Neural Machine Translation with External Phrase Memory

Yaohua Tang
The University of Hong Kong
3 PUBLICATIONS **1** CITATION
SEE PROFILE

Fandong Meng
Chinese Academy of Sciences
9 PUBLICATIONS **11** CITATIONS
SEE PROFILE

Hang Li
Mount Sinai Hospital, Toronto
185 PUBLICATIONS **4,550** CITATIONS
SEE PROFILE

Philip L H Yu
The University of Hong Kong
90 PUBLICATIONS **681** CITATIONS
SEE PROFILE

Neural Machine Translation with External Phrase Memory

Yaohua Tang
The University of Hong Kong
tangyh@hku.hk

Fandong Meng
Institute of Computing Technology
Chinese Academy of Sciences
mengfandong@ict.ac.cn

Zhengdong Lu and **Hang Li**
Noah's Ark Lab, Huawei Technologies
Lu.Zhengdong@huawei.com
HangLi.HL@huawei.com

Philip L.H. Yu
plhyu@hku.hk
The University of Hong Kong

Abstract

In this paper, we propose phraseNet, a neural machine translator with a phrase memory which stores phrase pairs in symbolic form, mined from corpus or specified by human experts. For any given source sentence, phraseNet scans the phrase memory to determine the candidate phrase pairs and integrates tagging information in the representation of source sentence accordingly. The decoder utilizes a mixture of word-generating component and phrase-generating component, with a specifically designed strategy to generate a sequence of multiple words all at once. The phraseNet not only approaches one step towards incorporating external knowledge into neural machine translation, but also makes an effort to extend the word-by-word generation mechanism of recurrent neural network. Our empirical study on Chinese-to-English translation shows that, with carefully-chosen phrase table in memory, phraseNet yields 3.45 BLEU improvement over the generic neural machine translator.

1 Introduction

Neural machine translation (NMT), although only proposed recently, has shown great potential, and arguably surpassed statistical machine translation on tasks like English-German translation (Sennrich et al., 2015). In addition to its superior ability in modeling the semantics of source sentence and language fluency of the target sentence, NMT as a framework also has remarkable flexibility in accommodating other form of knowledge.

In this paper, we explore the possibility of equipping regular neural machine translator with an external memory storing rules that specify phrase-level correspondence between the source and target languages. Those rules are in symbolic form, which can be either extracted from a parallel corpus or given by experts. We tailor the encoder, decoder and the attention model of the neural translator to help locate phrases in the source and generate their translations in the target. The proposed model is called phraseNet. phraseNet is not only one step towards incorporating external knowledge in to neural machine translation, but also an effort to extend the word-by-word generation mechanism of recurrent neural network.

1.1 Model Overview

The overall diagram of phraseNet is displayed in Figure 1. Basically, for a given source sentence, phraseNet first encodes the sentence to a representation with an RNN encoder, scans the phrase memory to select candidate phrase pairs and then tags the source representation accordingly (Section 3.2). phraseNet then generates both words and phrases with an RNN decoder. It dynamically determines at each time step with its probabilistic model consisting of a mixture of word-generation mode and phrase-generation mode (Section 3). To maintain the state consistency of RNN when running in different modes, the decoder of phraseNet will go through "idle run" (Section 3.6) after generating a multiple-word phrase.
Our contribution is of three-folds:

1. we propose an end-to-end learning algorithm for neural machine translation with an external

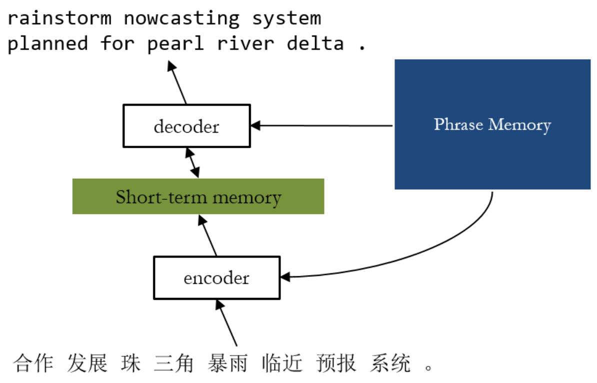

rainstorm nowcasting system
planned for pearl river delta .
decoder
Phrase Memory
Short-term memory
encoder
合作 发展 珠 三角 暴雨 临近 预报 系统 。

Figure 1: The overall diagram of phraseNet.

phrase memory, which to our knowledge it is the first effort in this direction;

2. we propose a way to handle the generation of multiple words with an RNN decoder;

3. our empirical studies on Chinese-English translation tasks show the efficacy of our models: phraseNet achieves on average 3.45 BLEU improvement over its generic counterpart.

RoadMap The remainder of this paper is organized in the following way. In Section 2, we will give a brief introduction to attention-based neural machine translation as the background. In Section 3, we will introduce phraseNet, including its two variants. In Section 4, we will report our experiments on applying phraseNet to Chinese-English translation tasks. Then in Section 5 and 6, we will give a brief review of related work and conclude the paper.

2 Background

Our work is built upon the attention-based neural machine translation model that learns to align and translate jointly (Bahdanau et al., 2015), which will be referred to as RNNsearch.

