Translating Phrases in Neural Machine Translation

Xing Wang[†] **Zhaopeng Tu**[‡] **Deyi Xiong**[†*] **Min Zhang**[†]
[†]Soochow University, Suzhou, China
xingwsuda@gmail.com, {dyxiong, minzhang}@suda.edu.cn
[‡] Tencent AI Lab, Shenzhen, China
tuzhaopeng@gmail.com

Abstract

Phrases play an important role in natural language understanding and machine translation (Sag et al., 2002; Villavicencio et al., 2005). However, it is difficult to integrate them into current neural machine translation (NMT) which reads and generates sentences word by word. In this work, we propose a method to translate phrases in NMT by integrating a phrase memory storing target phrases from a phrase-based statistical machine translation (SMT) system into the encoder-decoder architecture of NMT. At each decoding step, the phrase memory is first re-written by the SMT model, which dynamically generates relevant target phrases with contextual information provided by the NMT model. Then the proposed model reads the phrase memory to make probability estimations for all phrases in the phrase memory. If phrase generation is carried on, the NMT decoder selects an appropriate phrase from the memory to perform phrase translation and updates its decoding state by consuming the words in the selected phrase. Otherwise, the NMT decoder generates a word from the vocabulary as the general NMT decoder does. Experiment results on the Chinese→English translation show that the proposed model achieves significant improvements over the baseline on various test sets.

1 Introduction

Neural machine translation (NMT) has been receiving increasing attention due to its impressive translation performance (Kalchbrenner and Blunsom, 2013; Cho et al., 2014; Sutskever et al., 2014; Bahdanau et al., 2015; Wu et al., 2016). Significantly different from conventional statistical machine translation (SMT) (Brown et al., 1993; Koehn et al., 2003; Chiang, 2005), NMT adopts a big neural network to perform the entire translation process in one shot, for which an encoder-decoder architecture is widely used. Specifically, the encoder encodes a source sentence into a continuous vector representation, then the decoder uses the continuous vector representation to generate the corresponding target translation word by word.

The word-by-word generation philosophy in NMT makes it difficult to translate multi-word phrases. Phrases, especially multi-word expressions, are crucial for natural language understanding and machine translation (Sag et al., 2002; Villavicencio et al., 2005) as the meaning of a phrase cannot be always deducible from the meanings of its individual words or parts. Unfortunately current NMT is essentially a word-based or character-based (Chung et al., 2016; Costa-jussà and Fonollosa, 2016; Luong and Manning, 2016) translation system where phrases are not considered as translation units. In contrast, phrases are much better than words as translation units in SMT and have made a significant advance in translation quality. Therefore, a natural question arises: Can we translate phrases in NMT?

Recently, there have been some attempts on multi-word phrase generation in NMT (Stahlberg et al., 2016b; Zhang and Zong, 2016). However these efforts constrain NMT to generate either syntactic phrases or domain phrases in the word-by-word generation framework. To explore the phrase generation in NMT beyond the word-by-word generation framework, we propose a novel architecture that integrates a phrase-based SMT

*Corresponding author

Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, pages 1421–1431
Copenhagen, Denmark, September 7–11, 2017.

model into NMT. Specifically, we add an auxiliary phrase memory to store target phrases in symbolic form. At each decoding step, guided by the decoding information from the NMT decoder, the SMT model dynamically generates relevant target phrase translations and writes them to the memory. Then the NMT decoder scores phrases in the phrase memory and selects a proper phrase or word with the highest probability. If the phrase generation is carried out, the NMT decoder generates a multi-word phrase and updates its decoding state by consuming the words in the selected phrase.

Furthermore, in order to enhance the ability of the NMT decoder to effectively select appropriate target phrases, we modify the encoder of NMT to make it fit for exploring structural information of source sentences. Particularly, we integrate syntactic chunk information into the NMT encoder, to enrich the source-side representation. We validate our proposed model on the Chinese→English translation task. Experiment results show that the proposed model significantly outperforms the conventional attention-based NMT by 1.07 BLEU points on multiple NIST test sets.

The rest of this paper is organized as follows. Section 2 briefly introduces the attention-based NMT as background knowledge. Section 3 presents our proposed model which incorporates the phrase memory into the NMT encoder-decoder architecture, as well as the reading and writing procedures of the phrase memory. Section 4 presents our experiments on the Chinese→English translation task and reports the experiment results. Finally we discuss related work in Section 5 and conclude the paper in Section 6.

2 Background

Neural machine translation often adopts the encoder-decoder architecture with recurrent neural networks (RNN) to model the translation process. The bidirectional RNN encoder which consists of a forward RNN and a backward RNN reads a source sentence $\mathbf{x} = x_1, x_2, ..., x_{T_x}$ and transforms it into word annotations of the entire source sentence $\mathbf{h} = h_1, h_2, ..., h_{T_x}$. The decoder uses the annotations to emit a target sentence $\mathbf{y} = y_1, y_2, ..., y_{T_y}$ in a word-by-word manner.

