiments on two small data sets and two big data sets.

Small Data Sets

Chinese→English The training set consists of about 1.25M sentence pairs from LDC corpora with 27.9M Chinese words and 34.5M English words respectively[1]. We used MT02 for validation and MT03, MT04, MT05, MT06, MT08 for test. We tokenized and lowercased English sentences using the Moses scripts[2], and segmented the Chinese sentences with the Stanford Segmentor[3]. The two sides were further segmented into subword units using Byte-Pair Encoding(BPE) (Sennrich et al., 2016) with 30K merge operations. 32K size of the Chinese dictionary and 29K size of the English dictionary were built for the two sides.

English→Romanian We used the preprocessed version of WMT16 En-Ro dataset released by Lee et al. (2018) which includes 0.6M sentence pairs. We used news-dev 2016 for validation and newstest 2016 for test. The two languages share the 35K size of the joint vocabulary generated with 40K merge operations of BPE on the combined data.

Big Data Sets

Chinese→English The training data is from WMT 2017 Zh-En translation tasks that contains 20.18M sentence pairs after deleting duplicate ones. The newsdev2017 was used as the development set and newstest2017 was used as the test set. To avoid the effects of the translationese (Graham et al., 2019), we also tested the methods on the newstest2019 test set. We tokenized and truecased the English sentences with Moses scripts. For the Chinese data, we performed word segmentation by using Stanford Segmenter. 32K BPE sizes were applied to the training data seperately and then we filtered out the sentences which are longer than 128 sub-words. 44K size of the Chinese dictionary and 33K size of the English dictionary were built based on the corresponding data.

English→German The training data is from WMT2016 which consists of about 4.5M sentences pairs with 118M English words and 111M German words. The newstest2014 was used as the development set and newstest2016 and newstest2019 were used as the test sets. The two languages share the 32K size of the joint vocabulary generated with 30K merge operations of BPE on the combined data.

5.1.2 Systems

TRANSFORMER We used an open-source toolkit called *Fairseq-py* released by Facebook (Ott et al., 2019) which was implemented strictly following Vaswani et al. (2017).

RL-NMT We trained Transformer under the reinforcement learning framework using the REINFORCE algorithm (Williams, 1992) with the BLEU as the rewards. The implementation details for the RL part is the same as Yang et al. (2018).

ABDNMT Our implementation of Zhang et al. (2018) based on Transformer.

TWINNET Our implementation of Serdyuk et al. (2018) based on Transformer. The weight of L_2 loss was 0.2 .

EVANMT Our implementation of Feng et al. (2020).

SEER+L_2 Seer forcing with L_2 regularization. Similar to TWINNET, we set $\mathcal{L}_2 = \sum_{k=1}^{K}\sum_{i=1}^{I_k}\|g(\mathbf{s}_{ti}) - \mathbf{s}_{si})\|_2$ where g is a linear transformation. We first pretrained the two decoders together only with $\mathcal{L} = \mathcal{L}_t + \mathcal{L}_s$, then trained them with the loss of $\mathcal{L} = \mathcal{L}_t + \mathcal{L}_s + \alpha\mathcal{L}_2$ where $\alpha = 0.2$, too. Please note that the L_2 loss did not update the seer decoder and the encoder so that the conventional decoder would approach the seer decoder, which followed Serdyuk et al. (2018).

SEER+AL Seer forcing with adversarial learning. A discriminator is employed to distinguish the hidden state sequences generated by the conventional decoder and the seer decoder. The discriminator is based on CNN, implemented according to Gu et al. (2019). The translation model and the discriminator are trained jointly via a gradient reversal layer just like our method. The loss is $\mathcal{L} = \mathcal{L}_t + \mathcal{L}_s + \alpha\mathcal{L}_d$ where $\mathcal{L}_d$ is the loss of the discriminator and $\alpha = 0.3$ on the EN→RO data set and $\alpha = 0.2$ on the other data sets.

[1]The corpora include LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06.