RNNsearch uses a bidirectional RNN (Schuster and Paliwal, 1997) to encode the source sentence. It consists of two independent RNNs. The forward RNN reads the source sentence from left to right $\overrightarrow{\mathbf{h}} = (\overrightarrow{\mathbf{h}}_1, \dots, \overrightarrow{\mathbf{h}}_{T_x})$. The backward RNN reads the source sentence from right to left $\overleftarrow{\mathbf{h}} = (\overleftarrow{\mathbf{h}}_1, \dots, \overleftarrow{\mathbf{h}}_{T_x})$. The representation of the source sentence $\mathbf{h}$ is then defined as the concatenation of $\overrightarrow{\mathbf{h}}$ and $\overleftarrow{\mathbf{h}}$. Each element in $\mathbf{h}$ contains information about the source sentence, focusing on the parts surrounding the corresponding word.

At decoding time t, the attention model uses $\mathbf{s}_{t-1}$ (the RNN states) and $\mathbf{e}_{y_{t-1}}$ (the embedding of previous target word y_{t-1}) to "query" the encoded $\mathbf{h}$, marks each element of $\mathbf{h}$ a score $e_{tj} = f(\mathbf{s}_{t-1}, \mathbf{h}_j, \mathbf{e}_{y_{t-1}})$. The non-linear function f can take on many forms, but we concatenate the three inputs and feed it to a neural network with one hidden layer and tanh as activation function. The scores are then normalized to $\{\alpha_{tj}\}$, serving as the weights of $\{\mathbf{h}_j\{$ to the target word, which then gives the context vector $\mathbf{h}_j$, $\mathbf{c}_t = \sum_{j=1}^{T_x} \alpha_{tj}\mathbf{h}_j$. The context vector $\mathbf{c}_t$ is then used to update the hidden state of the decoder:

$$\mathbf{s}_t = f_u(\mathbf{s}_{t-1}, \mathbf{c}_t, \mathbf{e}_{y_{t-1}}), \tag{1}$$

where the function f_u is GRU (Cho et al., 2014; Chung et al., 2014). To predict a target word, the decoder combines $\mathbf{s}_t$, $\mathbf{c}_t$ and $\mathbf{e}_{y_{t-1}}$, feeds to a one-layer MLP with tanh as activation function, followed by a softmax function,

$$p(y_t = y_i | \mathbf{y}_{<t}, \mathbf{x}; \theta) \propto \exp\left\{\mathbf{v}_i^T \mathbf{W}_o \tanh\left(\mathbf{U}_o \mathbf{s}_{t-1} + \mathbf{C}_o \mathbf{c}_t + \mathbf{V}_o \mathbf{e}_{y_{t-1}}\right)\right\}, \tag{2}$$

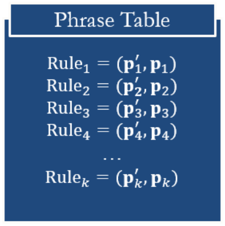
where $\mathbf{W}_o$, $\mathbf{U}_o$, $\mathbf{C}_o$ and $\mathbf{V}_o$ are weight matrices. $\mathbf{v}_i$ is an one-hot indicator vector for y_i.

3 Models

In this section, we will give more details of phraseNet, more specifically on the preprocessing, encoder and decoders. With two different variants of the mixture models used in decoder, we naturally have two variants of phraseNet, namely phraseNet$_{gate}$ and phraseNet$_{softmax}$.

3.1 Preprocessing

The phrase table $\mathcal{P}$ is a list of rules. Each rule contains a source phrase $\mathbf{p}_k^{'}$ and its translation, a target phrase $\mathbf{p}_k$. Figure 2 gives an example of the phrase table. For simplicity, we first limit ourselves to a subset of rules with the strongest source-target correspondence, which is in contrast to that in phrase-based statistical machine translation (SMT). In SMT, a phrase could have multiple translations with a probability distribution over them. Here we restrict that for each source phrase in $\mathcal{P}$, it has only one translation with probability almost equal to 1. This will limit the size of $\mathcal{P}$ but can guarantee that


Phrase Table

$\text{Rule}_1 = (\mathbf{p}_1', \mathbf{p}_1)$
$\text{Rule}_2 = (\mathbf{p}_2', \mathbf{p}_2)$
$\text{Rule}_3 = (\mathbf{p}_3', \mathbf{p}_3)$
$\text{Rule}_4 = (\mathbf{p}_4', \mathbf{p}_4)$
$\dots$
$\text{Rule}_k = (\mathbf{p}_k', \mathbf{p}_k)$

Figure 2: The phrase table $\mathcal{P}_{\mathbf{xy}}$. For the k^{th} pair $(\mathbf{p}_k^{'}, \mathbf{p}_k)$, $\mathbf{p}_k^{'}$ stands for the source phrase and $\mathbf{p}_k$ for the target phrase.

the feasible rules are "reliable" enough and greatly simplifies the model design and training.

We will introduce the collection of such a phrase table $\mathcal{P}$ later. At the moment, let us assume that we have the table $\mathcal{P}$ which will be utilized to preprocess the sentence pair before encoding. In order to tag a source sentence $\mathbf{x} = (x_1, x_2, \dots, x_{T_x})$, we need to locate its contained phrases. Hence we will find out the rules in $\mathcal{P}$ whose source phrase appears in $\mathbf{x}$, and denote these rules as $\mathcal{P}_{\mathbf{x}}$. To calculate the likelihood during training, we also need to find out the rules in $\mathcal{P}_{\mathbf{x}}$ whose target phrase appears in the target sentence $\mathbf{y}$, denote as $\mathcal{P}_{\mathbf{xy}}$ ($\mathcal{P}_{\mathbf{xy}} \subset \mathcal{P}_{\mathbf{x}}$).

