Dynamic Past and Future for Neural Machine Translation

Zaixiang Zheng
Nanjing University
zhengzx@smail.nju.edu.cn

Shujian Huang
Nanjing University
huangsj@nju.edu.cn

Zhaopeng Tu
Tencent AI Lab
zptu@tencent.com

Xin-Yu Dai
Nanjing University
dxy@nju.edu.cn

Jiajun Chen
Nanjing University
chenjj@nju.edu.cn

Abstract

Previous studies have shown that neural machine translation (NMT) models can benefit from explicitly modeling translated (PAST) and untranslated (FUTURE) source contents as recurrent states (Zheng et al., 2018). However, this less interpretable recurrent process hinders its power to model the dynamic updating of PAST and FUTURE contents during decoding. In this paper, we propose to model the *dynamic principles* by explicitly separating source words into groups of translated and untranslated contents through parts-to-wholes assignment. The assignment is learned through a novel variant of routing-by-agreement mechanism (Sabour et al., 2017), namely *Guided Dynamic Routing*, where the translating status at each decoding step *guides* the routing process to assign each source word to its associated group (i.e., translated or untranslated content) represented by a capsule, enabling translation to be made from holistic context. Experiments show that our approach achieves substantial improvements over both RNMT and Transformer by producing more adequate translations. Extensive analysis demonstrates that our method is highly interpretable, which is able to recognize the translated and untranslated contents as expected.[1]

1 Introduction

Neural machine translation (NMT) generally adopts an *attentive encoder-decoder* framework (Sutskever et al., 2014; Vaswani et al., 2017), where the encoder maps a source sentence into a sequence of contextual representations (*source contents*), and the decoder generates a target sentence word-by-word based on part of the source content assigned by an attention model (Bahdanau et al., 2015). Like human translators, NMT systems should have the ability to know the relevant source-side context for the current word (PRESENT), as well as recognize what parts in the source contents have been translated (PAST) and what parts have not (FUTURE), at each decoding step. Accordingly, the PAST, PRESENT and FUTURE are three *dynamically* changing states during the whole translation process.

Previous studies have shown that NMT models are likely to face the illness of inadequate translation (Kong et al., 2019), which is usually embodied in over- and under-translation problems (Tu et al., 2016, 2017). This issue may be attributed to the poor ability of NMT of recognizing the dynamic translated and untranslated contents. To remedy this, Zheng et al. (2018) first demonstrate that explicitly tracking PAST and FUTURE contents helps NMT models alleviate this issue and generate better translation. In their work, the running PAST and FUTURE contents are modeled as recurrent states. However, the recurrent process is still non-trivial to determine which parts of the source words are the PAST and which are the FUTURE, and to what extent the recurrent states represent them respectively, this less interpretable nature is probably not the best way to model and exploit the dynamic PAST and FUTURE.

We argue that an explicit separation of the source words into two groups, representing PAST and FUTURE respectively (Figure 1), could be more beneficial not only for easy and direct recognition of the translated and untranslated source contents, but also for better interpretation of model's behavior of the recognition. We formulate the explicit separation as a procedure of parts-to-wholes assignment: the representation of each source words (parts) should be assigned to its associated group of either PAST or FUTURE (wholes).

In this paper, we implement this idea using Cap-

[1] Codes are released at https://github.com/zhengzx-nlp/dynamic-nmt.

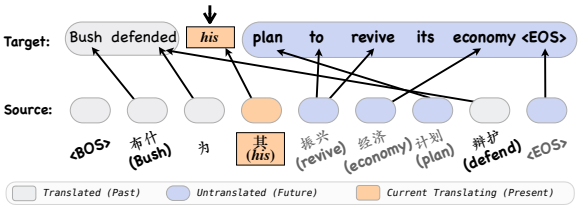


Figure 1: An example of separation of PAST and FUTURE in machine translation. When generating the current translation "his", the source tokens "⟨BOS⟩", "布什(Bush)" and phrase "为...辩护(defend)" are the translated contents (PAST), while the remaining tokens are untranslated contents (FUTURE).

sule Network (Hinton et al., 2011) with routing-by-agreement mechanism (Sabour et al., 2017), which has demonstrated its appealing strength of solving the problem of parts-to-wholes assignment (Hinton et al., 2018; Gong et al., 2018; Dou et al., 2019; Li et al., 2019), to model the separation of the PAST and FUTURE:

1. We first cast the PAST and FUTURE source contents as two groups of capsules.
2. We then design a novel variant of the routing-by-agreement mechanism, called *Guided Dynamic Routing* (GDR), which is *guided* by the current translating status at each decoding step to assign each source word to its associated capsules by assignment probabilities for several routing iterations.
3. Finally, the PAST and FUTURE capsules accumulate their expected contents from representations, and are fed into the decoder to provide a time-dependent holistic view of context to decide the prediction.

In addition, two auxiliary learning signals facilitate GDR's acquiring of our expected functionality, other than implicit learning within the training process of the NMT model.

We conducted extensive experiments and analysis to verify the effectiveness of our proposed model. Experiments on Chinese-to-English, English-to-German, and English-to-Romanian show consistent and substantial improvements over the Transformer (Vaswani et al., 2017) or RNMT (Bahdanau et al., 2015). Visualized evidence proves that our approach does acquire the expected ability to separate the source words into PAST and FUTURE, which is highly interpretable. We also observe that our model does alleviate the inadequate translation problem: Human subjective evaluation reveals that our model produces more adequate and high-quality translations than Transformer. Length analysis regarding source sentences shows that our model generates not only longer but also better translations.