[2]http://www.statmt.org/moses/

[3]https://nlp.stanford.edu/

	CN→EN								EN→RO		
	MT03	**MT04**	**MT05**	**MT06**	**MT08**	**AVG**	**Δ**	**TIME**	**WMT16**	**Δ**	**TIME**
TRANSFORMER	46.54	46.95	46.39	45.39	36.75	44.40		1.0	32.60		1.0
RL-NMT	45.75	47.41	46.44	47.08	37.65	44.87	*+0.47*	1.70	32.79	*+0.19*	2.38
ABDNMT	47.16	47.58	46.77	45.97	36.43	44.78	*+0.38*	2.78	33.80	*+1.20*	3.36
TWINNET	47.78	48.74	**48.59**	46.65	**38.80**	46.11	*+1.71*	2.56	33.79	*+1.19*	2.62
EVANMT	47.05	47.76	46.59	46.58	37.39	45.07	*+0.67*	2.19	33.29	*+0.69*	2.91
SEER+L_2	47.98**	48.66**	48.16**	47.02**	38.64**	46.09	*+1.69*	1.90	33.55**	*+0.95*	1.83
SEER+AL	47.91**	48.38**	47.97**	47.04**	38.18**	45.89	*+1.49*	2.64	33.59**	*+1.04*	2.35
Our Method	**48.12****	**48.85****	48.25**	**47.25****	38.71**	**46.24**	***+1.84***	1.92	**33.86****	***+1.26***	1.86

Table 1: BLEU scores on small data sets. ** mean the improvements over TRANSFORMER is statistically significant (Collins et al., 2005) ($\rho < 0.01$, respectively).

	CN→EN					EN→DE				
	2017	**Δ**	**2019**	**Δ**	**TIME**	**2016**	**Δ**	**2019**	**Δ**	**TIME**
TRANSFORMER	23.75		26.00		1.0	33.49		36.20		1.0
TWINNET	23.39	*-0.36*	26.09	*+0.09*	2.58	33.05	*-0.44*	35.69	*-0.51*	2.57
EVANMT	–	–	–	–	–	34.00	*+0.51*	37.25	*+1.05*	2.48
SEER+L_2	23.95	*+0.20*	25.82	*-0.18*	1.93	33.58	*+0.09*	36.65	*+0.45*	1.53
SEER+AL	24.01	*+0.26*	26.47*	*+0.47*	2.29	34.03	*+0.54*	36.81	*+0.61*	2.39
Our Method	**24.35***	***+0.60***	**26.80****	***+0.80***	1.97	**34.25****	***+0.76***	**37.34***	***+1.14***	1.57

Table 2: BLEU scores on big data sets. * and ** mean the improvements over TRANSFORMER is statistically significant (Collins et al., 2005) ($\rho < 0.05$ and $\rho < 0.01$, respectively).

Our Method Implemented based on Fairseq-py. The weight λ in Equation 6 for the small Chinese→English data set is set to 0.25, and for other data sets is set to 0.5.

All the Transformer-based systems have the same configuration as the base model described in Vaswani et al. (2017) except that dropout rate is 0.3. The translation quality was evaluated with BLEU (Papineni et al., 2002) with n=4 using the *SacreBLEU* tool (Post, 2018)[4], where small data sets employ case-insensitive BLEU while big data sets use case-sensitive BLEU.

5.2 Main Results

We compare our method with other methods that can make global planning, including the reinforcement-based method (RL-NMT), the two-pass decoding method (ABDNMT), twin networks which match past and future information (TWINNET) and the NMT model with an evaluate module to evaluate fluency and faithfulness (EVANMT). In addition, we also explore learning mechanisms which can transfer knowledge from the seer decoder to the conventional decoder, including L_2 regularization (SEER+L_2), adversarial learning (SEER+AL) and knowledge distillation (Our Method).