For simplicity, we remove from $\mathcal{P}_{\mathbf{x}}$ the short rules that intersect with the others, which means if the source phrases of two rules are overlapped, we remove the rule whose source phrase has fewer words. We choose at most n_p (a hype-parameter) phrases for each sentence with the maximum coverage.

The non-overlapping rules in $\mathcal{P}_{\mathbf{x}}$ and $\mathcal{P}_{\mathbf{xy}}$ split the words of source and target sentences into groups as showing in Figure 3. The words in source sentence $\mathbf{x}$ are split into two groups, phrases $\mathcal{P}_{\mathbf{x}}$ and words not-in-phrases $\mathcal{W}_x$, while the words in target sentence $\mathbf{y}$ are split into two groups, phrases $\mathcal{P}_{\mathbf{xy}}$ and words not-in-phrases $\mathcal{W}_y$.

3.2 Encoder

We use the regular encoder from RNNsearch, but add tags to $\mathbf{h}_i$ (Meng et al., 2015), $\mathbf{h}_i^{'} = [\mathbf{h}_i,\ tag_i]$. The tags are used to help the model locate and discriminate different phrases.

Each tag_i is an indicator vector with length n_p. For example in Figure 3, suppose $n_p = 5$ and we find three source phrases $\mathbf{p}_1^{'} = (x_2, x_3), \mathbf{p}_2^{'} = (x_6, x_7, x_8), \mathbf{p}_3^{'} = (x_{10}, x_{11})$ in sentence $\mathbf{x}$, we concatenate a tag vector $(1, 0, 0, 0, 0)$ to each of

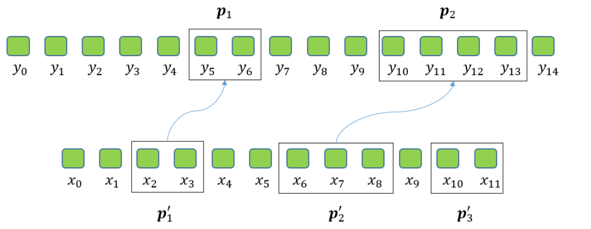

$\mathbf{p}_1$
$\mathbf{p}_2$
y_0 y_1 y_2 y_3 y_4 y_5 y_6 y_7 y_8 y_9 y_{10} y_{11} y_{12} y_{13} y_{14}
x_0 x_1 x_2 x_3 x_4 x_5 x_6 x_7 x_8 x_9 x_{10} x_{11}
$\mathbf{p}_1'$
$\mathbf{p}_2'$
$\mathbf{p}_3'$

Figure 3: Example of a sentence pair being split into groups. $\mathcal{P}_{\mathbf{x}} = \{(\mathbf{p}_1^{'}, \mathbf{p}_1), (\mathbf{p}_2^{'}, \mathbf{p}_2), (\mathbf{p}_3^{'}, \mathbf{p}_3)\}$, $\mathcal{P}_{\mathbf{xy}} = \{(\mathbf{p}_1^{'}, \mathbf{p}_1), (\mathbf{p}_2^{'}, \mathbf{p}_2)\}$. $\mathcal{W}_x = \{x_0, x_1, x_4, x_5, x_9\}$, $\mathcal{W}_y = \{y_0, y_1, y_2, y_3, y_4, y_7, y_8, y_9, y_{14}\}$.

$\mathbf{h}_2, \mathbf{h}_3$, concatenate a vector $(0, 1, 0, 0, 0)$ to each of $\mathbf{h}_6, \mathbf{h}_7, \mathbf{h}_8$, concatenate a vector $(0, 0, 1, 0, 0)$ to $\mathbf{h}_{10}$ and $\mathbf{h}_{11}$. For all other not-in-phrase words $\mathcal{W}_x$, we also add a trivial tag vector $(0, 0, 0, 0, 0)$ to their $\mathbf{h}_i$. Figure 4 shows an example of such concatenating.

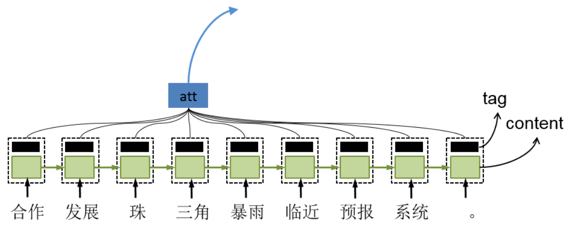


Figure 4: The diagram for encoder.

3.3 Decoder

Unlike the decoder of RNNsearch that only has *word mode*, our decoders also have *phrase mode*. For a two-word target phrase $\mathbf{p}_t = (y_t, y_{t+1})$, it can either be generated by *word mode* (one by one) or *phrase mode* (as a whole). In our models, we add upon the RNNsearch another component which has two functions, (1) makes decision between *phrase mode* and *word mode*, (2) chooses the right target phrase if the decision is *phrase mode*.