In the training phase, given a parallel sentence $(\mathbf{x}, \mathbf{y})$, NMT models the conditional probability as follows,

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{T_y} P(y_i|\mathbf{y}_{<\mathbf{i}}, \mathbf{x}) \quad (1)$$

where y_i is the target word emitted by the decoder at step i and $\mathbf{y}_{<\mathbf{i}} = y_1, y_2, ..., y_{i-1}$. The conditional probability $P(y_i|\mathbf{y}_{<\mathbf{i}}, \mathbf{x})$ is computed as

$$P(y_i|\mathbf{y}_{<\mathbf{i}}, \mathbf{x}) = softmax(f(s_i, y_{i-1}, c_i)) \quad (2)$$

where $f(\cdot)$ is a non-linear function and s_i is the hidden state of the decoder at step i:

$$s_i = g(s_{i-1}, y_{i-1}, c_i) \quad (3)$$

where $g(\cdot)$ is a non-linear function. Here we adopt Gated Recurrent Unit (Cho et al., 2014) as the recurrent unit for the encoder and decoder. c_i is the context vector, computed as a weighted sum of the annotations $\mathbf{h}$:

$$c_i = \sum_{j=1}^{T_x} \alpha_{t,j} h_j \quad (4)$$

where h_j is the annotation of source word x_j and its weight $\alpha_{t,j}$ is computed by the attention model.

We train the attention-based NMT model by maximizing the log-likelihood:

$$C(\theta) = \sum_{n=1}^{N} \sum_{i=1}^{T_y} \log P(y_i^n|\mathbf{y}_{<i}^n, \mathbf{x}^n) \quad (5)$$

given the training data with N bilingual sentences (Cho, 2015).

In the testing phase, given a source sentence $\mathbf{x}$, we use beam search strategy to search a target sentence $\hat{\mathbf{y}}$ that approximately maximizes the conditional probability $P(\mathbf{y}|\mathbf{x})$

$$\hat{\mathbf{y}} = \underset{\mathbf{y}}{\operatorname{argmax}}\, P(\mathbf{y}|\mathbf{x}) \quad (6)$$

3 Approach

In this section, we introduce the proposed model which incorporates a phrase memory into the encoder-decoder architecture of NMT. Inspired by the recent work on attaching an external structure to the encoder-decoder architecture (Gulcehre et al., 2016; Gu et al., 2016; Tang et al., 2016; Wang et al., 2017), we adopt a similar approach to incorporate the phrase memory into NMT.

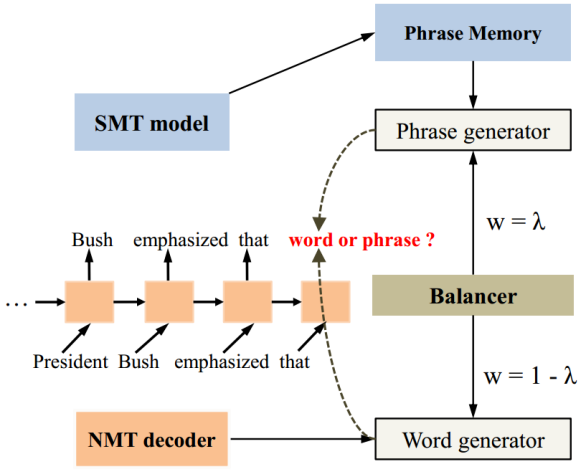

Figure 1: Architecture of the NMT decoder with the phrase memory. The NMT decoder performs phrase generation using the balancer and the phrase memory.

3.1 Framework

Figure 1 shows an example. Given the generated words "*President Bush emphasized that*", the model generates the next fragment either from a word generation mode or a phrase generation mode. If the model selects the word generation mode, it generates a word by the NMT decoder as in the standard NMT framework. Otherwise, it generates a multi-word phrase by enquiring a phrase memory, which is written by an SMT decoder based on the dynamic decoding information from the NMT model for each step. The trade-off between word generation mode and phrase generation mode is balanced by a weight λ, which is produced by a neural network based *balancer*.

Formally, a generated translation $\mathbf{y} = \{y_1, y_2, \ldots, y_{T_y}\}$ consists of two sets of fragments: words generated by NMT decoder $\mathbf{w} = \{w_1, w_2, \ldots, w_K\}$ and phrases generated from the phrase memory $\mathbf{p} = \{p_1, p_2, \ldots, p_L\}$. The probability of generating $\mathbf{y}$ is calculated by

$$P(\mathbf{y}|\mathbf{x}) = \prod_{w_k \in \mathbf{w}} (1 - \lambda_{t(w_k)}) P_{word}(w_k) \times \prod_{p_l \in \mathbf{p}} \lambda_{t(p_l)} P_{phrase}(p_l) \quad (7)$$

where $P_{word}(w_k)$ is the probability of generating the word w_k (see Equation 2), $P_{phrase}(p_l)$ is that of generating the phrase p_l which will be described in Section 3.2, and $t(\cdot)$ is the decoding step to generate the corresponding fragment.