We report results together with training time on the small and big data sets in Table 1 and Table 2, respectively.[5] As for different methods, in the small data sets, RL-NMT can only get small improvements over Transformer which are in line with the results reported in Wu et al. (2018), and ABDNMT cannot get consistent improvements over Transformer with an obvious difference on the EN→RO data set and a small difference on the CN→EN data set. TWINNET can get comparable BLEU scores with our method on the small data sets but mostly negative difference on the big data sets. EVANMT can achieve consistent improvements and greater improvements on the EN→DE data set. For the learning mechanisms, knowledge distillation show consistent superiority over L_2 regularization and adversarial learning, which is remarkable especially on the big data sets. Adversarial learning can bring improvements over Transformer on all the data sets while L_2 regularization acts unstable on the big data sets. In summary, our method proved to be effective not only in the term of the architecture but also in the learning mechanism.

[4]BLEU+case.mixed+numrefs.1+smooth.exp+tok.13a+version.1.2.11

[5]Please note that there is no comparability between our results and that of Zhang et al. (2019a) because we used different validation and test sets.

	MT03	MT04	MT05	MT06	MT08	AVG
CD *w/o* CA	6.24	6.68	6.70	6.77	4.49	6.12
SD *w/o* CA	16.39	16.70	16.64	17.21	11.97	15.78
CD *with* CA	29.45	25.03	30.14	32.07	23.39	28.02
SD *with* CA	52.61	45.57	52.02	52.68	44.14	49.40

Table 3: BLEU scores of teacher forcing and seer forcing with and without cross-attention on NIST CN→EN translation. CD and SD denote the conventional decoder and the seer decoder, respectively. CA represents cross-attention.

5.3 The Superiority of the Seer Decoder

To use seer forcing to guide teacher forcing, it should be ensured that the seer decoder can outperform the conventional decoder. To verify this, we trained the two decoders together with the loss $\mathcal{L} = \mathcal{L}_t + \mathcal{L}_s$ without knowledge distillation. Then we evaluated their performance on the small Chinese-English translation task as follows. Both decoders are fed with ground truth words as context at test so that they can inference in the same way as at training, where the conventional decoder uses the past ground truth as context and the seer decoder employs the past and future ground truth words as context in the past and future subdecoders.

Besides translation performance, we also check the superiority of seer decoder in target language modeling. We do this by dropping out cross-attention so that the decoder can only generate translation based on target language model. In this way, the translation performance without cross-attention can demonstrate the ability of the two decoders in target language modeling.

We used the first reference of the test set as ground truth and calculated BLEU scores only with this reference. From the results in Table 3, we can see that whether with or without cross-attention the seer decoder can make super large improvements over the conventional decoder consistently on all the test sets. However, without cross-attention, the BLEU scores of both decoders decrease dramatically which means language model information is not enough for the translation task. Therefore, we can conclude the seer decoder acts much better in target language modeling and cross-language projection and it is reasonable to use the seer decoder as the guider.

5.4 The Distillation of Future Information

As the seer decoder achieves its superiority with the help of future target information, we hope that the conventional decoder can learn future information from the seer decoder with knowledge distillation.

	Accuracy	Recall	F1-Score
TRANSFORMER	47.23	40.91	43.84
Our Method	52.24	42.10	46.63

Table 4: Comparison on the predicted bag of words between the conventional decoders

To check this, we tested whether the hidden states of the conventional decoder could derive more future ground truth words after knowledge distillation. The underlying belief is that the future ground information transferred from the seer decoder can help the conventional decoder derive more future ground truth words.

Assuming the hidden states generated by the conventional decoder are $\mathbf{S}_t = [\mathbf{s}_{t_1}; ...; \mathbf{s}_{t_I}]^T$, the future words for each target position i can be predicted with the distribution

$$\mathbf{P}_{wi} \sim \text{softmax}(\mathbf{W}_w \mathbf{s}_{ti}) \tag{7}$$

where $\mathbf{W}_w$ is the weight matrix. During training, we can get the bag of ground truth words for position i as $\mathbf{y}_i^* = \{y_{i+1}^*, ..., y_I^*\}$ and train $\mathbf{W}_w$ with other parameters fixed by maximizing the likelihood of $\mathbf{y}_i^*$ as

$$\mathcal{L}_w = -\sum_{k=1}^{K}\sum_{i=1}^{I_k}\sum_{w \in \mathbf{y}_i^*} \log p_{wi}(w) \tag{8}$$

where K is the size of training sentences, I_k is the length of the target sentence and $\log p_{wi}(w)$ is the probability of the word w in Equation 7.