With the help of attention model, the new component tries to capture the signals from the encoded representations of a source sentence and translate part of the source sentence (source phrase) directly to the target output as a whole at proper decoding moments. The tags added in Section 3.2 will play an important role in the process. If we view $\mathbf{h}$ as a short-term memory as it changes from sentence to sentence, then the phrase table could be called a phrase memory which is a long-term memory. The decoder queries the short-term memory to choose the segment of source while it consults the phrase memory to choose the right phrase.

We have two model variants, namely phraseNet$_{gate}$ and phraseNet$_{softmax}$, each corresponding to a different implementation of the mixture model in decoder.

3.4 phraseNet$_{gate}$

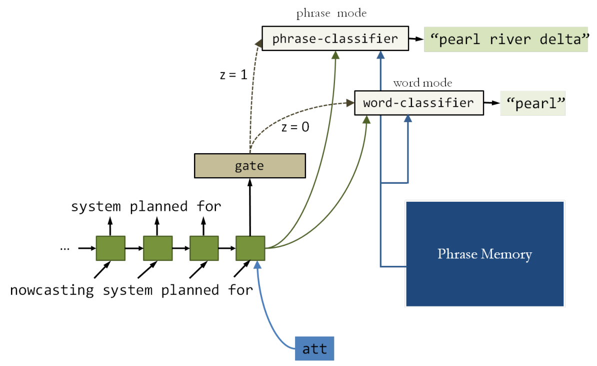


Figure 5: The diagram for phraseNet$_{gate}$.

The decoder for phraseNet$_{gate}$ is illustrated in Figure 5. In phraseNet$_{gate}$, the decoder will first use a gate f_z to determine the mode at time t by issuing a binary indicator variable ($z_t \in \{0, 1\}$), where 0 represents *word mode* and 1 represents *phrase mode*. Then for each mode, it will calculate the word probabilities and phrase probabilities respectively. In word mode, a classifier f_w outputs a probability distribution over the words in target vocabulary $\mathcal{V}$; while in phrase-mode a phrase classifier $f_\mathbf{p}$ determines the probability distribution over the phrases in $\mathcal{P}_x$. The final probability of output is given by the probabilities of modes as well as the probability of individual output generated in each mode. It is a mixture model since a phrase like `china daily` can be generated in both modes, and the final probability of it is therefore the sum of the probabilities of it being generated from each. For a target phrase that is not in word vocabulary, i.e., it contains `UNK`, the probability of that can only be from the phrase mode.

One snapshot of the decoding is the following. Let us suppose that at time $t = 1$, the decoder has generated the word y_1 in the *word mode*, and the current state is $\mathbf{s}_1$, moving to time $t = 2$. With the state $\mathbf{s}_1$, the attention model (same as in RNNsearch) first generates the context $\mathbf{c}_2$ as a weighted sum of the $\mathbf{h}$ (from encoder). With $\mathbf{s}_1$, $\mathbf{c}_2$ and $\mathbf{e}_{y_1}$, the decoder could move to update the next state $\mathbf{s}_2$ and generating the next word (or phrase). More specifically, in phraseNet$_{gate}$, the state $\mathbf{s}_2$ is updated in the same way as in RNNsearch (Equation (1)). The generation of the next word/phrase can be described as follows, let's denote $S_t = (\mathbf{s}_t, \mathbf{c}_t, \mathbf{e}_{y_{t-1}})$:

1. STEP-1: generate the decision variable z_2 with

$$\begin{aligned} p(z_2 = 1|\mathbf{S}_2; \theta) &= f_z(\mathbf{S}_2) \\ p(z_2 = 0|\mathbf{S}_2; \theta) &= 1 - f_z(\mathbf{S}_2) \end{aligned}$$

2. STEP-2a: if $z_2 = 0$ (*word mode*), generate a word based on $\mathbf{s}_2$ with the regular word vocabulary $\mathcal{V}$, same as RNNsearch (Equation 2);
3. STEP-2b: if $z_2 = 1$ (*phrase mode*), generate target phrase $\mathbf{p}_j \in \mathcal{P}_x$ with probability:

$$p_\mathbf{p}(y_2 = \mathbf{p}_j|\mathbf{S}_2, 1; \theta) \propto \\ \exp\left\{\mathbf{u}_j^T \mathbf{W}_p \tanh\left(\mathbf{U}_p \mathbf{s}_2 + \mathbf{C}_p \mathbf{c}_2 + \mathbf{V}_p \mathbf{e}_{y_1}\right)\right\},$$

where $\mathbf{W}_p$, $\mathbf{U}_p$, $\mathbf{C}_p$ and $\mathbf{V}_p$ are weight matrices. $\mathbf{u}_j$ is an one-hot indicator vector for $\mathbf{p}_j$.

4. STEP-3: calculate the final probabilities and sample the next word (or phrase):

$$\begin{aligned} p(y_2 = w_i) &= p(z_2 = 0|\mathbf{S}_2; \theta) p(w_i|\mathbf{S}_2, 0; \theta) \\ p(y_2 = \mathbf{p}_j) &= p(z_2 = 1|\mathbf{S}_2; \theta) p(\mathbf{p}_j|\mathbf{S}_2, 1; \theta) \\ p(y_2) &= \begin{bmatrix} p(y_2 = w) \\ p(y_2 = \mathbf{p}) \end{bmatrix}, \end{aligned}$$

where the size of $p(y_2)$ is n_p plus the number of words in vocabulary $\mathcal{V}$. The next word or phrase will be sampled according to $p(y_2)$. If next generation is a phrase, the decoder will go through an "idle run" process (Section 3.6) to generate words in $\mathbf{p}_2$, after that the decoder replaces the tags tag_i of those source words of $\mathbf{p}_2^{'}$ to all-zero vectors as $\mathbf{p}_2^{'}$ has already been decoded.