The balancing weight λ is produced by the *balancer* – a multi-layer network. The balancer network takes as input the decoding information, including the context vector c_i, the previous decoding state s_{i-1} and the previous generated word y_{i-1}:

$$\lambda_i = \sigma(f_b(s_i, y_{i-1}, c_i)) \quad (8)$$

where $\sigma(\cdot)$ is a sigmoid function and $f_b(\cdot)$ is the activation function. Intuitively, the weight λ can be treated as the estimated importance of the phrase to be generated. We expect λ to be high if the phrase is appropriate at the current decoding step.

Well-Formed Phrases We employ a source-side chunker to chunk the source sentence, and only phrases that corresponds to a source chunk are used in our model. We restrict ourselves to the well-formed chunk phrases based on the following considerations: (1) In order to take advantage of dynamic programming, we restrict ourselves to non-overlap phrases.[1] (2) We explicitly utilize the boundary information of the source-side chunk phrases, to better guide the proposed model to adopt a target phrase at an appropriate decoding step. (3) We enable the model to exploit the syntactic categories of chunk phrases to enhance the proposed model with its selection preference for special target phrases. With these information, we enrich the context vector c_i to enable the proposed model to make better decisions, as described below.

Following the commonly-used strategy in sequence tagging tasks (Xue and Shen, 2003), we allow the words in a phrase to share the same chunk tag and introduce a special tag for the beginning word. For example, the phrase " 信息 安全 (information security)" is tagged as a noun phrase "NP", and the tag sequence should be "NP_B NP". Partially motivated by the work on integrating linguistic features into NMT (Sennrich and Haddow, 2016), we represent the encoder input as the combination of word embeddings and chunking tag embeddings, instead of word embeddings alone in the conventional NMT. The new input is formulated as follows:

$$[E^w x_i, E^t t_i] \quad (9)$$

[1] Overlapped phrases may result in a high dimensionality in translation hypothesis representation and make it hard to employ shared fragments for efficient dynamic programming.

where $E^w \in \mathbb{R}^{dw \times |V^{NMT}|}$ is a word embedding matrix and dw is the word embedding dimensionality, $E^t \in \mathbb{R}^{dt \times |V^{TAG}|}$ is a tag embedding matrix and dt is the tag embedding dimensionality. $[\cdot]$ is the vector concatenation operation.

3.2 Phrase Memory

The phrase memory stores relevant target phrases provided by an SMT model, which is trained on the same bilingual corpora. At each decoding step, the memory is firstly erased and re-written by the SMT model, the decoding of which is based on the translation information provided by the NMT model. Then, the proposed model enquires phrases along with their probabilities P_{phrase} from the memory.

Writing to Phrase Memory Given a partial translation $\mathbf{y}_{<i} = \{y_1, y_2, \ldots, y_{t-1}\}$ generated from NMT, the SMT model picks potential phrases extracted from the translation table. The phrases are scored with multiple SMT features, including the language model score, the translation probabilities, the reordering score, and so on. Specially, the reordering score depends on alignment information between source and target words, which is derived from attention distribution produced by the NMT model (Wang et al., 2017). SMT coverage vector in (Wang et al., 2017) is also introduced to avoid repeat phrasal recommendations. In our work, the potential phrase is phrase with high SMT score which is defined as following:

$$SMT_{score}(p_l|\mathbf{y}_{<\mathbf{t}}, \mathbf{x}) = \sum_{m=1}^{M} w_m h_m(p_l, x(p_l)) \tag{10}$$

where p_l is a target phrase and $x(p_l)$ is its corresponding source span. $h_m(p_l, x(p_l))$ is a SMT feature function and w_m is its weight. The feature weights can be tuned by the minimum error rate training (MERT) algorithm (Och, 2003).

This leads to a better interaction between SMT and NMT models. It should be emphasized that our memory is dynamically updated at each decoding step based on the decoding history from both SMT and NMT models.

The proposed model is very flexible, where the phrase memory can be either fully dynamically generated by an SMT model or directly extracted from a bilingual dictionary, or any other bilingual resources storing idiomatic translations or bilingual multi-word expressions, which may lead to a further improvement. [2]

Reading Phrase Memory When phrases are read from the memory, they are rescored by a neural network based score function. The score function takes as input the phrase itself and decoding information from NMT ($i = t(p_l)$ denotes the current decoding step):

$$score_{phrase}(p_l) = g_s\big(e(p_l), s_i, y_{i-1}, c_i\big) \tag{11}$$

where $g_s(\cdot)$ is either an identity or a non-linear function. $e(p_l)$ is the representation of phrase p_l, which is modeled by a recurrent neural networks. Again, s_i is the decoder state, y_{i-1} is the lastly generated word, and c_i is the context vector. The scores are normalized for all phrases in the phrase memory, and the probability for phrase p_l is calculated as

$$P_{phrase}(p_l) = softmax(score_{phrase}(p_l)) \tag{12}$$

The probability calculation is controlled with parameters, which are trained together with the parameters from the NMT model.

