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.
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 \( x = x_1, x_2, ..., x_{T_x} \) and transforms it into word annotations of the entire source sentence \( h = h_1, h_2, ..., h_{T_x} \). The decoder uses the annotations to emit a target sentence \( y = y_1, y_2, ..., y_{T_y} \) in a word-by-word manner.

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

\[
P(y|x) = \prod_{i=1}^{T_y} P(y_i|y_{<i}, x) \tag{1}
\]

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

\[
P(y_i|y_{<i}, x) = \text{softmax}(f(s_i, y_{i-1}, c_i)) \tag{2}
\]

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

\[
s_i = g(s_{i-1}, y_{i-1}, c_i) \tag{3}
\]

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 \( h \):

\[
c_i = \sum_{j=1}^{T_x} \alpha_{t,j} h_j \tag{4}
\]

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:

\[
C(\theta) = \sum_{n=1}^{N} \sum_{i=1}^{T_y} \log P(y^n_i|y^n_{<i}, x^n) \tag{5}
\]

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

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

\[
\hat{y} = \arg\max_{y} P(y|x) \tag{6}
\]
The balancing weight $\lambda$ 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 $y_{i-1}$:

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

where $\sigma(\cdot)$ is a sigmoid function and $f_b(\cdot)$ is the activation function. Intuitively, the weight $\lambda$ can be treated as the estimated importance of the phrase to be generated. We expect $\lambda$ 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:

$$[E^w x_i, E^t t_i]$$  \hspace{1cm} (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 $E^w \in \mathbb{R}^{dw \times |V|}$ is a word embedding matrix and $dw$ is the word embedding dimensionality, $E^t \in \mathbb{R}^{dt \times |V^{T,AC}|}$ 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 $P_{\text{phrase}}$ from the memory.

**Writing to Phrase Memory** Given a partial translation $y_{<i} = \{y_1, y_2, \ldots, y_{i-1}\}$ 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:

$$SMT_{\text{score}}(p_i|y_{<i}, x) = \sum_{m=1}^{M} w_m h_m(p_i, x(p_i))$$

(10)

where $p_i$ is a target phrase and $x(p_i)$ is its corresponding source span. $h_m(p_i, x(p_i))$ 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.  

**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 = $t(p_i)$ denotes the current decoding step):

$$\text{score}_{\text{phrase}}(p_i) = g_s(e(p_i), s_i, y_{i-1}, c_i)$$

(11)

where $g_s(\cdot)$ is either an identity or a non-linear function. $e(p_i)$ is the representation of phrase $p_i$, 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 $p_i$ is calculated as

$$P_{\text{phrase}}(p_i) = \text{softmax}(\text{score}_{\text{phrase}}(p_i))$$

(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 $\{[x^n, y^n]\}_{n=1}^N$:

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

(13)

where $P(y^n|x^n)$ is defined in Equation 7. Ideally, the trained model is expected to produce a higher balance weight $\lambda$ and phrase probability $P_{\text{phrase}}$ 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 $P_{\text{phrase}}$ for all phrases in the phrase memory.

---

2Bilingual 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 \( P_{\text{word}} \) for all words in NMT vocabulary. Then the balancer outputs a balancing weight \( \lambda_i \), which is used to scale the phrase and word probabilities: 
\[ \lambda_i \times P_{\text{phrase}} \text{ and } (1 - \lambda_i) \times P_{\text{word}}. \]

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 \( \rho \) and \( \epsilon \) 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

\(^5\)http://kheafield.com/code/kenlm/

\(^6\)http://www.niuparser.com/
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 ($p < 0.01$).

<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>NIST04</th>
<th>NIST05</th>
<th>NIST08</th>
<th>Avg</th>
</tr>
</thead>
<tbody>
<tr>
<td>Moses</td>
<td>34.74</td>
<td>31.99</td>
<td>23.69</td>
<td>30.14</td>
</tr>
<tr>
<td>RNNSearch</td>
<td>37.80</td>
<td>34.70</td>
<td>24.93</td>
<td>32.48</td>
</tr>
<tr>
<td>+memory</td>
<td>38.21</td>
<td>35.15</td>
<td>25.48†</td>
<td>32.95</td>
</tr>
<tr>
<td>+memory +chunking tag</td>
<td>38.83‡</td>
<td>35.72‡</td>
<td>26.09‡</td>
<td>33.55</td>
</tr>
</tbody>
</table>

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

<table>
<thead>
<tr>
<th></th>
<th>NIST04</th>
<th>NIST05</th>
<th>NIST08</th>
</tr>
</thead>
<tbody>
<tr>
<td>+memory</td>
<td>34.3%</td>
<td>29.4%</td>
<td>22.2%</td>
</tr>
<tr>
<td>+chunking tag</td>
<td>66.4%</td>
<td>63.1%</td>
<td>58.4%</td>
</tr>
</tbody>
</table>

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.

<table>
<thead>
<tr>
<th>Type</th>
<th>All Total</th>
<th>Correct</th>
<th>New Total</th>
<th>Correct</th>
</tr>
</thead>
<tbody>
<tr>
<td>NP</td>
<td>81.0%</td>
<td>38.7%</td>
<td>46.0%</td>
<td>11.5%</td>
</tr>
<tr>
<td>VP</td>
<td>8.0%</td>
<td>1.7%</td>
<td>6.5%</td>
<td>0.8%</td>
</tr>
<tr>
<td>QP</td>
<td>10.8%</td>
<td>4.1%</td>
<td>6.2%</td>
<td>0.9%</td>
</tr>
<tr>
<td>Others</td>
<td>0.2%</td>
<td>0%</td>
<td>0.2%</td>
<td>0%</td>
</tr>
<tr>
<td>Sum</td>
<td>100%</td>
<td>44.5%</td>
<td>58.9%</td>
<td>13.2%</td>
</tr>
</tbody>
</table>

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. 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. 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

---

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.
be well learned by word-based NMT models.

| Words | All      | New
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Total Correct</td>
<td>Total Correct</td>
</tr>
<tr>
<td>2</td>
<td>66.2%</td>
<td>33.6%</td>
</tr>
<tr>
<td>3</td>
<td>20.7%</td>
<td>8.4%</td>
</tr>
<tr>
<td>4</td>
<td>7.4%</td>
<td>1.9%</td>
</tr>
<tr>
<td>⩾5</td>
<td>5.7%</td>
<td>0.6%</td>
</tr>
</tbody>
</table>

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 ⩾ 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.

**4.3 Effect of Balancer**

<table>
<thead>
<tr>
<th>Weight</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>Dynamic</td>
<td>33.55</td>
</tr>
<tr>
<td>Constant (λ = 0.1)</td>
<td>31.35</td>
</tr>
</tbody>
</table>

Table 6: Translation performance with a variety of balancing weight strategies. “Dynamic” is the proposed approach and “Constant (λ = 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 (λ = 0.1)) decreases the translation performance sharply. This demonstrates that the neural network based balancer is an essential component for the proposed model.

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.

**4.4 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 set-

---

9The parallel training data are mainly from news domain.
<table>
<thead>
<tr>
<th>SYSTEM</th>
<th>Test</th>
</tr>
</thead>
<tbody>
<tr>
<td>+word level recommendation</td>
<td>33.27</td>
</tr>
<tr>
<td>+memory +chunking tag</td>
<td>33.55</td>
</tr>
</tbody>
</table>

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.

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:

Incorporating linguistic information into NMT

NMT is essentially a sequence to sequence mapping network that treats the input/output units, e.g., 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

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.
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 Zhongdong 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


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