Similar to (Gulcehre et al., 2016a), in phraseNet$_{gate}$, f_z is a three-layered neural network using noisy-tanh activation for the first two layers (Gulcehre et al., 2016b) and we add residual connection (He et al., 2015) from the first layer to the second hidden layer, The output layer uses sigmoid as activation function.

3.5 phraseNet$_{softmax}$

With phraseNet$_{gate}$ the decision of phrase mode is made before seeing the actual content of the target phrase, which fails to make use of the language model and semantic relevance on the target side. To address this drawback we devise phraseNet$_{softmax}$, which takes the candidate phrases and candidate words in the same softmax, as illustrated in Figure 6. To do this, all the phrases need to embedded as vectors, where the embedding model is also learned in the NMT training. It is also worth to mention that phraseNet$_{softmax}$ can potentially handle the case where one source phrase may correspond to multiple candidate target phrases, since the decoder can distinguish them based on their content. This modeling advantage, however, will not be explored in this paper.

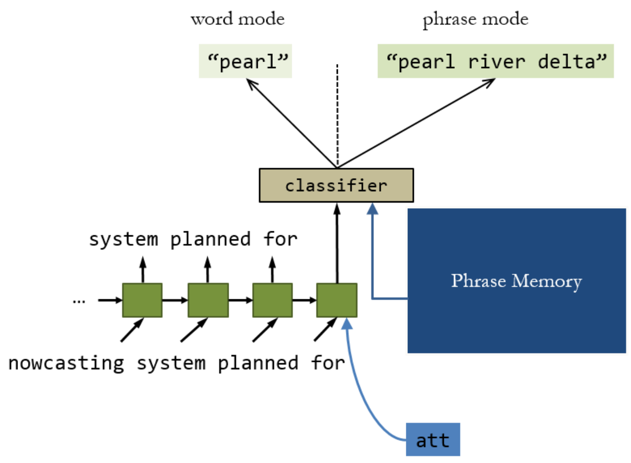


Figure 6: The diagram for phraseNet$_{softmax}$.

Given a rule $(\mathbf{p}_k^{'}, \mathbf{p}_k)$, we have several choices to calculate its embedding. In this paper, we choose to use a separate backward RNN to encode $\mathbf{p}_k$ and choose the last state as the embedding for it. That way, the embedding will keep more information of the first word of $\mathbf{p}_k$, therefore facilitate a potential language model in scoring $(y_{i-1}, \mathbf{s}_i, \mathbf{p}_k)$.

Suppose that at time $t = 1$, the decoder has generated the word y_1 in the *word mode*, and the current state is $\mathbf{s}_1$, moving to time $t = 2$. With the state $\mathbf{s}_1$, the decoder makes an attentive read to $\mathbf{h}$ to obtain $\mathbf{c}_2$. The state $\mathbf{s}_2$ is updated in the same way as in Equation (1). The generation of the next word/phrase can be described as follows:

1. STEP-1: calculate the word score for each word in $\mathcal{V}$,

$$\psi_{w_i} = \mathbf{v}_i^T \mathbf{W}_w \tanh(\mathbf{U}_w \mathbf{s}_2 + \mathbf{C}_w \mathbf{c}_2 + \mathbf{V}_w \mathbf{e}_{y_1}),$$

   where $\mathbf{W}_w$, $\mathbf{U}_w$, $\mathbf{C}_w$ and $\mathbf{V}_w$ are weight matrices.

2. STEP-2: calculate the phrase score for each phrase in $\mathcal{P}_x$

$$\psi_{\mathbf{p}_j} = \mathbf{W}_q \tanh(\mathbf{U}_q \mathbf{s}_2 + \mathbf{C}_q \mathbf{c}_2 + \mathbf{V}_q \mathbf{e}_{y_1} + \mathbf{R}_q \mathbf{e}_{\mathbf{p}_j}),$$

   where $\mathbf{e}_{\mathbf{p}_j}$ is the embeddings of rule $(\mathbf{p}_j^{'}, \mathbf{p}_j)$. $\mathbf{W}_q$, $\mathbf{U}_q$, $\mathbf{C}_q$, $\mathbf{V}_q$ and $\mathbf{R}_q$ are weight matrices.

3. STEP-3: calculate the probabilities of all words and phrases through softmax

$$p(y_2 | \mathbf{s}_2, \mathbf{c}_2, \mathbf{e}_{y_1}) = \text{softmax}([\psi_w, \psi_{\mathbf{p}}])$$

   In the softmax, the phrases will compete directly with words, which is different from the phraseNet$_{gate}$ where phrase probabilities and word probabilities are calculated independently. While phraseNet$_{gate}$ has difficulties in calculating the scores for each phrase, phraseNet$_{softmax}$ has the flexibility to adapt to the new setting with embeddings. If the choice is a phrase, the decoder will go through "idle run" process and the tags of those source words of the chosen phrase will be set to all-zero.