3.3 Training

Formally, we train both the default parameters of standard NMT and the new parameters associated with phrase generation on a set of training examples $\{[\mathbf{x}^n, \mathbf{y}^n]\}_{n=1}^N$:

$$C(\theta) = \sum_{n=1}^{N} \log P(\mathbf{y}^n|\mathbf{x}^n) \tag{13}$$

where $P(\mathbf{y}^n|\mathbf{x}^n)$ is defined in Equation 7. Ideally, the trained model is expected to produce a higher balance weight λ and phrase probability P_{phrase} when a phrase is selected from the memory, and lower scores in other cases.

3.4 Decoding

During testing, the NMT decoder generates a target sentence which consists of a mixture of words and phrases. Due to the different granularities of words and phrases, we design a variant of beam search strategy: At decoding step i, we first compute P_{phrase} for all phrases in the phrase memory

[2] Bilingual resources can be utilized in two ways: First, we can store the bilingual resources in a static memory and keep all items available to NMT in the whole decoding period. Second, we can integrate the bilingual resources into SMT and then dynamically feed them into the phrase memory.

and P_{word} for all words in NMT vocabulary. Then the balancer outputs a balancing weight λ_i, which is used to scale the phrase and word probabilities : $\lambda_i \times P_{phrase}$ and $(1 - \lambda_i) \times P_{word}$. Now outputs are normalized probabilities on the concatenation of phrase memory and the general NMT vocabulary. At last, the NMT decoder generates a proper phrase or word of the highest probability.

If a target phrase in the phrase memory has the highest probability, the decoder generates the target phrase to complete the multi-word phrase generation process, and updates its decoding state by consuming the words in the selected phrase as described in Equation 3. All translation hypotheses are placed in the corresponding beams according to the number of generated target words.

4 Experiments

In this section, we evaluated the effectiveness of our model on the Chinese→English machine translation task. The training corpora consisted of about 1.25 million sentence pairs[3] with 27.9 million Chinese words and 34.5 million English words respectively. We used NIST 2006 (NIST06) dataset as development set, and NIST 2004 (NIST04), 2005 (NIST05) and 2008 (NIST08) datasets as test sets. We report experiment results with case-insensitive BLEU score[4].

We compared our proposed model with two state-of-the-art systems:

* **Moses**: a state-of-the-art phrase-based SMT system (Koehn et al., 2007) with its default settings, where feature function weights are tuned by the minimum error rate training (MERT) algorithm (Och, 2003).

* **RNNSearch**: an in-house implementation of the attention-based NMT system (Bahdanau et al., 2015) with its default settings.

For Moses, we used the full bilingual training data to train the phrase-based SMT model and the target portion of the bilingual training data to train a 4-gram language model using KenLM[5]. We ran Giza++ on the training data in both Chinese-to-English and English-to-Chinese directions and applied the "grow-diag-final" refinement rule (Koehn et al., 2003) to obtain word alignments. The maximum phrase length is set to 7.

For RNNSearch, we generally followed settings in the previous work (Bahdanau et al., 2015; Tu et al., 2017a,b). We only kept a shortlist of the most frequent 30,000 words in Chinese and English, covering approximately 97.7% and 99.3% of the data in the two languages respectively. We constrained our source and target sequences to have a maximum length of 50 words in the training data. The size of embedding layer of both sides was set to 620 and the size of hidden layer was set to 1000. We used a minibatch stochastic gradient descent (SGD) algorithm of size 80 together with Adadelta (Zeiler, 2012) to train the NMT models. The decay rates ρ and ϵ were set as 0.95 and 10^{-6}. We clipped the gradient norm to 1.0 (Pascanu et al., 2013). We also adopted the dropout technique. Dropout was applied only on the output layer and the dropout rate was set to 0.5. We used a simple beam search decoder with beam size 10 to find the most likely translation.

For the proposed model, we used a Chinese chunker[6] (Zhu et al., 2015) to chunk the source-side Chinese sentences. 13 chunking tags appeared in our chunked sentences and the size of chunking tag embedding was set to 10. We used the trained phrase-based SMT to translate the source-side chunks. The top 5 translations according to their translation scores (Equation 10) were kept and among them multi-word phrases were used as phrasal recommendations for each source chunk phrase. For a source-side chunk phrase, if there exists phrasal recommendations from SMT, the output chunk tag was used as its chunking tag feature as described in Section 3.1. Otherwise, the words in the chunk were treated as general words by being tagged with the default tag. In the phrase memory, we only keep the top 7 target translations with highest SMT scores at each decoding step. We used a forward neural network with two hidden layers for both the balancer (Equation 8) and the scoring function (Equation 11). The numbers of units in the hidden layers were set to 2000 and 500 respectively. We used a backward RNN encoder to learn the phrase representations of target phrases in the phrase memory.

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

[4]ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v11b.pl

[5]https://kheafield.com/code/kenlm/

[6]http://www.niuparser.com/

SYSTEM	NIST04	NIST05	NIST08	Avg
Moses	34.74	31.99	23.69	30.14
RNNSearch	37.80	34.70	24.93	32.48
+memory	38.21	35.15	25.48†	32.95
+memory +chunking tag	38.83‡	35.72‡	26.09‡	33.55

Table 1: Main experiment results on the NIST Chinese-English translation task. BLEU scores in the table are case insensitive. Moses and RNNSearch are SMT and NMT baseline system respectively. "†": significantly better than RNNSearch ($p < 0.05$); "‡": significantly better than RNNSearch ($p < 0.01$).