At test, we select the top best I_{b_i} words according to Equation 7 as the bag of future words $\mathbf{b}_i$ for position i. As we cannot get the ground truth, the size of $\mathbf{b}_i$ is calculated approximately as $I_{b_i} = \max\{2, (J - i) \times 2\}$ where J is the length of source sentence. As we do not know the target length during prediction, it may occur that i is greater than J and calculating I_{b_i} in this way can ensure $\mathbf{b}_i$ contains 2 words at least.

We conducted experiments on Chinese-English translation and used MT02 as the test set only

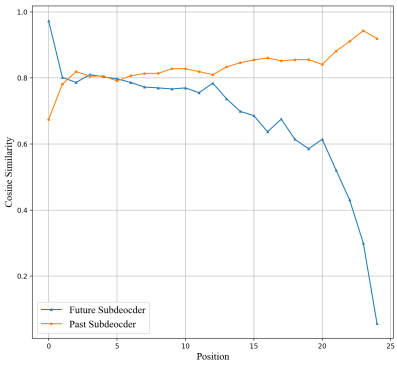

Figure 3: The similarity of the past and future information to the fused information

	AVG	Δ
Our Method	46.24	
-FUTURE	45.38	*-0.86*
-PAST	45.42	*-0.82*
-KD	44.84	*-1.40*
TRANSFORMER	44.40	*-1.84*

Table 5: Ablation study on NIST CN→EN translation. -FUTURE : dropping the future subdecoder; -PAST: dropping the past subdecoder; -KD: dropping knowledge distillation.

with the first reference as ground truth. We calculated the accuracy and recall by comparing each $\mathbf{b}_i$ against each $\mathbf{y}_i^*$. The results in Table 4 show the conventional decoder in our method can achieve higher accuracy and recall compared to the decoder of Transformer. This means knowledge distillation does transfer future information from the seer decoder to the conventional decoder.

5.5 The Contribution of Subdecoders

In the seer decoder of our method, the information from the past and future subdecoders is fused (as shown in Equation 1) to get the final cross-attention. The intuition is that at the beginning stage, the past subdecoder contains less information than the future subdecoder, so the fused information should rely more on the future subdecoder. As the translation gets longer, the information embodied in the past subdecoder grows, and the fused information should depend more on the past subdecoder. To confirm this hypothesis, we calculate the cosine similarity of the vectors in $\mathbf{A}$ given in Equation 1 with the corresponding weighted vectors of $\mathbf{W}_p\mathbf{H}'_p$ and $\mathbf{W}_f\mathbf{H}''_f$.

We selected 205 sentences the length of which

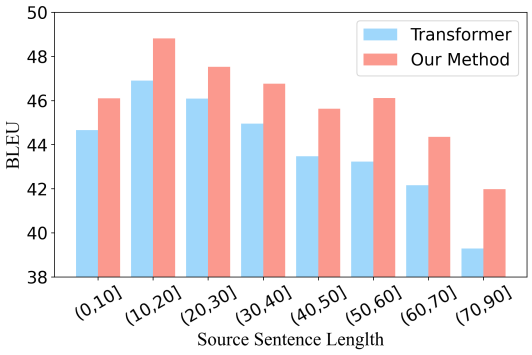

Figure 4: The BLEU scores on sentence bins with different lengths.

ranges $[15, 25]$, then calculated the cosine similarities word by word. Then the similarities at the same target position will be averaged and the chart over all the target positions is given in Figure 3. The figure confirms our conjecture that at first, the fused information is highly related to the future information, and over time the similarity to past information increases gradually while the similarity to future information decreases faster.