3.6 Idle Run for Multi-word Phrases

Our decoder is vastly different from that of RNNsearch, but it is still generally built on the basic word-by-word decoding mechanism. To further accommodate the phrase mode in which multiple words are generated all at once, we introduce the "idle run". Basically, if at time t a multiple-word phrase is chosen, the decoding RNN will run exactly the same way as in word mode with regard to state update and attention, only that the generation of words in the rest of phrase is pre-determined at t.

This process is called *idle run*, which can be illustrated through the following example. In Figure 3, if at time $t = 5$, the decoder decides to go with *phrase mode* and generate $\mathbf{p}_1$, the decoder will not really output $\mathbf{p}_1 = (y_5, y_6)$ at once. To keep the updating of $\mathbf{s}_t$ (Equation (1)), the decoder will first output y_5 and use $\mathbf{e}_{y_5}$ and other required elements to update $\mathbf{s}_5$

to $\mathbf{s}_6$, uses $\mathbf{s}_6$ to generates y_6. With $\mathbf{e}_{y_6}$ and other required elements, the decoder will update the state to $\mathbf{s}_7$ and at the time $t = 7$, the decoder starts to make its next decision, *phrase mode* or *word mode* for the coming words. During the output of $\mathbf{p}_1$, the decoder does not need to make decisions or sample words as it is already in one *phrase mode*.

3.7 The Probabilistic Model for Phrases

Given a target phrase $\mathbf{p} = \{y_t, y_{t+1}, y_{t+2}\}$, in principle, its words could be chosen either from vocabulary $\mathcal{V}$ or entirely retrieved from phrase table $\mathcal{P}$. So in general each word is potentially generated from a mixture probability model. In the case that there are out-of-vocabulary words (`UNKs`) in the phrases, which are faily common in practice, the mixture model degenerates to phrase mode only.

For an unified notation, we introduce an indicator variable I_{unk} into the mixture probability model, which is summarized as follows,

$$\begin{aligned} p(y_t, y_{t+1},&y_{t+2}|\mathbf{y}_{<t}, \mathbf{x}; \theta) \\ =&I_{\text{unk}} \times \prod_{i=t}^{t+2} p(z_i = 0, y_i|S_i; \theta) \qquad (3) \\ &+p(z_t = 1, \mathbf{p}_t = \mathbf{p}|S_t; \theta). \end{aligned}$$

where $I_{unk} = 1$ means there is no `UNKs` in the phrase, and 0 otherwise.

For phraseNet$_{gate}$, $p(z_t = 0, y_t|S_t; \theta)$ factorizes into $p(z_t = 0|S_t; \theta)p(y_t|S_t, 0; \theta)$, and $p(z_t = 1, \mathbf{p}_t = \mathbf{p}|S_t; \theta)$ factorizes into $p(z_t = 1|S_t; \theta)p_{\mathbf{p}}(\mathbf{p}_t = \mathbf{p}|S_t, 1; \theta)$. For phraseNet$_{softmax}$, there is no explicit variable z_t, the indicator of mode is implicitly absorbed into the choice of words and phrases. The probability $p(z_t = 0, y_t|S_t; \theta)$ can therefore be re-written as $p(y_t|S_t; \theta)$ and $p(z_t = 1, \mathbf{p}_t = \mathbf{p}|S_t; \theta)$ as $p(\mathbf{p}_t = \mathbf{p}|S_t; \theta)$.

For normal words that are not part of phrases $\mathcal{W}_y$, they can only be generated by *word mode*, which is the same as RNNsearch.

Given a pair of source and target sentence $\mathbf{x} = (x_1, x_2, \ldots, x_{T_x})$ and $\mathbf{y} = (y_1, y_2, \ldots, y_{T_y})$, the probability of this pair of sentences is:

$$p(\mathbf{y}|\mathbf{x}; \theta) = \prod_{y_i \in \mathcal{W}_y} p(y_i|\mathbf{y}_{<i}, \mathbf{x}; \theta) \prod_{p_j \in \mathcal{P}_{xy}} p(\mathbf{p}_j|\mathbf{y}_{<j}, \mathbf{x}; \theta),$$

here $p(\mathbf{p}_j|y_{<j}, \mathbf{x}; \theta)$ refers to the mixture probability (Equation (3)) of output the words in $\mathbf{p}_j$. For a given batch of the source and target sequences $\{X\}_N$ and $\{Y_N\}$, the objective is to minimize the negative log-likelihood:

$$\mathcal{L} = -\frac{1}{N} \sum_{k=1}^{N} p(\mathbf{y}^{(k)}|\mathbf{x}^{(k)}; \theta)$$

4 Experiments

We report our empirical study on applying phraseNet$_{gate}$ and phraseNet$_{softmax}$ to Chinese-to-English translation, and comparing it against RNNsearch and SMT models.

4.1 Phrase table $\mathcal{P}$

As mentioned before, when we design our model, our definition for "phrase" is different from that used in phrase-based statistical machine translation. For each source phrase, our models only support an unique translation (target phrase). Therefore, we only choose those phrase pairs that the source phrase almost always translates to the target phrase. We also hope the contexts for the source phrases are relatively fixed so that the models can learn the patterns of translation easier. With these considerations, we focus our attention on five categories of phrases: dates, names, numbers, locations and organizations. Apart from these five categories, we also collect some other phrases that fulfil our requirements. Figure 7 shows several examples of our Phrase table.