2 Neural Machine Translation

Neural models for sequence-to-sequence tasks such as machine translation often adopt an *encoder-decoder* framework. Given a source sentence $\mathbf{x} = \langle x_1, \ldots, x_I \rangle$, a NMT model learns to predict a target sentence $\mathbf{y} = \langle y_1, \ldots, y_T \rangle$ by maximizing the conditional probabilities $p(\mathbf{y}|\mathbf{x}) = \prod_{t=1}^{T} p(y_t|y_{<t}, \mathbf{x})$. Specifically, an encoder first maps the source sentence into a sequence of encoded representations:

$$\mathbf{h} = \langle \boldsymbol{h}_1, \ldots, \boldsymbol{h}_I \rangle = f_e(\mathbf{x}), \quad (1)$$

where f_e is the encoder's transformation function. Given the encoded representations of the source words, a decoder generates the sequence of target words $\mathbf{y}$ autoregressively:

$$\boldsymbol{z}_t = f_d(y_{<t}, \boldsymbol{a}_t), \quad (2)$$

$$p(y_t|y_{<t}, \mathbf{x}) = \mathrm{softmax}(E(y_t)^\top \boldsymbol{z}_t), \quad (3)$$

where $E(y_t)$ is the embedding of y_t. The current word is predicted based on the decoder state $\boldsymbol{z}_t$. f_d is the transformation function of decoder, which determines $\boldsymbol{z}_t$ based on the target translation trajectory $y_{<t}$, and the lexical-level source content $\boldsymbol{a}_t$ that is most relevant to PRESENT translation by an attention model (Bahdanau et al., 2015). Ideally, with all the source encoded representations in the encoder, NMT models should be able to update translated and untranslated source contents and keep them in mind. However, most of existing NMT models lack an explicit functionality to maintain the translated and untranslated contents, failing to distinguish the source words being of either PAST or FUTURE (Zheng et al., 2018), which is likely to suffer from severe inadequate translation problem (Tu et al., 2016; Kong et al., 2019).

3 Approach

Motivation Our intuition arises straightforwardly: if we could tell the translated and untranslated source contents apart by directly separating the source words into PAST and FUTURE categories at each decoding step, the PRESENT translation could benefit from the dynamically holistic context (i.e., PAST+ PRESENT+ FUTURE). To this

purpose, we should design a mechanism by which each word could be recognized and assigned to a distinct category, i.e., PAST or FUTURE contents, subject to the translation status at present. This procedure can be seen as a parts-to-wholes assignment, in which the encoder hidden states of the source words (parts) are supposed to be assigned to either PAST or FUTURE (wholes).

Capsule network (Hinton et al., 2011) has shown its capability of solving the problem of assigning parts to wholes (Sabour et al., 2017). A capsule is a vector of neurons which represents different properties of the same entity from the input (Sabour et al., 2017). The functionality relies on a fast iterative process called routing-by-agreement, whose basic idea is to iteratively refine the proportion of how much a part should be assigned to a whole, based on the agreement between the part and the whole (Dou et al., 2019). Therefore, it is appealing to investigate if this mechanism could be employed for our intuition.

3.1 Guided Dynamic Routing (GDR)

Dynamic routing (Sabour et al., 2017) is an implementation of routing-by-agreement, where it runs intrinsically without any external guidance. However, what we expect is a mechanism driven by the decoding status at present. Here we propose a variant of dynamic routing mechanism called *Guided Dynamic Routing* (GDR), where the routing process is *guided* by the translating information at each decoding step (Figure 2).

Formally, we cast the source encoded representations $\mathbf{h}$ of I source words to be input capsules, while we denote $\mathbf{\Omega}$ as output capsules, which consist of J entries. Initially, we assume that $J/2$ of them ($\mathbf{\Omega}^P$) represent the PAST contents, and the rest $J/2$ capsules ($\mathbf{\Omega}^F$) represent the FUTURE:

$$\mathbf{\Omega}^P = \langle \Omega_1^P, \cdots, \Omega_{J/2}^P \rangle, \quad \mathbf{\Omega}^F = \langle \Omega_1^F, \cdots, \Omega_{J/2}^F \rangle.$$

where each capsule is represented by a d_c-dimension vector. We assemble these PAST and FUTURE capsules together, which are expected to competing for source information, i.e., we now have $\mathbf{\Omega} = \mathbf{\Omega}^P \cup \mathbf{\Omega}^F$. We will describe how to teach these capsules to retrieve their relevant parts from source contents in the Section 3.3. ***Note*** that we employ GDR at every decoding step t to obtain the time-dependent PAST and FUTURE and omit the subscript t for simplicity.

In the dynamic routing process, each vector output of capsule j is calculated with a non-linear

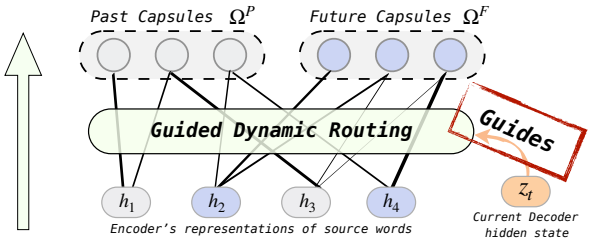


Figure 2: Illustration of the Guided Dynamic Routing.

