Modeling Past and Future for Neural Machine Translation

Zaixiang Zheng[*]
Nanjing University
zhengzx@nlp.nju.edu.cn

Hao Zhou[*]
Toutiao AI Lab
zhouhao.nlp@bytedance.com

Shujian Huang
Nanjing University
huangsj@nlp.nju.edu.cn

Lili Mou
University of Waterloo
doublepower.mou@gmail.com

Xinyu Dai
Nanjing University
dxy@nlp.nju.edu.cn

Jiajun Chen
Nanjing University
chenjj@nlp.nju.edu.cn

Zhaopeng Tu
Tencent AI Lab
zptu@tencent.com

Abstract

Existing neural machine translation systems do not explicitly model what has been translated and what has not during the decoding phase. To address this problem, we propose a novel mechanism that separates the source information into two parts: translated PAST contents and untranslated FUTURE contents, which are modeled by two additional recurrent layers. The PAST and FUTURE contents are fed to both the attention model and the decoder states, which provides Neural Machine Translation (NMT) systems with the knowledge of translated and untranslated contents. Experimental results show that the proposed approach significantly improves the performance in Chinese-English, German-English, and English-German translation tasks. Specifically, the proposed model outperforms the conventional coverage model in terms of both the translation quality and the alignment error rate.[†]

1 Introduction

Neural machine translation (NMT) generally adopts an *encoder-decoder* framework (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014), where the *encoder* summarizes the source sentence into a *source context vector*, and the *decoder* generates the target sentence word-by-word based on the given source. During translation, the decoder implicitly serves several functionalities at the same time:

1. Building a language model over the target sentence for translation fluency (LM).
2. Acquiring the most relevant source-side information to generate the current target word (PRESENT).
3. Maintaining what parts in the source have been translated (PAST) and what parts have not (FUTURE).

However, it may be difficult for a single recurrent neural network (RNN) decoder to accomplish these functionalities simultaneously. A recent successful extension of NMT models is the *attention* mechanism (Bahdanau et al., 2015; Luong et al., 2015), which makes a soft selection over source words and yields an *attentive vector* to represent the most relevant source parts for the current decoding state. In this sense, the attention mechanism separates the PRESENT functionality from the decoder RNN, achieving significant performance improvement.

In addition to PRESENT, we address the importance of modeling PAST and FUTURE contents in machine translation. The PAST contents indicate translated information, whereas the FUTURE contents indicate untranslated information, both being crucial to NMT models, especially to avoid under-translation and over-translation (Tu et al., 2016). Ideally, PAST grows and FUTURE declines during the translation process. However, it may be difficult for a single RNN to explicitly model the above processes.

In this paper, we propose a novel neural machine translation system that explicitly models PAST and FUTURE contents with two additional RNN layers. The RNN modeling the PAST contents (called PAST layer) starts from scratch and accumulates the in-

[*] Equal contributions.

[†] Our code can be downloaded from `https://github.com/zhengzx-nlp/past-and-future-nmt`.

Transactions of the Association for Computational Linguistics, vol. 6, pp. 145–157, 2018. Action Editor: Philipp Koehn.
Submission batch: 6/2017; Revision batch: 9/2017; Published 3/2018.

formation that is being translated at each decoding step (i.e., the PRESENT information yielded by attention). The RNN modeling the FUTURE contents (called FUTURE layer) begins with holistic source summarization, and subtracts the PRESENT information at each step. The two processes are guided by proposed auxiliary objectives. Intuitively, the RNN state of the PAST layer corresponds to source contents that have been translated at a particular step, and the RNN state of the FUTURE layer corresponds to source contents of untranslated words. At each decoding step, PAST and FUTURE together provide a full summarization of the source information. We then feed the PAST and FUTURE information to both the attention model and decoder states. In this way, our proposed mechanism not only provides coverage information for the attention model, but also gives a holistic view of the source information at each time.

We conducted experiments on Chinese-English, German-English, and English-German benchmarks. Experiments show that the proposed mechanism yields 2.7, 1.7, and 1.1 improvements of BLEU scores in three tasks, respectively. In addition, it obtains an alignment error rate of 35.90%, significantly lower than the baseline (39.73%) and the coverage model (38.73%) by Tu et al. (2016). We observe that in traditional attention-based NMT, most errors occur due to over- and under-translation, which is probably because the decoder RNN fails to keep track of what has been translated and what has not. Our model can alleviate such problems by explicitly modeling PAST and FUTURE contents.

2 Motivation

In this section, we first introduce the standard attention-based NMT, and then motivate our model by several empirical findings.

The attention mechanism, proposed in Bahdanau et al. (2015), yields a dynamic source context vector for the translation at a particular decoding step, modeling PRESENT information as described in Section 1. This process is illustrated in Figure 1.

Formally, let $\mathbf{x} = \{x_1, \ldots, x_I\}$ be a given input sentence. The encoder RNN—generally implemented as a bi-directional RNN (Schuster and Paliwal, 1997)—transforms the sentence to a sequence of annotations with $\mathbf{h}_i = \big[\overrightarrow{\mathbf{h}}_i; \overleftarrow{\mathbf{h}}_i\big]$ being the annotation of x_i. ($\overrightarrow{\mathbf{h}}_i$ and $\overleftarrow{\mathbf{h}}_i$ refer to RNN's hidden states in both directions.)

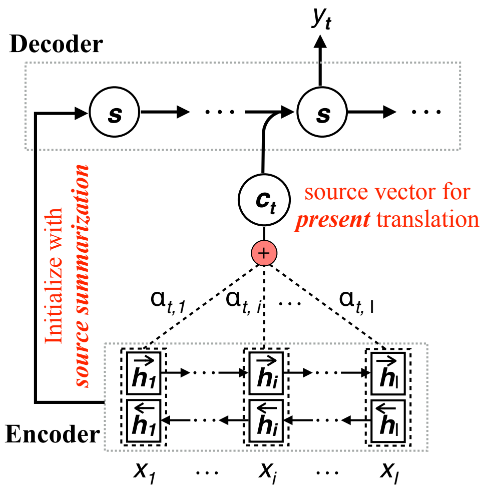


Figure 1: Architecture of attention-based NMT.