5.6 Ablation Study

We have proven that in our method the past and future information collaborate to achieve better global planning. In this section, we will explore the influence of past and future information by separately deleting the *future* and *past* subdecoders from the seer decoder. In both cases, only the structure of the seer decoder changes and the whole model is trained with knowledge distillation in the same way. We also remove knowledge distillation loss in which case the seer and conventional decoders only interact via the shared encoder and only optimize their own cross-entropy losses during training. The results are given in Table 5.

When we exclude future or past information, the translation performance decreases dramatically at almost the same extent, but they still have an obvious gain compared to Transformer. This demonstrates that both the past and future information are necessary for global planning. It is interesting that the translation performance still rise without future subdecoder where there is no additional information fed compared to Transformer. The reason may be the conventional and seer decoder can restrict each other to avoid bad behaviors. When knowledge distillation is dropped, the performance decline greatly which means only communicating via the encoder the conventional and seer decoders

is not enough. Hence we need to introduce knowledge distillation to reinforce the influence of the seer decoder to the conventional decoder.

5.7 Performance with Sentence Length

As the translation is generated word by word, the translation errors will be accumulated while the the translation grows, which will influence the later prediction. In our method, the conventional decoder can learn future information from the seer decoder and hence it should make better global planning for the whole sequence. From this, we deduce that our method performs better on long sentences than Transformer.

We checked this on the NIST CN→EN translation task and split the sentences in all the test sets into 8 bins according to their length. Then we translated for each bin and tested the BLEU scores. The results in Figure 4 show that our method can achieve bigger improvements on longer sentences, especially in the last three bins.

6 Conclusion

In order to help the NMT model to make good global planning at inference, we propose to introduce a seer decoder which embodies future ground truth to guide the behaviors of the conventional decoder. To this end, we employ the method of knowledge distillation to transfer future information from the seer decoder to the conventional decoder. At test, the conventional decoder can perform translation on its own as if it knew some future information. The experiments indicate our method can outperform strong baselines significantly on four data sets. We are also the first to explore learning mechanisms of knowledge distillation, adversarial learning and L_2 regularization and knowledge distillation has proven to be the most effective one.

Acknowledgement

This paper was supported by National Key R&D Program of China (NO. 2017YFE0192900). Thank Wanying Xie for running experiments of EVANMT. Thank all the anonymous reviewers for the insightful and valuable comments.

References

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.

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

Cristian Buciluǎ, Rich Caruana, and Alexandru Niculescu-Mizil. 2006. Model compression. In *Proceedings of the 12th ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 535–541.

Michael Collins, Philipp Koehn, and Ivona Kucerova. 2005. Clause restructuring for statistical machine translation. In *Proceedings of ACL 2015*, pages 531–540.

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 AAAI Conference on Artificial Intelligence*, volume 34, pages 59–66.

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*, pages 1243–1252. JMLR. org.

Xinwei Geng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. Adaptive multi-pass decoder for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 523–532.

Yvette Graham, Barry Haddow, and Philipp Koehn. 2019. Translationese in machine translation evaluation. *CoRR*, abs/1906.09833.

Shuhao Gu, Yang Feng, and Qun Liu. 2019. Improving domain adaptation translation with domain invariant and specific information. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 3081–3091.

Hany Hassan, Anthony Aue, Chang Chen, Vishal Chowdhary, Jonathan Clark, Christian Federmann, Xuedong Huang, Marcin Junczys-Dowmunt, William Lewis, Mu Li, et al. 2018. Achieving human parity on automatic chinese to english news translation. *arXiv preprint arXiv:1803.05567*.

Zhongjun He, Hua Wu, Haifeng Wang, et al. 2019. Multi-agent learning 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 855–864.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *arXiv preprint arXiv:1503.02531*.

Cong Duy Vu Hoang, Gholamreza Haffari, and Trevor Cohn. 2017. Towards decoding as continuous optimisation in neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 146–156.

Søren Johansen and Katarina Juselius. 1990. Maximum likelihood estimation and inference on cointegration with applications to the demand for money. *Oxford Bulletin of Economics and statistics*, 52(2):169–210.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. 2018. Deterministic non-autoregressive neural sequence modeling by iterative refinement. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 1173–1182.