squashing function (Sabour et al., 2017):

$$\Omega_j = \frac{\|\boldsymbol{s}_j\|^2}{1+\|\boldsymbol{s}_j\|^2}\frac{\boldsymbol{s}_j}{\|\boldsymbol{s}_j\|}, \quad \boldsymbol{s}_j = \sum_i^I c_{ij}\boldsymbol{v}_{ij}, \quad (4)$$

where $\boldsymbol{s}_j$ is the accumulated input of capsule Ω_j, which is a weighted sum over all *vote vectors* $\boldsymbol{v}_{ij}$. $\boldsymbol{v}_{ij}$ is transformed from the input capsule $\boldsymbol{h}_i$:

$$\boldsymbol{v}_{ij} = \boldsymbol{W}_j \boldsymbol{h}_i, \quad (5)$$

where $\boldsymbol{W}_j \in \mathbb{R}^{d \times d_c}$ is a trainable matrix for j-th output capsule[2]. c_{ij} is the assignment probability (i.e. the agreement) that is determined by the iterative dynamic routing. The assignment probabilities $c_{i\cdot}$ associated with each input capsule $\boldsymbol{h}_i$ sum to 1: $\sum_j c_{ij} = 1$, and are computed by:

$$c_{ij} = \mathrm{softmax}(b_{ij}), \quad (6)$$

where routing logit b_{ij} is initialized as all 0s, which measures the degree that $\boldsymbol{h}_i$ should be sent to Ω_j. The initial assignment probabilities are then iteratively updated by measuring the agreement between the vote vector $\boldsymbol{v}_{ij}$ and capsule Ω_j by an MLP, considering the current decoding state $\boldsymbol{z}_t$:

$$b_{ij} \leftarrow b_{ij} + \boldsymbol{w}^\top \tanh(\boldsymbol{W}_b[\boldsymbol{z}_t; \boldsymbol{v}_{ij}; \Omega_j]), \quad (7)$$

where $\boldsymbol{W}_b \in \mathbb{R}^{d+d_c*2}$ and $\boldsymbol{w} \in \mathbb{R}^{d_c}$ are learnable parameters. Instead of using simple scalar product, i.e., $b_{ij} = \boldsymbol{v}_{ij}^\top \Omega_j$ (Sabour et al., 2017), which could not consider the current decoding state as a condition signal, we resort to the MLP to take $\boldsymbol{z}_i$ into account inspired by MLP-based attention mechanism (Bahdanau et al., 2015; Luong et al., 2015). That is why we call it "guided" dynamic routing.

[2]Note that unlike Sabour et al. (2017), where each pair of input capsule i and output capsule j has a distinct transformation matrix $\boldsymbol{W}_{ij}$ as their numbers are predefined ($I \times J$ transformation matrices in total), here we share the transformation matrix $\boldsymbol{W}_j$ of output capsule j among all the input capsules due to the varied amount of the source words. So there are J transformation matrices in our model.

Algorithm 1 Guided Dynamic Routing (GDR)

Input: Encoder hidden state $\mathbf{h}$, current decoding hidden state $\boldsymbol{z}_t$, and number of routing iterations r.
Output: PAST, FUTURE, and redundant capsules.
procedure: GDR($\mathbf{h}$, $\boldsymbol{z}_t$, r)

$\forall i \in \mathbf{h}, j \in \boldsymbol{\Omega} : b_{ij} \leftarrow 0, \boldsymbol{v}_{ij} \leftarrow \boldsymbol{W}_j \boldsymbol{h}_i$ ▷ *Initializing routing logits, and vote vectors.*
for r iterations **do**
  $\forall i \in \mathbf{h}, j \in \boldsymbol{\Omega}$: Compute assign. probs. c_{ij} by Eq. 6
  $\forall j \in \boldsymbol{\Omega}$: Compute capsules Ω_j by Eq. 4
  $\forall i \in \mathbf{h}, j \in \boldsymbol{\Omega}$: Update routing logits b_{ij} by Eq. 7
end for
$[\boldsymbol{\Omega}^P; \boldsymbol{\Omega}^F; \boldsymbol{\Omega}^R] = \boldsymbol{\Omega}$ ▷ *Return past, future, and redundant capsules*
return $\boldsymbol{\Omega}^P, \boldsymbol{\Omega}^F, \boldsymbol{\Omega}^R$

Now with the awareness of the current decoding status, the hidden state (input capsule) of a source word prefers to send its representation to the output capsules, which have large routing agreements associated with the input capsule. After a few rounds of iterations, the output capsules are able to ignore all but the most relevant information from the source hidden states, representing a distinct aspect of either PAST or FUTURE.

Redundant Capsules In some cases, some parts of the source sentence may belong to neither past contents nor future contents. For example, function words in English (e.g., "the") could not find its counterpart translation in Chinese. Therefore, we add additional Redundant Capsules $\boldsymbol{\Omega}^R$ (also known as "orphan capsules" in Sabour et al. (2017)), which are expected to receive higher routing assignment probabilities when a source word should not belong to either PAST or FUTURE.

We show the algorithm of our guided dynamic routing in Algorithm 1.

3.2 Integrating into NMT

The proposed GDR can be applied on the top of any sequence-to-sequence architecture, which does not require any specific modification. Let us take a Transformer-fashion architecture as example (Figure 3). Given a sentence $\mathbf{x} = \langle x_1, \ldots, x_I \rangle$, the encoder leverages N stacked identical layers to map the sentence into contextual representations:

$$\mathbf{h}^{(l)} = \text{EncoderLayer}(\mathbf{h}^{(l-1)}),$$

where the superscript l indicates layer depth. Based on the encoded source representations $\mathbf{h}^N$, a decoder generates translation word by word. The

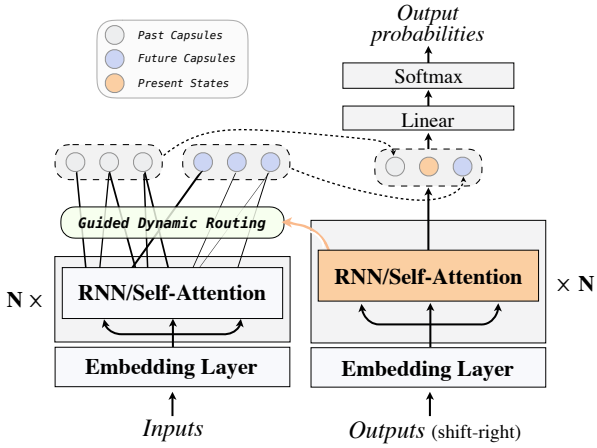


Figure 3: Illustration of our architecture.