	NIST04	NIST05	NIST08
+memory	34.3%	29.4%	22.2%
+chunking tag	66.4%	63.1%	58.4%

Table 2: Percentages of sentences that contain phrases generated by the proposed model.

Type	All		New	
	Total	Correct	Total	Correct
NP	81.0%	38.7%	46.0%	11.5%
VP	8.0%	1.7%	6.5%	0.8%
QP	10.8%	4.1%	6.2%	0.9%
Others	0.2%	0%	0.2%	0%
Sum	100%	44.5%	58.9%	13.2%

Table 3: Percentages of phrase categories to the total number of generated ones. "All" denotes all generated phrases, and "New" means new phrases that cannot be found in translations generated by the baseline system. "Total" is the total number of generated phrases and "Correct" denotes the fully correct ones.

4.1 Main Results

Table 1 reports main results of different models measured in terms of BLEU score. We observe that our implementation of RNNSearch outperforms Moses by 2.34 BLEU points. (+*memory*) which is the proposed model with the phrase memory obtains an improvement of 0.47 BLEU points over the baseline RNNSearch. With the source-side chunking tag feature, (+*memory*+*chunking tag*) outperforms the baseline RNNSearch by 1.07 BLEU points, showing the effectiveness of chunking syntactic categories on the selection of appropriate target phrases. From here on, we use "+*memory*+*chunking tag*" as the default setting in the following experiments if not otherwise stated.

Number of Sentences Affected by Generated Phrases We also check the number of translations that contain phrases generated by the proposed model, as shown in Table 2. As seen, a large portion of translations take the recommended phrases, and the number increases when the chunking tag feature is used.[7] Considering BLEU scores reported in Table 1, we believe that the chunking tag feature benefits the proposed model on its phrase generation.

4.2 Analysis on Generated Phrases

Syntactic Categories of Generated Phrases We first investigate which category of phrases is more likely to be selected by the proposed approach. There are some phrases, such as noun phrases (NPs, e.g., "national laboratory" and "vietnam airlines") and quantifier phrases (QPs, e.g., "15 seconds" and "two weeks") , that we expect to be favored by our approach. Statistics shown in Table 3 confirm our hypothesis. Let's first concern all generated phrases (i.e., column "All"): most selected phrases are noun phrases (81.0%) and quantifier phrases (10.8%). Among them, 44.5% percent of them are fully correct[8]. Specifically, NPs have relative higher generation accuracy (i.e., $47.8\% = 38.7\%/81.0\%$) while VPs have lower accuracy (i.e., $21.2\% = 1.7\%/8.0\%$). By looking into the wrong cases, we found most errors are related to verb tense, which is the drawback of SMT models.

Concerning the newly introduced phrases that cannot be found in baseline translations (i.e., column "New"), 13.2% of generated phrases are both new and fully correct, which contribute most to the performance improvement. We can also find that most newly introduced verb phrases and quantifier phrases are not correct, the patterns of which can be well learned by word-based NMT models.

[7] The numbers on NIST08 are relatively lower since part of the test set contains sentences from Web forums, which contain less multi-word expressions.

[8] Fully correct means that the generated phrases can be retrieved in corresponding references as a whole unit.

Words	All		New	
	Total	Correct	Total	Correct
2	66.2%	33.6%	34.9%	9.1%
3	20.7%	8.4%	13.4%	3.2%
4	7.4%	1.9%	5.4%	0.6%
⩾5	5.7%	0.6%	5.2%	0.3%

Table 4: Percentages of phrases with different word counts to the total number of generated ones.

Number of Words in Generated Phrases Table 4 lists the distribution of generated phrases based on the number of inside words. As seen, most generated phrases are short phrases (e.g., 2-gram and 3-gram phrases), which also contribute most to the new and fully correct phrases (i.e., $12.3\% = 9.1\% + 3.2\%$). Focusing on long phrases (e.g., order$\geqslant$ 4), most of them are newly introduced (10.6% out of 13.1%). Unfortunately, only a few portion of these phrases are fully correct, since long phrases have higher chance to contain one or two unmatched words.

SYSTEM	Test
+memory	32.95
+memory +NULL	31.63
+memory +chunking tag	33.55
+memory +chunking tag +NULL	30.81

Table 5: Additional experiment results on the translation task to directly measure the improvement obtained by the phrase generation. "+NULL" denotes that we replace the generated target phrases with a special symbol "NULL" in test sets. BLEU scores in the table are case insensitive.

Effect of Generated Phrases on Translation Performance Note that the proposed model benefits not only from fully matched phrases, but also from partially matched phrases. For example, the baseline system translates " 国家 航空 暨 太空 总署" in a word-by-word manner and outputs "state aviation and space department". The generated phrase provided by SMT is "national aviation and space administration", but the only correct reference is "national aeronautics and space administration". The generated phrase is not fully correct but still useful.