Category	Source phrase	Target phrase
dates	二〇〇三年三月一日	march 1 , 2003
names	董建华	tung chee - hwa
numbers	一千七百万	17 million
locations	珠三角	pearl river delta
organizations	哈佛大学	harvard university
others	叶酸	folic

Figure 7: Examples of phrase for each category.

The phrase pairs are collected from several sources. The first source consists of extracted phrase pairs from a bi-lingual corpus using the method described in (Meng et al., 2014), (Ren et al., 2009) and (Frantzi et al., 2000). The second source is the LDC dictionary. The third source is from proper nouns dictionaries, which contain many commonly used Chinese-to-English translation pairs for proper nouns. We have also generated some Chinese-to-English phrase translations, especially dates, numbers and Chinese names, by predefined rules. There are two formats of Chinese names, Mandarin names and Cantonese names, both of their English counterparts could be generated according to their pronunciation rules. Numbers can also be generated by predefined rules, like "1345 $\rightarrow$ 1,345". Using these rules, we transform several formats of Chinese numbers to English numbers.

4.2 Setup

Our training data contains 1.25M sentence pairs obtained from LDC corpora[1], with 27.9M Chinese words and 34.5M English words respectively. We use NIST 2002 (NIST02) dataset as our development set, and the NIST 2003 (NIST03), NIST 2004 (NIST04), NIST 2005 (NIST05), 2006 (NIST06) and 2008 (NIST08) datasets as our test sets. The case-insensitive 4-gram NIST BLEU score (Papineni et al.2002) is used as our evaluation metric.

In training the neural networks, we limit the source and target vocabularies to the most frequent 16K words (one of the words is reserved for the unknown words (UNK)) in Chinese and English, covering approximately 95.8% and 98.3% of the two corpora respectively. We train each model with the sentences of length up to 50 words in training data. The word embedding dimension is 620 and the size of a hidden layer is 1000. We set n_p as 10. In both the RNNsearch and our models, we adopt the coverage models introduced in (Tu et al., 2016) to mitigate the problem of over-translation.

We compare our models with state-of-the-art SMT and RNNsearch:

1. Moses (Koehn et al.2007): an open source phrase-based translation system with default configuration and a 4-gram language model trained on the target portion of training data;

2. RNNsearch (Bahdanau et al.2015): an attentional NMT model with default setting.[2]

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

[2]We use the code from (`https://github.com/kyunghyuncho/dl4mt-material`) with minor modifications.

4.3 Translation Performance

Table 1 shows the translation performances measured in BLEU score. Clearly both the proposed phraseNet$_{gate}$ and phraseNet$_{softmax}$ significantly improves the translation quality in all cases. More specifically, On average, phraseNet$_{gate}$ yields about 3.45 BLEU score improvement over our baseline, phraseNet$_{softmax}$ yields about 2.13 BLEU score improvement over our baseline. Also, RNNsearch with expanded vocabulary (30K words) is 1.65 BLEU behind phraseNet$_{gate}$. Surprisingly, phraseNet$_{softmax}$ comes behind phraseNet$_{gate}$, despite its potential ability to take the content of the target phrase into the decision. We conjecture that this might be due to the difficulty in directly comparing the scores from two different types of scoring functions in the same pool of softmax.

It is also reasonable to doubt that our models are just generate the target phrases without considering the positions, as this will also (almost surely) increase the 1-gram and 2-gram BLEU scores and hence increase the final BLEU scores. To further verfy this, Table 2 compares our models with RNNsearch measured in 4-gram BLEU score, which capture overlapping of generated targets and reference on longer segments.

Our models, especially phraseNet$_{gate}$, still perform better than RNNsearch, incidating that the phrases are put into the right places.

4.4 Samples of Translation

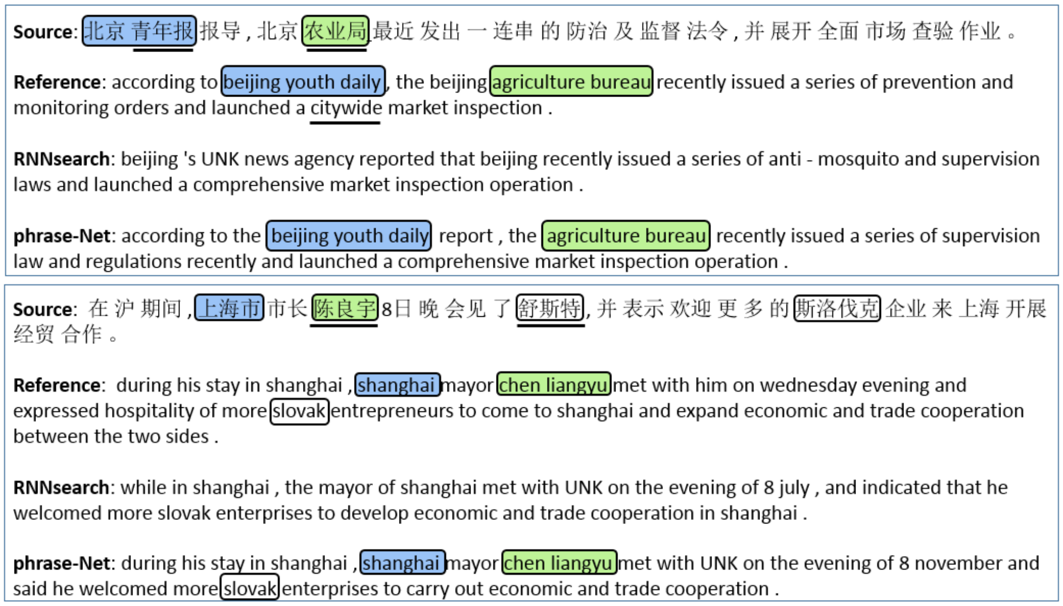
We also give two examples from test set comparing our phraseNet$_{gate}$ with RNNsearch, and more examples can be found in supplementary materials. As demonstrated through those examples, when there are phrases found in the source sentences, phraseNet$_{gate}$ has a better chance to generate the corresponding target phrases correctly at proper locations. This could happen when the source phrases consist words all in the vocabulary, but more frequently when there are UNK words there, showing that phraseNet$_{gate}$ is also a strong model to solve the UNK problem. Another interesting observation, e.g., the second example in Figure 8 is that for some common phrases phraseNet sometimes ignores the