decoder also has N stacked identical layers:

$$\mathbf{z}^{(l)} = \text{DecoderLayer}(\mathbf{z}^{(l-1)}, \mathbf{a}^{(l)}),$$
$$\mathbf{a}^{(l)} = \text{Attention}(\mathbf{z}^{(l-1)}, \mathbf{h}^{(N)}),$$

where $\mathbf{a}^{(l)}$ is the lexical-level source context assigned by an attention mechanism between current decoder layer and the last encoder layer. Given the hidden states of the last decoder layer $\mathbf{z}^{(N)}$, we perform our proposed guided dynamic routing (GDR) mechanism to compute the PAST and FUTURE contents from the source side and obtain the holistic context of each decoding step:

$$\boldsymbol{\Omega}^P, \boldsymbol{\Omega}^F, \boldsymbol{\Omega}^R = \text{GDR}(\mathbf{z}^{(N)}, \mathbf{h}^{(N)}),$$
$$\mathbf{o} = \text{FeedForward}(\mathbf{z}^{(N)}, \boldsymbol{\Omega}^P, \boldsymbol{\Omega}^F) + \mathbf{z}^{(N)},$$

where $\mathbf{o} = \langle \boldsymbol{o}_1, \cdots, \boldsymbol{o}_T \rangle$ is the sequence of the holistic context of each decoding step. Based on the holistic context, the output probabilities are computed as:

$$p(y_t | y_{<t}, \mathbf{x}) = \text{softmax}(g(\boldsymbol{o}_t)).$$

The NMT model is now able to employ the dynamic holistic context for better generation.

3.3 Learning PAST and FUTURE as Expected Auxiliary Guided Losses

To ensure that the dynamic routing process runs as expected, we introduce the following auxiliary guided signals to assist the learning process.

Bag-of-Word Constraint Weng et al. (2017) propose a multitasking scheme to boost NMT by predicting the bag-of-words of target sentence using the Word Predictions approach. Inspired by

this work, we introduce a BOW constraint to encourage the PAST and FUTURE capsules to be predictive of the preceding and subsequent bag-of-words regarding each decoding step respectively:

$$\mathcal{L}_{\text{BOW}} = \frac{1}{T}\sum_{t=0}^{T}\big(-\log p_{\text{PRE}}(y_{\leq t}|\mathbf{\Omega}_t^P) - \log p_{\text{SUB}}(y_{\geq t}|\mathbf{\Omega}_t^F)\big),$$

where $p_{pre}(y_{\leq t}|\mathbf{\Omega}_t^P)$ and $p_{sub}(y_{\geq t}|\mathbf{\Omega}_t^F)$ are the predicted probabilities of the preceding bag-of-words and subsequent words at decoding step t, respectively. For instance, the probabilities of the preceding bag-of-words are computed by:

$$p_{\text{PRE}}(y_{<t}|\mathbf{\Omega}_t^P) = \prod_{\tau\in[1,t]} p_{\text{PRE}}(y_\tau|\mathbf{\Omega}_t^P) \propto \prod_{\tau\in[1,t]} \exp(\boldsymbol{E}(y_\tau)^\top \boldsymbol{W}_{\text{BOW}}^P \mathbf{\Omega}_t^P).$$

The computation of $p_{\text{SUB}}(y_{\geq t}|\mathbf{\Omega}_t^F)$ is similar. By applying the BOW constraint, the PAST and FUTURE capsules can learn to reflect the target-side past and future bag-of-words information.

Bilingual Content Agreement Intuitively, the translated source contents should be semantically equivalent to the translated target contents, and so do untranslated contents. Thus, a natural idea is to encourage the source PAST contents, modeled by the PAST capsule to be close to the target PAST representation at each decoding step, and the same for the FUTURE. Hence, we introduce a Bilingual Content Agreement (BCA) to require the bilingual semantic-equivalent contents to be predictive to each other by Minimum Square Estimation (MSE) loss:

$$\mathcal{L}_{\text{BCA}} = \frac{1}{T}\sum_{t=1}^{T}\|\mathbf{\Omega}_t^P - \boldsymbol{W}_{\text{BCA}}^P(\frac{1}{t}\sum_{\tau=1}^{t}\boldsymbol{z}_\tau)\|^2 + \|\mathbf{\Omega}_t^F - \boldsymbol{W}_{\text{BCA}}^F(\frac{1}{T-t+1}\sum_{\tau=t}^{T}\boldsymbol{z}_\tau)\|^2,$$

where the target-side past information is represented by the averaged results of the decoder hidden states of all preceding words, while the average of subsequent decoder hidden states represents the target-side future information.

Training

Given the dataset of parallel training examples $\{\langle \mathbf{x}^{(m)}, \mathbf{y}^{(m)}\rangle\}_{m=1}^{M}$, the model parameters are trained by minimizing the loss $\mathcal{L}(\boldsymbol{\theta})$, where $\boldsymbol{\theta}$ is the set of all the parameter of the proposed model:

$$\mathcal{L}(\boldsymbol{\theta}) = \frac{1}{M}\sum_{m=1}^{M}\big(-\log p(\mathbf{y}^{(m)}|\mathbf{x}^{(m)}) + \lambda_1\cdot\mathcal{L}_{\text{BOW}} + \lambda_2\cdot\mathcal{L}_{\text{BCA}}\big),$$

where λ_1 and λ_2 are hyper-parameters.