Based on the source annotations, another decoder RNN generates the translation by predicting a target word y_t at each time step t:

$$P(y_t|y_{<t}, \mathbf{x}) = \mathrm{softmax}(g(y_{t-1}, \mathbf{s}_t, \mathbf{c}_t)), \qquad (1)$$

where $g(\cdot)$ is a non-linear activation, and $\mathbf{s}_t$ is the decoding state for time step t, computed by

$$\mathbf{s}_t = f(y_{t-1}, \mathbf{s}_{t-1}, \mathbf{c}_t). \qquad (2)$$

Here $f(\cdot)$ is an RNN activation function, e.g., the Gated Recurrent Unit (GRU) (Cho et al., 2014) and Long Short-Term Memory (LSTM) (Hochreiter and Schmidhuber, 1997). $\mathbf{c}_t$ is a vector summarizing relevant source information. It is computed as a weighted sum of the source annotations:

$$\mathbf{c}_t = \sum_{i=1}^{I} \alpha_{t,i} \cdot \mathbf{h}_i, \qquad (3)$$

where the weights ($\alpha_{t,i}$ for $i = 1 \cdots, I$) are given by the attention mechanism:

$$\alpha_{t,i} = \mathrm{softmax}\big(a(\mathbf{s}_{t-1}, \mathbf{h}_i)\big). \qquad (4)$$

Here, $a(\cdot)$ is a scoring function, measuring the degree to which the decoding state and source information match to each other.

SRC	与此同时,他呼吁提高民事服务效率, 这也是鼓舞民心之举。
REF	in the meanwhile he calls for better efficiency in civil service , **which helps to promote people 's trust .**
NMT	at the same time , he called for a higher *efficiency* in civil service *efficiency* .

(a) Translation example. We highlight **under-translated** words in bold and italicize *over-translated* words.

Initialize Decoder States with ...	BLEU
Source Summarization	35.13
All-Zero Vector	35.01

(b) Source summarization is not fully exploited by NMT decoder.

Table 1: Evidence shows that attention-based NMT fails to make full use of source information, thus losing the holistic picture of source contents.

Intuitively, the attention-based decoder selects source annotations that are most relevant to the decoder state, based on which the current target word is predicted. In other words, $\mathbf{c}_t$ is some source information for the PRESENT translation.

The decoder RNN is initialized with the summarization of the entire source sentence $[\overrightarrow{\mathbf{h}}_I; \overleftarrow{\mathbf{h}}_1]$, given by:

$$\mathbf{s}_0 = \tanh(W_s[\overrightarrow{\mathbf{h}}_I; \overleftarrow{\mathbf{h}}_1]). \tag{5}$$

After we analyze existing attention-based NMT in detail, our intuition arises as follows. Ideally, with the source summarization in mind, after generating each target word y_t from the source contents $\mathbf{c}_t$, the decoder should keep track of (1) translated source contents by accumulating $\mathbf{c}_t$, and (2) untranslated source contents by subtracting $\mathbf{c}_t$ from the source summarization. However, such information is not well learned in practice, as there lacks explicit mechanisms to maintain translated and untranslated contents. Evidence shows that attention-based NMT still suffers from serious over- and under-translation problems (Tu et al., 2016; Tu et al., 2017b). Examples of under-translation are shown in Table 1a.

Another piece of evidence also shows that the decoder may lack a holistic view of the source information, as explained below. We conduct a pilot experiment by removing the initialization of the RNN decoder. If the "holistic" context is well exploited by the decoder, translation performance would significantly decrease without the initialization. As shown in Table 1b, however, translation performance only decreases slightly after we remove the initialization. This indicates NMT decoders do not make full use of source summarization, that the initialization only helps the prediction at the beginning of the sentence. We attribute the vanishing of such signals to the overloaded use of decoder states (e.g., LM, PAST, and FUTURE functionalities), and hence we propose to explicitly model the holistic source summarization by PAST and FUTURE contents at each decoding step.

3 Related Work

Our research is built upon an attention-based sequence-to-sequence model (Bahdanau et al., 2015), but is also related to coverage modeling, future modeling, and functionality separation. We discuss these topics in the following.

Coverage Modeling. Tu et al. (2016) and Mi et al. (2016) maintain a coverage vector to indicate which source words have been translated and which source words have not. These vectors are updated by accumulating attention probabilities at each decoding step, which provides an opportunity for the attention model to distinguish translated source words from untranslated ones. Viewing coverage vectors as a (soft) indicator of translated source contents, following this idea, we take one step further. We model translated and untranslated source contents by directly manipulating the attention vector (i.e., the source contents that are being translated) instead of attention probability (i.e., the probability of a source word being translated).

In addition, we explicitly model both translated (with PAST-RNN) and untranslated (with FUTURE-RNN) instead of using a single coverage vector to indicate translated source words. The difference with Tu et al. (2016) is that the PAST and FUTURE contents in our model are fed not only to the attention mechanism but also the decoder's states.