To directly measure the improvement obtained by the phrase generation, we replace the generated target phrases with a special symbol "NULL" in test sets. As shown in Table 5, when deleting the generated target phrases, ("+*memory*+*chunking tag*") and ("+*memory*") translation performances decrease by 2.74 BLEU points and 1.32 BLEU points respectively. Moreover, translation performances on NIST08 decrease less than those on NIST04 and NIST05 in both settings. The reason is that NIST08 which contains sentences from web data has little influence on generating target phrases which are provided from a different domain [9]. The overall results demonstrate that neural machine translation benefits from phrase translation.

4.3 Effect of Balancer

Weight	Test
Dynamic	33.55
Constant ($\lambda = 0.1$)	31.35

Table 6: Translation performance with a variety of balancing weight strategies. "Dynamic" is the proposed approach and "Constant ($\lambda = 0.1$)" denotes fixing the balancing weight to 0.1. BLEU scores in the table are case insensitive.

The balancer which is used to coordinate the phrase generation and word generation is very crucial for the proposed model. We conducted an additional experiment to validate the effectiveness of the neural network based balancer. We use the setting "+memory +chunking tag" as baseline system to conduct the experiments. In this experiment, we fixed the balancing weight λ (Equation 8) to 0.1 during training and testing and report the results. As shown in Table 6, we find that using the fixed value for the balancing weight (*Constant ($\lambda = 0.1$)*) decreases the translation performance sharply. This demonstrates that the neural network based balancer is an essential component for the proposed model.

4.4 Comparison to Word-Level Recommendations and Discussions

Our approach is related to our previous work (Wang et al., 2017) which integrates the SMT word-level knowledge into NMT. To make a comparison, we conducted experiments followed settings in (Wang et al., 2017). The comparison results are reported in Table 7. We find that our approach is marginally better than the word-level

[9]The parallel training data are mainly from news domain.

SYSTEM	Test
+word level recommendation	33.27
+memory +chunking tag	33.55

Table 7: Experiment results on the translation task. "+word level recommendation" is the proposed model in (Wang et al., 2017). BLEU scores in the table are case insensitive.

model proposed in (Wang et al., 2017) by 0.28 BLEU points.

In our approach, the SMT model translates source-side chunk phrases using the NMT decoding information. Although we use high-quality target phrases as phrasal recommendations, our approach still suffers from the errors in segmentation and chunking. For example, the target phrase "laptop computers" cannot be recommended by the SMT model if the Chinese phrase "手 提 电 脑" is not chunked as a phrase unit. This is the reason why some sentences do not have corresponding phrasal recommendations (Table 2). Therefore, our approach can be further enhanced if we can reduce the error propagations from the segmenter or chunker, for example, by using n-best chunk sequences instead of the single best chunk sequence.

Additionally, we also observe that some target phrasal recommendations have been also generated by the baseline system in a word-by-word manner. These phrases, even taken as parts of final translations by the proposed model, do not lead to improvements in terms of BLEU as they have already occurred in translations from the baseline system. For example, the proposed model successfully carries out the phrase generation mode to generate a target phrase "guangdong province" (the translation of Chinese phrase "广 东 省") which has appeared in the baseline system.

As external resources, e.g., bilingual dictionary, which are complementary to the SMT phrasal recommendations, are compatible with the proposed model, we believe that the proposed model will get further improvement by using external resources.

5 Related work

Our work is related to the following research topics on NMT:

Generating phrases for NMT In these studies, the generated NMT multi-word phrases are either from an SMT model or a bilingual dictionary. In syntactically guided neural machine translation (SGNMT), the NMT decoder uses phrase translations produced by the hierarchical phrase-based SMT system Hiero, as hard decoding constraints. In this way, syntactic phrases are generated by the NMT decoder (Stahlberg et al., 2016b). Zhang and Zong (2016) use an SMT translation system, which is integrated an additional bilingual dictionary, to synthesize pseudo-parallel sentences and feed the sentences into the training of NMT in order to translate low-frequency words or phrases. Tang et al. (2016) propose an external phrase memory that stores phrase pairs in symbolic forms for NMT. During decoding, the NMT decoder enquires the phrase memory and properly generates phrase translations. The significant differences between these efforts and ours are 1) that we dynamically generate phrase translations via an SMT model, and 2) that at the same time we modify the encoder to incorporate structural information to enhance the capability of NMT in phrase translation.

Incorporating linguistic information into NMT NMT is essentially a sequence to sequence mapping network that treats the input/output units, eg., words, subwords (Sennrich et al., 2016), characters (Chung et al., 2016; Costa-jussà and Fonollosa, 2016), as non-linguistic symbols. However, linguistic information can be viewed as the task-specific knowledge, which may be a useful supplementary to the sequence to sequence mapping network. To this end, various kinds of linguistic annotations have been introduced into NMT to improve its translation performance. Sennrich and Haddow (2016) enrich the input units of NMT with various linguistic features, including lemmas, part-of-speech tags, syntactic dependency labels and morphological features. García-Martínez et al. (2016) propose factored NMT using the morphological and grammatical decomposition of the words (factors) in output units. Eriguchi et al. (2016) explore the phrase structures of input sentences and propose a tree-to-sequence attention model for the vanilla NMT model. Li et al. (2017) propose to linearize source-side parse trees to obtain structural label sequences and explicitly incorporated the structural sequences into NMT, while Aharoni and Goldberg (2017) propose to incorporate target-side syntactic information into NMT by serializing the target sequences into linearized, lexicalized constituency trees. Zhang

et al. (2016) integrate topic knowledge into NMT for domain/topic adaptation.