4 Experiment

We mainly evaluated our approaches on the widely used NIST Chinese-to-English (Zh-En) translation task. We also conducted translation experiments on WMT14 English-to-German (En-De) and WMT16 English-to-Romanian (En-Ro):

1. NIST Zh-En. The training data consists of 1.09 million sentence pairs extracted from LDC[3]. We used NIST MT03 as the development set (Dev); MT04, MT05, MT06 as the test sets.

2. WMT14 En-De. The training data consists of 4.5 million sentence pairs from WMT14 news translation task. We used newstest2013 as the development set and newstest2014 as the test set.

3. WMT16 En-Ro. The training data consists of 0.6 million sentence pairs from WMT16 news translation task. We used newstest2015 as the development set and newstest2016 as the test set.

We used `transformer_base` configuration (Vaswani et al., 2017) for all the models. We run the dynamic routing for r=3 iterations. The dimension d_c of a single capsule is 256. Either PAST or FUTURE content was represented by $\frac{J}{2} = 2$ capsules. Our proposed models were trained on the top of pre-trained baseline models[4]. λ_1 and λ_2 in training objective were set to 1. In Appendix, we provide details for the training settings.

4.1 NIST Zh-En Translation

We list the results of our experiments on NIST Zh-En task in Table 1 concerning two different architectures, i.e., Transformer and RNMT. As we can see, all of our models substantially outperform the baselines in terms of averaged BLEU score of all the test sets. Among them, our best model achieves 45.65 BLEU based on Transformer architecture. We also find that redundant capsules are helpful while discarding them leads to -0.35 BLEU degradation (45.65 vs 45.30).

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

[4]Pre-training is only for efficiency purpose. Our approach could also learn from scratch.

Model	$\|\boldsymbol{\theta}\|$	v_{train}	v_{test}	Dev	MT04	MT05	MT06	Tests Avg.
Transformer	66.1m	1.00×	1.00×	45.83	46.66	43.36	42.17	44.06
GDR	68.9m	0.77×	0.94×	46.50	47.03	45.50	42.21	44.91 (+0.75)
+ $\mathcal{L}_{\text{BOW}}$	69.2m	0.70×	0.94×	47.12	48.09	45.98	42.68	45.58 (+1.42)
+ $\mathcal{L}_{\text{BCA}}$	69.4m	0.75×	0.94×	46.86	48.00	45.67	42.62	45.43 (+1.37)
+ $\mathcal{L}_{\text{BOW}}$ + $\mathcal{L}_{\text{BCA}}$ [OURS]	69.7m	0.67×	0.94×	**47.52**	**48.13**	**45.98**	**42.85**	**45.65 (+1.59)**
OURS - *redundant capsules*	68.7m	0.69×	0.94×	47.20	47.82	45.59	42.51	45.30 (+1.24)
RNMT	50.2m	1.00×	1.00×	35.98	37.85	36.12	35.86	36.61
+PFRNN (Zheng et al., 2018)	N/A	0.54×	0.74×	37.90	40.37	36.75	36.44	37.85 (+1.24)
+AOL (Kong et al., 2019)	N/A	0.57×	1.00×	37.61	40.05	37.58	36.87	38.16 (+1.55)
OURS	53.9m	0.62×	0.90×	**38.10**	**40.87**	**37.50**	**37.00**	**38.45 (+1.84)**

Table 1: Experiment ressuts on NIST Zh-En task, including number of parameters ($|\boldsymbol{\theta}|$, excluding word embeddings), training/testing speeds (v_{train}/v_{test}), and translation results in case-insensitive BLEU.

Architectures Our approach shows consistent effects on both Transformer and RNMT architectures. In comparison to the Transformer baseline, our model achieves at most +1.59 BLEU improvement (45.65 v.s 44.06), while +1.84 BLEU improvement over RNMT baselines (38.45 v.s 36.61). These results indicate the compatibility of our approach to different architectures.

Auxiliary Guided Losses Both the auxiliary guided losses help our model for better learning. The BOW constraint leads to a +0.67 improvement compared to the vanilla GDR, while the benefit is +0.62 for BCA. Combination of both gains the most margins (+0.84), which means that they can supplement each other.

Efficiency To examine the efficiency of the proposed approach, we also list the relative speed of both training and testing. Our approach is 0.67× slower than the Transformer baseline in training phase, however, it does not hurt the speed of testing too much (0.94×). It is because the most extra computation in training phrase is related to the softmax operations of BOW losses, the degradation of the testing efficiency is moderate.

Comparison to Other Work On the experiments on RNMT architecture, we list two related works. Zheng et al. (2018) use extra PAST and FUTURE RNNs to capture translated and untranslated contents recurrently (PFRNN), while Kong et al. (2019) directly leverage translation adequacy as learning reward by their proposed Adequacy-oriented Learning (AOL). Compared to them, our model also enjoys competitive improvements due to the explicit separation of source contents. In addition, PFRNN is non-trivial to adapt to Transformer, because it requires a recurrent process which fails to be compatible with parallel training of Transformer, scarifying Transformer's efficiency advantage.

Model	En-De	En-Ro
GNMT+RL (Wu et al., 2016)	24.6	N/A
ConvS2S (Gehring et al., 2017)	25.2	29.88
Transformer (Vaswani et al., 2017)	27.3	N/A
+AOL (Kong et al., 2019)	28.01	N/A
Transformer (Gu et al., 2017)	N/A	31.91
Transformer	27.14	32.10
OURS	**28.10**	**32.96**

Table 2: Case-sensitive BLEU on WMT14 En-De and WMT16 En-Ro tasks.