In the context of semantic-level coverage, Wang et al. (2016) propose a memory-enhanced decoder

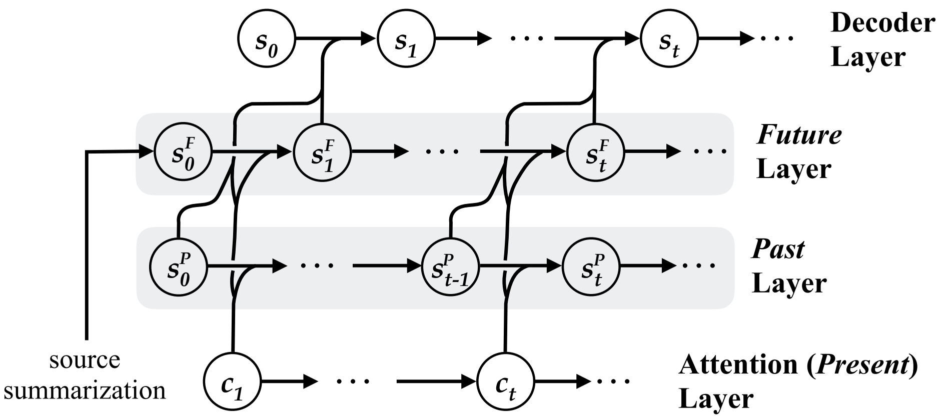


Figure 2: NMT decoder augmented with PAST and FUTURE layers.

and Meng et al. (2016) propose a memory-enhanced attention model. Both implement the memory with a Neural Turing Machine (Graves et al., 2014), in which the reading and writing operations are expected to erase translated contents and highlight untranslated contents. However, their models lack an explicit objective to guide such intuition, which is one of the key ingredients for the success in this work. In addition, we use two separate layers to explicitly model translated and untranslated contents, which is another distinguishing feature of the proposed approach.

Future Modeling. Standard neural sequence decoders generate target sentences from left to right, thus failing to estimate some desired properties in the *future* (e.g., the length of target sentence). To address this problem, actor-critic algorithms are employed to predict future properties (Li et al., 2017; Bahdanau et al., 2017), in their models, an interpolation of the actor (the standard generation policy) and the critic (a value function that estimates the future values) is used for decision making. Concerning the future generation at each decoding step, Weng et al. (2017) guide the decoder's hidden states to not only generate the current target word, but also predict the target words that remain untranslated. Along the direction of future modeling, we introduce a FUTURE layer to maintain the untranslated source contents, which is updated at each decoding step by subtracting the source content being translated (i.e., attention vector) from the last state (i.e., the untranslated source content so far).

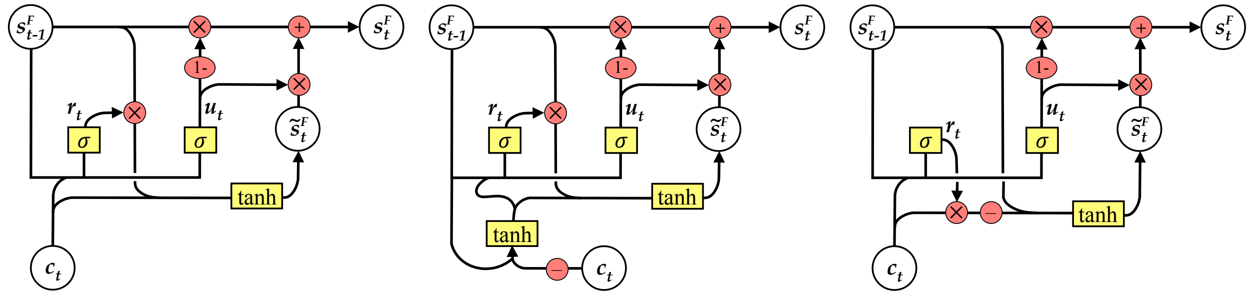
Functionality Separation. Recent work has revealed that the overloaded use of representations makes model training difficult, and such problems can be alleviated by explicitly separating these functions (Reed and Freitas, 2015; Ba et al., 2016; Miller et al., 2016; Gulcehre et al., 2016; Rocktäschel et al., 2017). For example, Miller et al. (2016) separate the functionality of look-up keys and memory contents in memory networks (Sukhbaatar et al., 2015). Rocktäschel et al. (2017) propose a *key-value-predict* attention model, which outputs three vectors at each step: the first is used to predict the next-word distribution; the second serves as the key for decoding; and the third is used for the attention mechanism. In this work, we further separate PAST and FUTURE functionalities from the decoder's hidden representations.

4 Modeling PAST and FUTURE for Neural Machine Translation

In this section, we describe how to separate PAST and FUTURE functions from decoding states. We introduce two additional RNN layers (Figure 2):

- FUTURE Layer (Section 4.1) encodes source contents to be translated.
- PAST Layer (Section 4.2) encodes translated source contents.

Let us take $\mathbf{y} = \{y_1, y_2, y_3, y_4\}$ as an example of the target sentence. The initial state of the FUTURE layer is a summarization of the whole source sentence, indicating that all source contents need to be translated. The initial state of the PAST layer is an all-zero vector, indicating no source content is yet



(a) GRU

(b) GRU-o

(c) GRU-i

Figure 3: Variants of activation functions for the FUTURE layer.

translated.

After $\mathbf{c}_1$ is obtained by the attention mechanism, we (1) update the FUTURE layer by "subtracting" $\mathbf{c}_1$ from the previous state, and (2) update the PAST layer state by "adding" $\mathbf{c}_1$ to the previous state. The two RNN states are updated as described above at every step of generating y_1, y_2, y_3, and y_4. In this way, at each time step, the FUTURE layer encodes source contents to be translated in the future steps, while the PAST layer encodes translated source contents up to the current step.

The advantages of the PAST and the FUTURE layers are two-fold. First, they provide coverage information, which is fed to the attention model and guides NMT systems to pay more attention to untranslated source contents. Second, they provide a holistic view of the source information, since we would anticipate "PAST + FUTURE = HOLISTIC." We describe them in detail in the rest of this section.

4.1 Modeling FUTURE

Formally, the FUTURE layer is a recurrent neural network (the first gray layer in Figure 2) , and its state at time step t is computed by

$$\mathbf{s}^F_t = \mathbb{F}(\mathbf{s}^F_{t-1}, \mathbf{c}_t), \tag{6}$$

where $\mathbb{F}$ is the activation function for the FUTURE layer. We have several variants of $\mathbb{F}$, aiming to better model the expected subtraction, as described in Section 4.1.1. The FUTURE RNN is initialized with the summarization of the whole source sentence, as computed by Equation 5.