Combining NMT and SMT A variety of approaches have been explored for leveraging the advantages of both NMT and conventional SMT. He et al. (2016) integrate SMT features with the NMT model under the log-linear framework in order to help NMT alleviate the limited vocabulary problem (Luong et al., 2015; Jean et al., 2015) and coverage problem (Tu et al., 2016). Arthur et al. (2016) observe that NMT is prone to making mistakes in translating low-frequency content words and therefore attempt at incorporating discrete translation lexicons into the NMT model, to alliterate the imprecise translation problem (Wang et al., 2017). Motivated by the complementary strengths of syntactical SMT and NMT, different combination schemes of Hiero and NMT have been exploited to form SGNMT (Stahlberg et al., 2016a,b). Wang et al. (2017) propose an approach to incorporate the SMT model into attention-based NMT. They combine NMT posteriors with SMT word recommendations through linear interpolation implemented by a gating function which dynamically assigns the weights. Niehues et al. (2016) propose to use SMT to pre-translate the inputs into target translations and employ the target pre-translations as input sequences in NMT. Zhou et al. (2017) propose a neural system combination framework to directly combine NMT and SMT outputs. The combination of NMT and SMT has been also introduced in interactive machine translation to improve the system's suggestion quality (Wuebker et al., 2016). In addition, word alignments from the traditional SMT pipeline are also used to improve the attention mechanism in NMT (Cohn et al., 2016; Mi et al., 2016; Liu et al., 2016).

6 Conclusion

In this paper, we have presented a novel model to translate source phrases and generate target phrase translations in NMT by integrating the phrase memory into the encoder-decoder architecture. At decoding, the SMT model dynamically generates relevant target phrases with contextual information provided by the NMT model and writes them to the phrase memory. Then the proposed model reads the phrase memory and uses the balancer to make probability estimations for the phrases in the phrase memory. Finally the NMT decoder selects a phrase from the phrase memory or a word from the vocabulary of the highest probability to generate. Experiment results on Chinese→English translation have demonstrated that the proposed model can significantly improve the translation performance.

Acknowledgments

We would like to thank three anonymous reviewers for their insightful comments, and also acknowledge Zhengdong Lu, Lili Mou for useful discussions. This work was supported by the National Natural Science Foundation of China (Grants No.61525205, 61373095 and 61622209).

References

Roee Aharoni and Yoav Goldberg. 2017. Towards string-to-tree neural machine translation. *arXiv preprint arXiv:1704.04743*.

Philip Arthur, Graham Neubig, and Satoshi Nakamura. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceedings of the 2016 Conference on EMNLP*.

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

Peter F Brown, Vincent J Della Pietra, Stephen A Della Pietra, and Robert L Mercer. 1993. The mathematics of statistical machine translation: Parameter estimation. *Computational linguistics*.

David Chiang. 2005. A hierarchical phrase-based model for statistical machine translation. In *Proceedings of the 43rd ACL*.

Kyunghyun Cho. 2015. Natural language understanding with distributed representation. *arXiv preprint*.

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 *Proceedings of the 2014 Conference on EMNLP*.

Junyoung Chung, Kyunghyun Cho, and Yoshua Bengio. 2016. A character-level decoder without explicit segmentation for neural machine translation. In *Proceedings of the 54th ACL*, pages 1693–1703, Berlin, Germany.

Trevor Cohn, Cong Duy Vu Hoang, Ekaterina Vymolova, Kaisheng Yao, Chris Dyer, and Gholamreza Haffari. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of NAACL 2016*, San Diego, California.

Marta R. Costa-jussà and José A. R. Fonollosa. 2016. Character-based neural machine translation. In *Proceedings of the 54th ACL*, pages 357–361, Berlin, Germany.

Akiko Eriguchi, Kazuma Hashimoto, and Yoshimasa Tsuruoka. 2016. Tree-to-sequence attentional neural machine translation. In *Proceedings of the 54th ACL*, pages 823–833, Berlin, Germany.

Mercedes García-Martínez, Loïc Barrault, and Fethi Bougares. 2016. Factored neural machine translation. *arXiv preprint arXiv:1609.04621*.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th ACL*, Berlin, Germany.

Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016. Pointing the unknown words. In *Proceedings of ACL 2016*, Berlin, Germany.

Wei He, Zhongjun He, Hua Wu, and Haifeng Wang. 2016. Improved neural machine translation with smt features. In *Proceedings of the 30th AAAI Conference on Artificial Intelligence*.

Sébastien Jean, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. On using very large target vocabulary for neural machine translation. In *Proceedings of the 53rd ACL and the 7th IJCNLP*.