4.2 WMT En-De and En-Ro Translation

We evaluated our approach on WMT14 En-De and WMT16 En-Ro tasks. As shown in Table 2, our reproduced Transformer baseline systems are close to the state-of-the-art results in previous work, which guarantee the comparability of our experiments. The results show a consistent trend of improvements as NIST Zh-En task on WMT14 En-De (+0.96 BLEU) and WMT16 En-Ro (+0.86 BLEU) benchmarks. We also list the results of other published research for comparison, where our model outperforms the previous results in both language pairs. Note that our approach also surpasses Kong et al. (2019) on WMT14 En-De task. These experiments demonstrate the effectiveness of our approach across different language pairs.

4.3 Analysis and Discussion

***Our model learns* PAST *and* FUTURE.** We visualize the assignment probabilities in the last routing iteration (Figure 4). Interestingly, there is a clear trend that the assignment probabilities to the PAST capsules gradually raise up, while

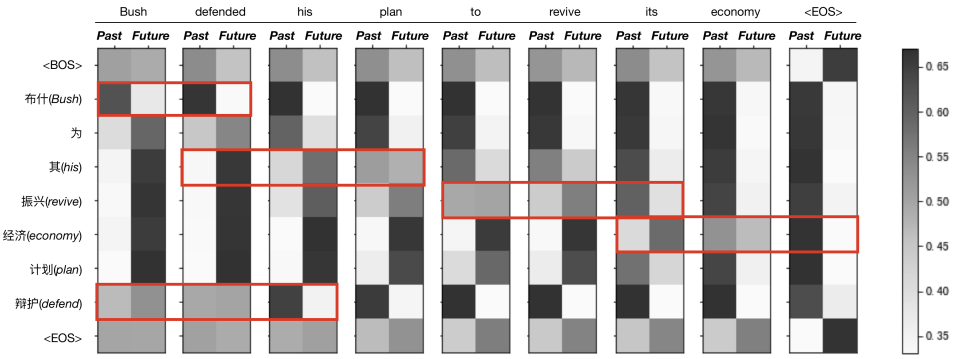


Figure 4: Visualization of the assignment probabilities of iterative routing. Each sub-heatmap is associated with a target word, where the left column is the probabilities of each source words routing to the PAST capsules, and the right one is to the FUTUREExamples in the red frame indicate the changes before and after the generation of the central word. We omit the assignment probabilities associated with the redundant capsules for simplicity. For instance, after the target word "defended" was generated, the assignment probabilities of its source translation "辩护" changed from FUTURE to PAST. Results of "Bush", "his", "revive" and "economy" are similar, except a adverse case ("plan").

those to the FUTURE capsules reduce to around zeros. This phenomenon is consistent with the intuition that the translated contents should aggregate and the untranslated should decline (Zheng et al., 2018). The assignment weights of a specific word change from FUTURE to PAST after being generated. These pieces of evidence give a strong verification that our GDR mechanism indeed has learned to distinguish the PAST contents and FUTURE contents in the source-side.

Moreover, we measure how well our capsules accumulate the expected contents by comparison between the BOW predictions and ground-truth target words. Accordingly, we define a *top-5 overlap rate* (r_{OL}) for predicting preceding and subsequent words are defined as follow, respectively: $r^P_{\text{OL}} = \frac{1}{T}\sum_{t=1}^{T}\frac{|\text{Top}_{5\text{t}}(p_{pre}(\mathbf{\Omega}^P_t)) \cap y_{<=t}|}{|y_{<=t}|}$, $r^F_{\text{OL}} = \frac{1}{T}\sum_{t=1}^{T}\frac{|\text{Top}_{5(\text{T}-\text{t})}(p_{sub}(\mathbf{\Omega}^F_t)) \cap y_{>=t}|}{|y_{>=t}|}$. The PAST capsules achieves r^P_{OL} of 0.72, while r^F_{OL} of 0.70 for the FUTURE capsules. The results indicate that the capsules could predict the corresponding words to a certain extent, which implies the capsules contain the expected information of PAST or FUTURE contents.

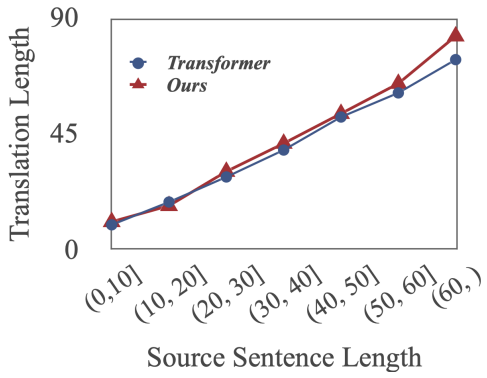
Translations become better and more adequate. To validate the translation adequacy of our model, we use Coverage Difference Ratio (CDR) proposed by Kong et al. (2019), i.e., $\text{CDR} = 1 - \frac{|C_{\text{ref}} \backslash C_{\text{gen}}|}{|C_{\text{ref}}|}$, where C_{ref} and C_{gen} are the set of source words covered by the reference and translation, respectively. The CDR reflects the translation adequacy by comparing the source coverages between reference and translation. As shown in Table 3, our approach achieves a better CDR than the Transformer baseline, which means superiority in translation adequacy.