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

Lemao Liu, Andrew Finch, Masao Utiyama, and Eiichiro Sumita. 2016a. Agreement on target-bidirectional lstms for sequence-to-sequence learning. In *Thirtieth AAAI Conference on Artificial Intelligence*.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016b. Agreement on target-bidirectional neural machine translation. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 411–416.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 1828–1836.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of NAACL-HLT 2019: Demonstrations*.

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.

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, Belgium, Brussels. Association for Computational Linguistics.

Weizhen Qi, Yu Yan, Yeyun Gong, Dayiheng Liu, Nan Duan, Jiusheng Chen, Ruofei Zhang, and Ming Zhou. 2020. Prophetnet: Predicting future n-gram for sequence-to-sequence pre-training. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: Findings*, pages 2401–2410.

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)*, volume 1, pages 1715–1725.

Dmitriy Serdyuk, Nan Rosemary Ke, Alessandro Sordoni, Adam Trischler, Chris Pal, and Yoshua Bengio. 2018. Twin networks: Matching the future for sequence generation.

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.

Chenze Shao, Yang Feng, Jinchao Zhang, Fandong Meng, Xilin Chen, and Jie Zhou. 2019. Retrieving sequential information for non-autoregressive neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 3013–3024.

Chenze Shao, Jinchao Zhang, Yang Feng, Fandong Meng, and Jie Zhou. 2020. Minimizing the bag-of-ngrams difference for non-autoregressive neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 198–205.

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.

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.

Ronald J Williams. 1992. Simple statistical gradient-following algorithms for connectionist reinforcement learning. *Machine learning*, 8(3-4):229–256.

Ronald J Williams and David Zipser. 1989. A learning algorithm for continually running fully recurrent neural networks. *Neural computation*, 1(2):270–280.

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

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*.

Yingce Xia, Fei Tian, Lijun Wu, Jianxin Lin, Tao Qin, Nenghai Yu, and Tie-Yan Liu. 2017. Deliberation networks: Sequence generation beyond one-pass decoding. In *Advances in Neural Information Processing Systems*, pages 1784–1794.

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

Lantao Yu, Weinan Zhang, Jun Wang, and Yong Yu. 2017. Seqgan: Sequence generative adversarial nets with policy gradient. In *Thirty-First AAAI Conference on Artificial Intelligence*.

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

Jiajun Zhang, Long Zhou, Yang Zhao, and Chengqing Zong. 2019b. Synchronous bidirectional inference for neural sequence generation. *arXiv preprint arXiv:1902.08955*.

Ruiyi Zhang, Changyou Chen, Zhe Gan, Wenlin Wang, Dinghan Shen, Guoyin Wang, Zheng Wen, and Lawrence Carin. 2020a. Improving adversarial text generation by modeling the distant future. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2516–2531.

Shaolei Zhang, Yang Feng, and Liangyou Li. 2020b. Future-guided incremental transformer for simultaneous translation. *arXiv preprint arXiv:2012.12465*.

Wen Zhang, Yang Feng, Fandong Meng, Di You, and Qun Liu. 2019c. 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.

Xiangwen Zhang, Jinsong Su, Yue Qin, Yang Liu, Rongrong Ji, and Hongji Wang. 2018. Asynchronous bidirectional decoding for neural machine translation. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Zhirui Zhang, Shuangzhi Wu, Shujie Liu, Mu Li, Ming Zhou, and Tong Xu. 2019d. Regularizing neural machine translation by target-bidirectional agreement. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 443–450.

Zaixiang Zheng, Shujian Huang, Zhaopeng Tu, Xin-Yu Dai, and Jiajun Chen. 2019. Dynamic past and future for neural machine translation. *arXiv preprint arXiv:1904.09646*.

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. 2019a. Synchronous bidirectional neural machine translation. *Transactions of the Association for Computational Linguistics*, 7:91–105.

Long Zhou, Jiajun Zhang, Chengqing Zong, and Heng Yu. 2019b. Sequence generation: From both sides to the middle. *Proceedings of the Twenty-Eighth International Joint Conference on Artificial Intelligence*.