When calculating attention context at time step t, we feed the attention model with the FUTURE state from the last time step, which encodes source contents to be translated. We rewrite Equation 4 as

$$\alpha_{t,i} = \text{softmax}\big(a(\mathbf{s}_{t-1}, \mathbf{h}_i, \mathbf{s}^F_{t-1})\big). \tag{7}$$

After obtaining attention context $\mathbf{c}_t$, we update FUTURE states via Equation 6, and feed both of them to decoder states:

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{c}_t, \mathbf{s}^F_t), \tag{8}$$

where $\mathbf{c}_t$ encodes the source context of the present translation, and $\mathbf{s}^F_t$ encodes the source context on the future translation.

4.1.1 Activation Functions for Subtraction

We design several variants of RNN activation functions to better model the subtractive operation (Figure 3):

GRU. A natural choice is the standard GRU[1], which learns subtraction directly from the data:

$$\begin{aligned}
\mathbf{s}^F_t &= \text{GRU}(\mathbf{s}^F_{t-1}, \mathbf{c}_t) && (9)\\
&= \mathbf{u}_t \cdot \mathbf{s}^F_{t-1} + (1 - \mathbf{u}_t) \cdot \tilde{\mathbf{s}}^F_t;\\
\tilde{\mathbf{s}}^F_t &= \tanh(U(\mathbf{r}_t \cdot \mathbf{s}^F_{t-1}) + W\mathbf{c}_t); && (10)\\
\mathbf{r}_t &= \sigma(U_r\mathbf{s}^F_{t-1} + W_r\mathbf{c}_t); && (11)\\
\mathbf{u}_t &= \sigma(U_u\mathbf{s}^F_{t-1} + W_u\mathbf{c}_t), && (12)
\end{aligned}$$

where $\mathbf{r}_t$ is a reset gate determining the combination of the input with the previous state, and $\mathbf{u}_t$ is an update gate defining how much of the previous state to keep around. The standard GRU uses a feed-forward

[1] Our work focuses on GRU, but can be applied to any RNN architectures such as LSTM.

neural network (Equation 10) to model the subtraction without any explicit operation, which may lead to the difficulty of the training.

In the following two variants, we provide GRU with explicit subtraction operations, which are inspired by the well known phenomenon that minus operation can be applied to the semantics of word embeddings (Mikolov et al., 2013).[2] Therefore we subtract the semantics being translated from the untranslated FUTURE contents at each decoding step.

GRU with Outside Minus (GRU-*o*). Instead of directly feeding $\mathbf{c}_t$ to GRU, we compute the current untranslated contents $\mathrm{M}(\mathbf{s}_{t-1}^F, \mathbf{c}_t)$ with an explicit minus operation, and then feed it to GRU:

$$\mathbf{s}_t^F = \mathrm{GRU}(\mathbf{s}_{t-1}^F, \mathrm{M}(\mathbf{s}_{t-1}^F, \mathbf{c}_t)); \tag{13}$$

$$\mathrm{M}(\mathbf{s}_{t-1}^F, \mathbf{c}_t) = \tanh(U_m \mathbf{s}_{t-1}^F - W_m \mathbf{c}_t). \tag{14}$$

GRU with Inside Minus (GRU-*i*). We can alternatively integrate a minus operation into the calculation of $\tilde{\mathbf{s}}_t^F$:

$$\tilde{\mathbf{s}}_t^F = \tanh(U \mathbf{s}_{t-1}^F - W(\mathbf{r_t} \cdot \mathbf{c}_t)). \tag{15}$$

Compared with Equation 10, the differences between GRU-*i* and standard GRU are:

1. Minus operation is applied to produce the energy of the intermediate candidate state $\tilde{\mathbf{s}}_t^F$;
2. The reset gate $\mathbf{r}_t$ is used to control the amount of information flowing from inputs instead of from the previous state $\mathbf{s}_{t-1}^F$.

Note that for both GRU-o and GRU-i, we leave enough freedom for GRU to decide the extent of integrating with subtraction operations. In other words, the information subtraction is "soft."

4.2 Modeling PAST

Formally, the PAST layer is another recurrent neural network (the second gray layer in Figure 2), and its state at time step t is calculated by:

$$\mathbf{s}_t^P = \mathrm{GRU}(\mathbf{s}_{t-1}^P, \mathbf{c}_t). \tag{16}$$

Initially, $\mathbf{s}_t^P$ is an all-zero vector, which denotes no source content is yet translated. We choose GRU as the activation function for the PAST layer, since the internal structure of GRU is in accord with the "addition" operation.

We feed the PAST state from last time step to both attention model and decoder state:

$$\alpha_{t,i} = \mathrm{softmax}\big(a(\mathbf{s}_{t-1}, \mathbf{h}_i, \mathbf{s}_{t-1}^P)\big); \tag{17}$$

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{c}_t, \mathbf{s}_{t-1}^P). \tag{18}$$

4.3 Modeling PAST and FUTURE

We integrate PAST and FUTURE layers together in our final model (Figure 2):

$$\alpha_{t,i} = \mathrm{softmax}\big(a(\mathbf{s}_{t-1}, \mathbf{h}_i, \mathbf{s}_{t-1}^F, \mathbf{s}_{t-1}^P)\big); \tag{19}$$

$$\mathbf{s}_t = f(\mathbf{s}_{t-1}, y_{t-1}, \mathbf{c}_t, \mathbf{s}_{t-1}^F, \mathbf{s}_{t-1}^P). \tag{20}$$

In this way, both the attention model and the decoder state are aware of what has, and what has not yet been translated.

4.4 Learning

We introduce additional loss functions to estimate the semantic subtraction and addition, which guide the training of the FUTURE layer and PAST layer, respectively.