Model	Transformer	OURS
CDR	0.73	**0.79**
HUMAN EVALUATION		
QUALITY	4.39±.11	**4.66±.10**
OVER(%)	0.03±.01	**0.01±.01**
UNDER(%)	3.83±.97	**2.41±.80**

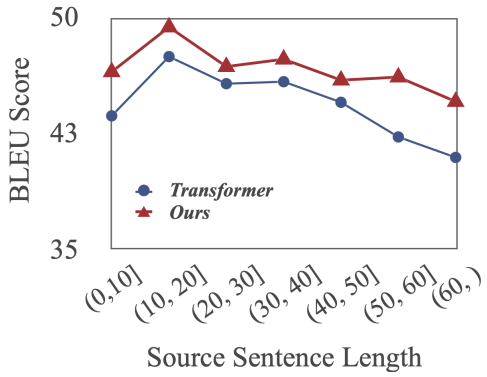
Table 3: Evaluation on translation quality and adequacy. For HUMAN evaluation, we asked three evaluators to score translations from 100 source sentences, which are randomly sampled from the testsets from anonymous systems, the QUALITY from 1 to 5 (higher is better), and the proportions of source words concerning OVER- and UNDER-translation, respectively.

Following Zheng et al. (2018), we also conduct subjective evaluations to validate the benefit of modeling PAST and FUTURE (the last three rows of Table 3). Surprisingly, we find that the modern NMT model, i.e., Transformer, rarely produces over-translation but still suffers from under-translation. Our model obtains the highest human rating on translation quality while substantially alleviates the under-translation problem than Transformer.

Longer sentences benefit much more. We report the comparison with sentence lengths (Figure 5). In all the intervals of length, our model does generate better (Figure 5b) and longer (Figure 5a) translations. Interestingly, our approach



(a) Translation length v.s source length



(b) BLEU v.s source length

Figure 5: Comparison regarding source length.

gets a larger improvement when the input sentences become longer, which are commonly thought hard to translate. We attribute this to the less number of under-translation cases in our model, meaning that our model learns better translation quality and adequacy, especially for long sentences.

Does guided dynamic routing really matter? Despite the promising numbers of the GDR and the auxiliary guided losses, a straightforward question rises: will other more simple models also work if they are just equipped with the guided losses to recognize PAST and FUTURE contents? In other word, does the proposed guided dynamic routing really matter?

To answer this question, we integrate the proposed auxiliary losses into two simple baselines to guide the recognition of past and future: A MLP classifier model (CLF) that determines if a source word is a past word, otherwise future[5]; and an attention-based model (ATTN) that uses two individual attention modules to retrieve past or future parts from the source words. As shown in Table 6, surprisingly, the simple baselines can obtain improvements, emphasizing the function of the proposed guided losses, while there remain a considerable gaps between our model and them. In fact, the CLF is essentially a *one-iteration* variant of GDR, and iterative refinement by multiple iterations is necessary and effective[6]. And the attention mechanism is used for feature pooling, not suitable for parts-to-wholes assignment[7]. These experiments reveal that our guided dynamic routing is a better choice to model and exploit the dynamic PAST and FUTURE.

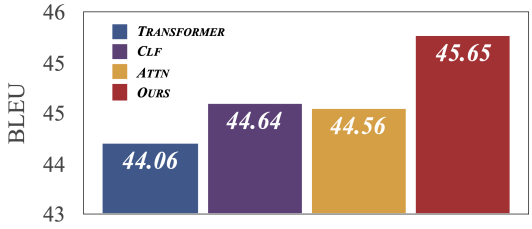


Figure 6: Comparison with simple baselines with the same auxiliary guided loss on NIST Zh-En.

5 Related Work

Inadequate translation problem is a widely known weakness of NMT models, especially when translating long sentences (Kong et al., 2019; Tu et al., 2016; Lei et al., 2019). To alleviate this problem, one direction is to recognize the translated and untranslated contents, and pay more attention to untranslated parts. Tu et al. (2016), Mi et al. (2016) and Li et al. (2018) employ coverage vector or coverage ratio to indicate the lexical-level coverage of source words. Meng et al. (2018) influence the attentive vectors by translated/untranslated information. Our work mainly follows the path of Zheng et al. (2018), which introduce two extra recurrent layers in the decoder to maintain the representations of the past and future translation contents. However, it may be not easy to show the direct correspondence between the source contents and learned representations in the past/future

[5]CLF is a 3-way classifier that computes the probabilities $p^P(x_i)$, $p^F(x_i)$ and $p^R(x_i)$ (they sum to 1) as past, future and redundant weights, which is similar to Equation 6. The PAST and FUTURE representations are computed by weighted summation, which is similar to Equation 4.

[6]See Appendix for analysis of iteration numbers.

[7]Consider an extreme case that in the end of translation, there is no FUTURE content left, but the attention model still produces a *weighted average* over all the source representations, which is nonsense. In contrast, the GDR is able to assign zero probabilities to the FUTURE capsules, solving the source of the problem.

RNN layers, nor compatible with the state-of-the-art Transformer for the additional recurrences prevent Transformer decoder from being parallelized.

Another direction is to introduce global representations. Lin et al. (2018) model a global source representation by deconvolution networks. Xia et al. (2017); Zhang et al. (2018); Geng et al. (2018) propose to provide a holistic view of target sentence by multi-pass decoding. Zhou et al. (2019) improve Zhang et al. (2018) to a synchronous bidirectional decoding fashion. Similarly, Weng et al. (2019) deploy bidirectional decoding in interactive translation setting. Different from these work aiming at providing static global information in the whole translation process, our approach models a dynamically global (holistic) context by using capsules network to separate source contents at every decoding steps.

Other efforts explore exploiting future hints. Serdyuk et al. (2018) design a Twin Regularization to encourage the hidden states in forward decoder RNN to estimate the representations of a backward RNN. Weng et al. (2017) require the decoder states to not only generate the current word but also predict the remain untranslated words. Actor-critic algorithms are employed to predict future properties (Li et al., 2017; Bahdanau et al., 2017; He et al., 2017) by estimating the future rewards for decision making. Kong et al. (2019) propose a policy gradient based adequacy-oriented approach to improve translation adequacy. These methods use future information only at the training stage, while our model could also exploit past and future information at inference, which provides accessible clues of translated and untranslated contents.