Nal Kalchbrenner and Phil Blunsom. 2013. Recurrent continuous translation models. In *Proceedings of the 2013 Conference on EMNLP*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondrej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th ACL Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic.

Philipp Koehn, Franz Josef Och, and Daniel Marcu. 2003. Statistical phrase-based translation. In *Proceedings of the 2003 NAACL*.

Junhui Li, Deyi Xiong, Zhaopeng Tu, Muhua Zhu, Min Zhang, and Guodong Zhou. 2017. Modeling source syntax for neural machine translation. *arXiv preprint arXiv:1705.01020*.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016*, pages 3093–3102, Osaka, Japan.

Minh-Thang Luong and Christopher D. Manning. 2016. Achieving open vocabulary neural machine translation with hybrid word-character models. In *Proceedings of the 54th ACL*.

Thang Luong, Ilya Sutskever, Quoc Le, Oriol Vinyals, and Wojciech Zaremba. 2015. Addressing the rare word problem in neural machine translation. In *Proceedings of the 53rd ACL and the 7th IJCNLP*.

Haitao Mi, Zhiguo Wang, and Abe Ittycheriah. 2016. Supervised attentions for neural machine translation. In *Proceedings of EMNLP 2016*, pages 2283–2288, Austin, Texas.

Jan Niehues, Eunah Cho, Thanh-Le Ha, and Alex Waibel. 2016. Pre-translation for neural machine translation. In *Proceedings of COLING 2016*, Osaka, Japan.

Franz Josef Och. 2003. Minimum error rate training in statistical machine translation. In *Proceedings of the 41st ACL*.

Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. 2013. On the difficulty of training recurrent neural networks. In *Proceedings of the 30th International Conference on Machine Learning, ICML 2013*, pages 1310 – –1318.

Ivan A Sag, Timothy Baldwin, Francis Bond, Ann Copestake, and Dan Flickinger. 2002. Multiword expressions: A pain in the neck for nlp. In *International Conference on Intelligent Text Processing and Computational Linguistics*, pages 1–15. Springer.

Rico Sennrich and Barry Haddow. 2016. Linguistic input features improve neural machine translation. In *Proceedings of the First Conference on Machine Translation*, pages 83–91, Berlin, Germany.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th ACL*, pages 1715–1725, Berlin, Germany.

Felix Stahlberg, Adrià de Gispert, Eva Hasler, and Bill Byrne. 2016a. Neural machine translation by minimising the bayes-risk with respect to syntactic translation lattices. *arXiv preprint arXiv:1612.03791*.

Felix Stahlberg, Eva Hasler, Aurelien Waite, and Bill Byrne. 2016b. Syntactically guided neural machine translation. In *Proceedings of the 54th ACL (Volume 2: Short Papers)*, Berlin, Germany.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in Neural Information Processing Systems 27*.

Yaohua Tang, Fandong Meng, Zhengdong Lu, Hang Li, and Philip LH Yu. 2016. Neural machine translation with external phrase memory. *arXiv preprint arXiv:1606.01792*.

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

Zhaopeng Tu, Yang Liu, Lifeng Shang, Xiaohua Liu, and Hang Li. 2017b. Neural machine translation with reconstruction. In *Proceedings of the 31th AAAI Conference on Artificial Intelligence*.

Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation. In *Proceedings of the 54th ACL*.

Aline Villavicencio, Francis Bond, Anna Korhonen, and Diana McCarthy. 2005. Introduction to the special issue on multiword expressions: Having a crack at a hard nut. *Computer Speech & Language*, 19(4):365–377.

Xing Wang, Zhengdong Lu, Zhaopeng Tu, Hang Li, Deyi Xiong, and Min Zhang. 2017. Neural machine translation advised by statistical machine translation. In *Proceedings of the 31th AAAI Conference on Artificial Intelligence*.

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*.

Joern Wuebker, Spence Green, John DeNero, Sasa Hasan, and Minh-Thang Luong. 2016. Models and inference for prefix-constrained machine translation. In *Proceedings of the 54th ACL*, pages 66–75, Berlin, Germany.

Nianwen Xue and Libin Shen. 2003. Chinese word segmentation as lmr tagging. In *Proceedings of the Second SIGHAN Workshop on Chinese Language Processing*, pages 176–179, Sapporo, Japan.

Matthew D Zeiler. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.

Jiajun Zhang and Chengqing Zong. 2016. Bridging neural machine translation and bilingual dictionaries. *arXiv preprint arXiv:1610.07272*.

Jian Zhang, Liangyou Li, Andy Way, and Qun Liu. 2016. Topic-informed neural machine translation. In *Proceedings of COLING 2016*, pages 1807–1817, Osaka, Japan.

Long Zhou, Wenpeng Hu, Jiajun Zhang, and Chengqing Zong. 2017. Neural system combination for machine translation. *arXiv preprint arXiv:1704.06393*.

Jingbo Zhu, Muhua Zhu, Qiang Wang, and Tong Xiao. 2015. Niuparser: A chinese syntactic and semantic parsing toolkit. In *Proceedings of ACL-IJCNLP 2015 System Demonstrations*, Beijing, China.