Loss Function for Subtraction. As described above, the FUTURE layer models the future semantics in a declining way: $\mathbf{\Delta}_t^F = \mathbf{s}_{t-1}^F - \mathbf{s}_t^F \approx \mathbf{c}_t$. Since source and target sides contain equivalent semantic information in machine translation (Tu et al., 2017a): $\mathbf{c}_t \approx \mathrm{E}(y_t)$, we directly measure the consistence between $\mathbf{\Delta}_t^F$ and $\mathrm{E}(y_t)$, which guides the subtraction to learn the right thing:

$$loss(\mathbf{\Delta}_t^F, \mathrm{E}(y_t)) = -\log \frac{\exp\big(l(\mathbf{\Delta}_t^F, \mathrm{E}(y_t))\big)}{\sum_y \exp\big(l(\mathbf{\Delta}_t^F, \mathrm{E}(y))\big)};$$

$$l(\mathbf{u}, \mathbf{v}) = \mathbf{u}^\top W \mathbf{v} + b.$$

In other words, we explicitly guide the FUTURE layer by this subtractive loss, expecting $\mathbf{\Delta}_t^F$ to be discriminative of the current word y_t.

Loss Function for Addition. Likewise, we introduce another loss function to measure the information incrementation of the PAST layer. Notice that $\mathbf{\Delta}_t^P = \mathbf{s}_t^P - \mathbf{s}_{t-1}^P \approx \mathbf{c}_t$, is defined similarly to $\mathbf{\Delta}_t^F$ except a minus sign. In this way, we can reasonably assume the FUTURE and PAST layers are indeed doing subtraction and addition, respectively.

[2] E("King") – E("Man") = E("Queen") – E("Woman"), where $\mathrm{E}(\cdot)$ is the embedding of a word.

Training Objective. We train the proposed model $\hat{\theta}$ on a set of training examples $\{[\mathbf{x}^n, \mathbf{y}^n]\}_{n=1}^N$, and the training objective is

$$\hat{\theta} = \arg\min_{\theta} \sum_{n=1}^{N} \sum_{t=1}^{|\mathbf{y}|} \Big\{ \underbrace{-\log P(y_t|y_{<t}, \mathbf{x}; \theta)}_{\textit{neg. log-likelihood}} + \underbrace{loss(\mathbf{\Delta}_t^F, \mathrm{E}(y_t)|\theta)}_{\textrm{FUTURE } \textit{loss}} + \underbrace{loss(\mathbf{\Delta}_t^P, \mathrm{E}(y_t)|\theta)}_{\textrm{PAST } \textit{loss}} \Big\}.$$

5 Experiments

Dataset. We conduct experiments on Chinese-English (Zh-En), German-English (De-En), and English-German (En-De) translation tasks.

For Zh-En, the training set consists of 1.6m sentence pairs, which are extracted from the LDC corpora[3]. The NIST 2003 (MT03) dataset is our development set; the NIST 2002 (MT02), 2004 (MT04), 2005 (MT05), 2006 (MT06) datasets are test sets. We also evaluate the alignment performance on the standard benchmark of Liu and Sun (2015), which contains 900 manually aligned sentence pairs. We measure the alignment quality with the alignment error rate (Och and Ney, 2003).

For De-En and En-De, we conduct experiments on the WMT17 (Bojar et al., 2017) corpus. The dataset consists of 5.6M sentence pairs. We use `newstest2016` as our development set, and `newstest2017` as our testset. We follow Sennrich et al. (2017a) to segment both German and English words into subwords using byte-pair encoding (Sennrich et al., 2016, BPE).

We measure the translation quality with BLEU scores (Papineni et al., 2002). We use the `multi-bleu` script for Zh-En[4], and the `multi-bleu-detok` script for De-En and En-De[5].

Training Details. We use the `Nematus`[6] (Sennrich et al., 2017b), implementing a baseline translation system, RNNSEARCH. For Zh-En, we limit the vocabulary size to 30K. For De-En and En-De, the number of joint BPE operations is 90,000. We use the total BPE vocabulary for each side.

We tie the weights of the target-side embeddings and the output weight matrix (Press and Wolf, 2017) for De-En. All out-of-vocabulary words are mapped to a special token `UNK`.

We train each model with sentences lengths of up to 50 words in the training data. The dimension of word embeddings is 512, and all hidden sizes are 1024. In training, we set the batch size to 80 for Zh-En, and 64 for De-En and En-De. We set the beam size to 12 in testing. We shuffle the training corpus after each epoch.

We use Adam (Kingma and Ba, 2014) with annealing (Denkowski and Neubig, 2017) as our optimization algorithm. We set the initial learning rate as 0.0005, which halves when the validation cross-entropy does not decrease.

For the proposed model, we use the same setting as the baseline model. The FUTURE and PAST layer sizes are 1024. We employ a two-pass strategy for training the proposed model, which has proven useful to ease training difficulty when the model is relatively complicated (Shen et al., 2016; Wang et al., 2017; Wang et al., 2018). Model parameters shared with the baseline are initialized by the baseline model.

5.1 Results on Chinese-English

We first evaluate the proposed model on the Chinese-English translation and alignment tasks.

5.1.1 Translation Quality

Table 2 shows the translation performances on Chinese-English. Clearly the proposed approach significantly improves the translation quality in all cases, although there are still considerable differences among different variants.