Capsule networks (Hinton et al., 2011) and its associated assignment policy of dynamic routing (Hinton et al., 2011) and EM-routing (Hinton et al., 2018) aims at addressing the limited expressive ability of the parts-to-wholes assignment in computer vision. In natural language processing community, however, the capsule network has not yet been widely investigated. Zhao et al. (2018) testify capsule network on text classification and Gong et al. (2018) propose to aggregate a sequence of vectors via dynamic routing for sequence encoding. Dou et al. (2019) first propose to employ capsule network in NMT using routing-by-agreement mechanism for layer representation aggregation. Wang (2019) develops a constant time NMT model using capsule networks. These studies mainly use capsule network for information aggregation, where the capsules could have a less interpretable meaning. In contrast, our model learns what we expect by the aid of auxiliary learning signals, which endows our model with better interpretability.

6 Conclusion

In this paper, we propose to recognize the translated PAST and untranslated FUTURE contents via parts-to-wholes assignment in neural machine translation. We propose the guided dynamic routing, a novel mechanism that explicitly separates source words into PAST and FUTURE guided by PRESENT target decoding status at each decoding step. We empirically demonstrate that such explicit separation of source contents benefit neural machine translation with considerable and consistent improvements on three language pairs. Extensive analysis shows that our approach learns to model the PAST and FUTURE as expected, and alleviates the inadequate translation problem. It is interesting to apply our approach to other sequence-to-sequence tasks, e.g., text summarization (as listed in Appendix).

Acknowledgement

We would like to thank the anonymous reviewers for their insightful comments. Shujian Huang is the corresponding author. This work is supported by the National Science Foundation of China (No. U1836221 and No. 61772261), the Jiangsu Provincial Research Foundation for Basic Research (No. BK20170074).

References

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*.

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

Ziyi Dou, Zhaopeng Tu, Xing Wang, Longyue Wang, Shuming Shi, and Tong Zhang. 2019. Dynamic layer aggregation for neural machine translation. In *AAAI*.

Jonas Gehring, Michael Auli, David Grangier, Denis Yarats, and Yann Dauphin. 2017. Convolutional sequence to sequence learning. In *ICML 2017*.

Xinwei Geng, Xiaocheng Feng, Bing Qin, and Ting Liu. 2018. Adaptive multi-pass decoder for neural machine translation. In *EMNLP*, pages 523–532.

Jingjing Gong, Xipeng Qiu, Shaojing Wang, and Xuanjing Huang. 2018. Information aggregation via dynamic routing for sequence encoding. In *COLING*, pages 2742–2752.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O. K. Li, and Richard Socher. 2017. Non-Autoregressive Neural Machine Translation.

Di He, Hanqing Lu, Yingce Xia, Tao Qin, Liwei Wang, and Tie-Yan Liu. 2017. Decoding with value networks for neural machine translation. In *NIPS*, pages 178–187.

Geoffrey E Hinton, Alex Krizhevsky, and Sida D Wang. 2011. Transforming auto-encoders. In *ICANN*, pages 44–51. Springer.

Geoffrey E Hinton, Sara Sabour, and Nicholas Frosst. 2018. Matrix capsules with EM routing. In *ICLR*.

Xiang Kong, Zhaopeng Tu, Shuming Shi, Eduard Hovy, and Tong Zhang. 2019. Neural machine translation with adequacy-oriented learning. In *AAAI*.

Wenqiang Lei, Weiwen Xu, Ai Ti Aw, Yuanxin Xiang, and Tat-Seng Chua. 2019. Revisit automatic error detection for wrong and missing translation – a supervised approach. In *EMNLP*.

Jian Li, Baosong Yang, Zi-Yi Dou, Xing Wang, Michael R Lyu, and Zhaopeng Tu. 2019. Information aggregation for multi-head attention with routing-by-agreement. In *NAACL-HLT*, pages 3566–3575.

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

Yanyang Li, Tong Xiao, Yinqiao Li, Qiang Wang, Changming Xu, and Jingbo Zhu. 2018. A simple and effective approach to coverage-aware neural machine translation. In *ACL*, volume 2, pages 292–297.

Junyang Lin, Xu Sun, Xuancheng Ren, Shuming Ma, Jinsong Su, and Qi Su. 2018. Deconvolution-based global decoding for neural machine translation. In *COLING*, pages 3260–3271.

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

Fandong Meng, Zhaopeng Tu, Yong Cheng, Haiyang Wu, Junjie Zhai, Yuekui Yang, and Di Wang. 2018. Neural machine translation with key-value memory-augmented attention. In *IJCAI*, pages 2574–2580. AAAI Press.

Haitao Mi, Baskaran Sankaran, Zhiguo Wang, and Abe Ittycheriah. 2016. Coverage Embedding Models for Neural Machine Translation. In *EMNLP*.

Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. 2017. Dynamic routing between capsules. In *NIPS*, pages 3856–3866.

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

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

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

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

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 *NIPS*.

Mingxuan Wang. 2019. Towards linear time neural machine translation with capsule networks. In *EMNLP*.

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

Rongxiang Weng, Hao Zhou, Shujian Huang, Lei Li, Yifan Xia, and Jiajun Chen. 2019. Correct-and-memorize: Learning to translate from interactive revisions. In *IJCAI*.

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 *NIPS*, pages 1784–1794.

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

Wei Zhao, Jianbo Ye, Min Yang, Zeyang Lei, Suofei Zhang, and Zhou Zhao. 2018. Investigating capsule networks with dynamic routing for text classification. In *AAAI*.

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

Long Zhou, Jiajun Zhang, and Chengqing Zong. 2019. Synchronous bidirectional neural machine translation. *TACL*, 7:91–105.