FUTURE Layer. (Rows 1-4). All the activation functions for the FUTURE layer obtain BLEU score improvements: GRU +0.52, GRU-o +1.03, and GRU-i +1.12. Specifically, GRU-o is better than

[3] The corpora includes LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06

[4] `https://github.com/moses-smt/mosesdecoder/blob/master/scripts/generic/multi-bleu.perl`

[5] `https://github.com/EdinburghNLP/nematus/blob/master/data/multi-bleu-detok.perl`

[6] `https://github.com/EdinburghNLP/nematus`

#	**Model**	Dev	MT02	MT04	MT05	MT06	Avg.	Δ
0	RNNSearch	35.90	36.84	37.16	34.17	31.56	35.13	-
1	+ Frnn (GRU)	36.11	36,94	38.52	34.58	32.08	35.65	+0.52
2	+ Frnn (GRU-o)	36.70	37.81	38.59	35.10	32.60	36.16	+1.03
3	+ Frnn (GRU-i)	36.98	38.24	38.66	34.68	32.66	36.24	+1.12
4	+ Frnn (GRU-i) + Loss	37.15	38.80	39.13	35.79	33.75	36.92	+1.80
5	+ Prnn	36.90	37.62	39.04	35.24	32.80	36.32	+1.19
6	+ Prnn + Loss	36.95	39.06	39.55	35.05	33.80	36.88	+1.76
7	+ Frnn (GRU-i) + Prnn	37.44	37.26	39.10	35.29	32.78	36.37	+1.25
8	+ Frnn (GRU-i) + Prnn + Loss	**37.90**	**39.65**	**40.37**	**36.75**	**34.55**	**37.84**	**+2.71**
9	RNNSearch-2dec	35.56	36.74	37.38	34.09	31.82	35.12	-0.01
10	RNNSearch-3dec	36.07	37.64	37.62	34.14	32.73	35.64	+0.51
11	Coverage (Tu et al., 2016)	36.56	37.54	38.39	34.47	32.38	35.87	+0.74

Table 2: Case-insensitive BLEU on Chinese-English Translation. "Loss" means applying loss functions for Future layer (Frnn) and Past layer (Prnn).

a regular GRU for its minus operation, and GRU-i is the best, which shows that our elaborately designed architecture is more proper for modeling the decreasing phenomenon of the future semantics.

Adding subtractive loss gives an extra 0.68 BLEU score improvement, which indicates that adding g is beneficial guided objective for Frnn to learn the minus operation.

Past Layer. (Rows 5-6). We observe the same trend on introducing the Past layer: using it alone achieves a significant improvement (+1.19), and with the additional objective, it further improves the translation performance (+0.57).

Stacking the Future and the Past Together. (Rows 7-8). The model's final architecture outperforms our intermediate models (1-6) by combining Frnn and Prnn. By further separating the functionaries of past content modeling and language modeling into different neural components, the final model is more flexible, obtaining a 0.91 BLEU improvement over the best intermediate model (Row 4) and an improvement of 2.71 BLEU points over the RNNSearch baseline.

Comparison with Other Work. (Rows 9-11). We also conduct experiments with multi-layer decoders (Wu et al., 2016) to see whether the NMT system can automatically model the translated and untranslated contents with additional decoder layers (Rows 9-10). However, we find that the performance is not improved using a two-layer decoder (Row 9), until a deeper version (three-layer decoder, Row 10) is used. This indicates that enhancing performance by simply adding more RNN layers into the decoder without any explicit instruction is non-trivial, which is consistent with the observation of Britz et al. (2017).

Our model also outperforms the word-level Coverage (Tu et al., 2016), which considers the coverage information of the source words independently. Our proposed model can be regarded as a high-level coverage model, which captures higher level coverage information, and gives more specific signals for the decision of attention and target prediction. Our model is more deeply involved in generating target words, by being fed not only to the attention model as in Tu et al. (2016), but also to the decoder state.

5.1.2 Subjective Evaluation

Following Tu et al. (2016), we conduct subjective evaluations to validate the benefit of modeling the Past and the Future (Table 3). Four human evaluators are asked to evaluate the translations of 100 source sentences, which are randomly sampled from the testsets without knowing from which system the translation is selected. For the Base system, 1.7% of the source words are over-translated and 8.8% are under-translated. Our proposed model alleviates these problems by explicitly modeling the dynamic

System	Architecture	De-En		En-De	
		Dev	Test	Dev	Test
Rikters et al. (2017)	cGRU + BPE + dropout	31.9	27.2	27.4	21.0
	+ name entity forcing + *synthetic data*	36.9	29.0	30.9	22.7
Escolano et al. (2017)	Char2Char + Rescoring with inverse model	32.1	-	27.0	-
	+ *synthetic data*	-	28.1	-	21.2
Sennrich et al. (2017a)	cGRU + BPE + *synthetic data*	38.0	32.0	32.2	26.1
this work	Base	32.0	27.8	28.3	23.3
	Coverage	32.2	28.7	28.9	23.6
	Ours	33.5	29.7	29.5	24.3

Table 5: Results of De-En and En-De "*synthetic data*" denotes additional 10M monolingual sentences, which is not used in our work.

Model	Over-Trans		Under-Trans	
	Ratio	Δ	Ratio	Δ
Base	1.7%	–	8.8%	–
Coverage	1.5%	-11.8%	7.7%	-12.4%
Ours	1.6%	-5.9%	5.7%	-35.2%

Table 3: Subjective evaluation on over- and under-translation for Chinese-English. "Ratio" denotes the percentage of source words which are over- or under-translated, "Δ" indicates relative improvement. "Base" denotes RNNSearch and "Ours" denotes "+ Frnn (GRU-i) + Prnn + Loss".

source contents by the Past and the Future layers, reducing 11.8% and 35.2% of over-translation and under-translation errors, respectively. The proposed model is especially effective for alleviating the under-translation problem, which is a more serious translation problem for NMT systems, and is mainly caused by lacking necessary coverage information (Tu et al., 2016).

5.1.3 Alignment Quality

Table 4 lists the alignment performances of our proposed model. We find that the Coverage model does improve attention model. But our model can produce much better alignments compared to the word level coverage (Tu et al., 2016). Our model distinguishes the Past and Future directly, which is a higher level coverage mechanism than the word coverage model.

Model	AER	Δ
Base	39.73	–
Coverage	38.73	-1.00
Ours	35.90	-3.83

Table 4: Evaluation of the alignment quality. The lower the score, the better the alignment quality.

5.2 Results on German-English

We also evaluate our model on the WMT17 benchmarks for both De-En and En-De. As shown in Table 5, our baseline gives comparable BLEU scores to the state-of-the-art NMT systems of WMT17. Our proposed model improves the strong baseline on both De-En and En-De. This shows that our proposed model works well across different language pairs. Rikters et al. (2017) and Sennrich et al. (2017a) obtain higher BLEU scores than our model, because they use additional large scale synthetic data (about 10M) for training. It maybe unfair to compare our model to theirs directly.

5.3 Analysis

We conduct analyses on Zh-En, to better understand our model from different perspectives.

Parameters and Speeds. As shown in Table 6, the baseline model (Base) has 80M parameters. A single Future or Past layer introduces 15M to 17M parameters, and the corresponding objective introduces 18M parameters. In this work, the most complex model introduces 65M parameters, which

Model	**#Para.**	**Speed**	
		Train	Test
Base	80M	42.59	2.05
+ Frnn (GRU)	96M	31.91	1.99
+ Frnn (GRU-o)	97M	30.88	1.93
+ Frnn (GRU-i)	97M	31.06	1.95
+ Loss	115M	29.40	1.68
+ Prnn	95M	32.01	1.98
+ Loss	113M	29.60	1.69
+ Frnn + Prnn	110M	26.01	1.88
+ Loss	145M	22.94	1.52
RNNSearch-2dec	93M	39.89	2.00
RNNSearch-3dec	105M	35.57	1.82
Coverage	80M	40.48	1.90

Table 6: Statistics of parameters, training and testing speeds (sentences per second).

Model	**Loss used in**		**BLEU**	**Δ**
	Train	*Test*		
Base	–	–	35.13	–
Ours	×	×	36.37	+1.25
	✓	×	36.85	+1.72
	✓	✓	37.84	+2.71

Table 7: Contributions of loss functions from parameter training ("*Train*") and reranking of candidates in testing ("*Test*").

Initialize Frnn with ...	**BLEU**
Source Summarization	36.24
All-Zero Vector	35.81

Table 8: Influence of initialization of Frnn layer (GRU-i)

leads to a relatively slower training speed. However, our proposed model does not significant slow down the decoding speed. The most time consuming part is the calculation of the subtraction and addition losses. As we show in the next paragraph, our system works well by only using the losses in training, which further improve the decoding speed of our model.

Effectiveness of Subtraction and Addition Loss. Adding subtraction and addition loss functions helps twofold: (1) guiding the training of the proposed subtraction and addition operation; (2) enabling better reranking of generated candidates in testing. Table 7 lists the improvements from the two perspectives. When applied only in training, the two loss functions lead to an improvement of 0.48 BLEU points by better modeling subtraction and addition operations. On top of that, reranking with Future and Past loss scores in testing further improves the performance by +0.99 BLEU points.

Initialization of the Future Layer. The baseline model does not obtain abundant accuracy improvement by feeding the source summarization into the decoder (Table 1). We also experiment to not feed the source summarization into the decoder of the proposed model, which leads to a significant BLEU score drop on Zh-En. This shows that our proposed model better use the source summarization with explicitly modeling the Future compared to the conventional encoder-decoder baseline.

Case Study. We also compare the translation cases for the baseline, word level coverage and our proposed models. As shown in Table 9, our baseline system suffers from the over-translation problems (case 1), which is consistent with the results of human evaluation (Section 3). The Base system also incorrectly translates "the royal family" into "the people of hong kong", which is totally irrelevant here. We attribute the former case to the lack of untranslated future modeling, and the latter one to the overloaded use of the decoder state where the language modeling of the decoder leads to the fluent but wrong predictions. In contrast, the proposed approach almost addresses the errors in these cases.

6 Conclusion

Modeling source contents well is crucial for encoder-decoder based NMT systems. However, current NMT models suffer from distinguishing translated and untranslated translation contents, due to the lack of explicitly modeling past and future translations. In this paper, we separate Past and Future functionalities from decoder states, which can maintain a dynamic yet holistic view of the source content at each decoding step. Experimental results show that the proposed approach signifi-

Source	布什 还 表示 , 应 巴基斯坦 和 印度 政府 的 邀请 , 他 将 于 3月份 对 巴基斯坦 和 印度 进行 访问 。
Reference	bush also said that **at the invitation of the pakistani and indian governments** , he would visit pakistan and india in march .
Base	bush also said that he would visit pakistan and india in march .
Coverage	bush also said that at the invitation of pakistan and india , he will visit pakistan and india in march .
Ours	bush also said that at the invitation of the pakistani and indian governments , he will visit pakistan and india in march .
Source	所以 有 不少 人 认为 说 , 如果 是 这样 的 话 , 对 皇室 、 对 日本 的 社会 也 是 会 有 很 大 的 影响 的 。
Reference	therefore , many people say that it will have a great impact on **the royal family** and japanese society .
Base	therefore , many people are of the view that if this is the case , it will also have a great impact on *the people of hong kong* and the japanese society .
Coverage	therefore , many people think that if this is the case , there will be great impact on the royal and japanese society .
Ours	therefore , many people think that if this is the case , it will have a great impact on the royal and japanese society .

Table 9: Comparison on Translation Examples. We italicize some *translation errors* and highlight the **correct ones** in bold.

cantly improves translation performances across different language pairs. With better modeling of past and future translations, our approach performs much better than the standard attention-based NMT, reducing the errors of under and over translations.

7 Acknowledgement

We would like to thank the anonymous reviewers as well as the Action Editor, Philipp Koehn, for insightful comments and suggestions. Shujian Huang is the corresponding author. This work is supported by the National Science Foundation of China (No. 61672277, 61772261), the Jiangsu Provincial Research Foundation for Basic Research (No. BK20170074).

References

Jimmy Ba, Geoffrey Hinton, Volodymyr Mnih, Joel Z Leibo, and Catalin Ionescu. 2016. Using fast weights to attend to the recent past. In *NIPS 2016*.

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

Dzmitry Bahdanau, Philemon Brakel, Kelvin Xu, Anirudh Goyal, Ryan Lowe, Joelle Pineau, Aaron Courville, and Yoshua Bengio. 2017. An actor-critic algorithm for sequence prediction. In *ICLR 2017*.

Ondřej Bojar, Christian Buck, Rajen Chatterjee, Christian Federmann, Yvette Graham, Barry Haddow, Matthias Huck, Antonio Jimeno Yepes, Philipp Koehn, and Julia Kreutzer. 2017. Proceedings of the second conference on machine translation. In *Proceedings of the Second Conference on Machine Translation*. Association for Computational Linguistics.

Denny Britz, Anna Goldie, Minh-Thang Luong, and Quoc Le. 2017. Massive exploration of neural machine translation architectures. In *EMNLP 2017*.

Kyunghyun Cho, Bart van Merrienboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2014. Learning phrase representations using RNN encoder–decoder for statistical machine translation. In *EMNLP 2014*.

Michael Denkowski and Graham Neubig. 2017. Stronger baselines for trustable results in neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*.

Carlos Escolano, Marta R. Costa-jussà, and José A. R. Fonollosa. 2017. The TALP-UPC neural machine translation system for German/Finnish-English using the inverse direction model in rescoring. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers*.

Alex Graves, Greg Wayne, and Ivo Danihelka. 2014. Neural turing machines. *arXiv:1410.5401*.

Caglar Gulcehre, Sarath Chandar, Kyunghyun Cho, and Yoshua Bengio. 2016. Dynamic neural turing machine with soft and hard addressing schemes. *arXiv:1607.00036*.

Sepp Hochreiter and Jürgen Schmidhuber. 1997. Long short-term memory. *Neural computation*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *EMNLP 2013*.

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

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

Yang Liu and Maosong Sun. 2015. Contrastive unsupervised word alignment with non-local features. In *AAAI 2015*.

Thang Luong, Hieu Pham, and Christopher D. Manning. 2015. Effective approaches to attention-based neural machine translation. In *EMNLP 2015*.

Fandong Meng, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Interactive attention for neural machine translation. In *COLING 2016*.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage embedding models for neural machine translation. *EMNLP 2016*.

Tomas Mikolov, Greg Corrado, Kai Chen, and Jeffrey Dean. 2013. Efficient estimation of word representations in vector space. *ICLR 2013*.

Alexander Miller, Adam Fisch, Jesse Dodge, Amir-Hossein Karimi, Antoine Bordes, and Jason Weston. 2016. Key-value memory networks for directly reading documents. In *EMNLP 2016*.

Franz Josef Och and Hermann Ney. 2003. A Systematic Comparison of Various Statistical Alignment Models. *Computational Linguistics*.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. BLEU: a method for automatic evaluation of machine translation. In *ACL 2002*.

Ofir Press and Lior Wolf. 2017. Using the output embedding to improve language models. In *EACL 2017*.

Scott Reed and Nando De Freitas. 2015. Neural programmer-interpreters. *Computer Science*.

Matīss Rikters, Chantal Amrhein, Maksym Del, and Mark Fishel. 2017. C-3MA: Tartu-Riga-Zurich translation systems for WMT17.

Tim Rocktäschel, Johannes Welbl, and Sebastian Riedel. 2017. Frustratingly short attention spans in neural language modeling. In *ICLR 2017*.

Mike Schuster and Kuldip K. Paliwal. 1997. Bidirectional recurrent neural networks. *IEEE Transactions on Signal Processing*.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. *Computer Science*.

Rico Sennrich, Alexandra Birch, Anna Currey, Ulrich Germann, Barry Haddow, Kenneth Heafield, Antonio Valerio Miceli Barone, and Philip Williams. 2017a. The University of Edinburgh's Neural MT Systems for WMT17. In *Proceedings of the Second Conference on Machine Translation, Volume 2: Shared Task Papers in ACL*.

Rico Sennrich, Orhan Firat, Kyunghyun Cho, Alexandra Birch, Barry Haddow, Julian Hitschler, Marcin Junczys-Dowmunt, Samuel Läubli, Antonio Valerio Miceli Barone, Jozef Mokry, and Maria Nadejde. 2017b. Nematus: a toolkit for neural machine translation. In *EACL 2017*.

Shiqi Shen, Yong Cheng, Zhongjun He, Wei He, Hua Wu, Maosong Sun, and Yang Liu. 2016. Minimum risk training for neural machine translation. In *ACL 2016*.

Sainbayar Sukhbaatar, Arthur Szlam, Jason Weston, and Rob Fergus. 2015. End-to-end memory networks. In *NIPS 2015*.

Ilya Sutskever, Oriol Vinyals, and Quoc V. Le. 2014. Sequence to sequence learning with neural networks. In *NIPS 2014*.

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

Zhaopeng Tu, Yang Liu, Zhengdong Lu, Xiaohua Liu, and Hang Li. 2017a. Context gates for neural machine translation. *Transactions of the Association for Computational Linguistics*.

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017b. Neural machine translation with reconstruction. In *AAAI 2017*.

Mingxuan Wang, Zhengdong Lu, Hang Li, and Qun Liu. 2016. Memory-enhanced decoder for neural machine translation. In *EMNLP 2016*.

Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. 2017. Neural machine translation advised by statistical machine translation. In *AAAI 2017*.

Longyue Wang, Zhaopeng Tu, Shuming Shi, Tong Zhang, Yvette Graham, and Qun Liu. 2018. Translating pro-drop languages with reconstruction models. In *AAAI 2018*.

Rongxiang Weng, Shujian Huang, Zaixiang Zheng, Xin-Yu Dai, and Jiajun Chen. 2017. Neural machine translation with word predictions. In *EMNLP 2017*.

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:1609.08144*.