Models	NIST02	NIST03	NIST04	NIST05	NIST06	NIST08	Ave.
Moses	33.41	31.61	33.48	30.75	31.07	23.37	30.06
RNNSearch (16K)	34.96	32.19	33.85	30.79	30.32	22.13	29.86
RNNSearch (30K)	36.04	33.96	35.82	33.05	31.88	23.61	31.66
phraseNet$_{gate}$ (16K)	**37.68**	**36.01**	**37.69**	**34.61**	**32.70**	**25.52**	**33.31**
phraseNet$_{softmax}$ (16K)	36.60	34.07	35.93	33.37	31.96	24.62	31.99

Table 1: Evaluation of translation quality, where we use boldface digits to denote the best performance.

Models	NIST02	NIST03	NIST04	NIST05	NIST06	NIST08	Ave.
RNNSearch (16K)	16.89	15.64	17.77	16.02	15.26	10.16	14.97
phraseNet$_{gate}$ (16K)	18.97	17.95	19.11	17.21	16.14	11.92	16.47
phraseNet$_{softmax}$ (16K)	17.72	16.23	17.94	16.42	15.53	10.98	15.42

Table 2: Evaluation of translation quality in 4-gram BLEU score.



Source: 北京 青年报 报导 , 北京 农业局 最近 发出 一 连串 的 防治 及 监督 法令 , 并 展开 全面 市场 查验 作业 。

Reference: according to beijing youth daily, the beijing agriculture bureau recently issued a series of prevention and monitoring orders and launched a citywide market inspection .

RNNsearch: beijing 's UNK news agency reported that beijing recently issued a series of anti - mosquito and supervision laws and launched a comprehensive market inspection operation .

phrase-Net: according to the beijing youth daily report , the agriculture bureau recently issued a series of supervision law and regulations recently and launched a comprehensive market inspection operation .

Source: 在 沪 期间 , 上海市 市长 陈良宇 8日 晚 会见 了 舒斯特 , 并 表示 欢迎 更 多 的 斯洛伐克 企业 来 上海 开展 经贸 合作 。

Reference: during his stay in shanghai , shanghai mayor chen liangyu met with him on wednesday evening and expressed hospitality of more slovak entrepreneurs to come to shanghai and expand economic and trade cooperation between the two sides .

RNNsearch: while in shanghai , the mayor of shanghai met with UNK on the evening of 8 july , and indicated that he welcomed more slovak enterprises to develop economic and trade cooperation in shanghai .

phrase-Net: during his stay in shanghai , shanghai mayor chen liangyu met with UNK on the evening of 8 november and said he welcomed more slovak enterprises to carry out economic and trade cooperation .

Figure 8: Example of phraseNet$_{gate}$ on test sets compared with RNNsearch. Word segmentation is applied on the input, where underlined are UNK words. The phrases highlighted by boxes (with or without colors) are those phrases in our phrase table. The highlighted phrases without colors are phrases generated by *word mode* or not generated.

suggestion of phrase mode, but still generate the entire phrase correctly from its word mode. This shows phraseNet maintains a healthy and flexible balance between word and phrase mode.

5 Related Work

Probably the work that is closest to phraseNet is the recently proposed Neural Generative QA (genQA) (Yin et al., 2015), where a set of triples are stored in a QA memory, and a neural network queries this memory for words to use in generating the answer. More specifically, phraseNet$_{gate}$ has the same gating strategy as in genQA. Still, phraseNet is different from that in several important ways: 1) phraseNet can handle multiple phrases in one sentence, and 2) phraseNet can generate multi-word expression.

The softmax with multiple modes in phraseNet$_{softmax}$ is very similar to the recently proposed CopyNet (Gu et al.,). But the generative mode in CopyNet still follows a strict word-by-word fashion and therefore a soft-decision between modes has to be made for each mode. In a similar way, phraseNet is related to (Gulcehre et al., 2016a) and (Cheng and Lapata, 2016).

6 Conclusions and Future Work

We propose a neural machine translator which can leverage an external phrase memory, and empirically show its efficacy on Chinese-English translation.

References

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

[Cheng and Lapata2016] Jianpeng Cheng and Mirella Lapata. 2016. Neural summarization by extracting sentences and words. *arXiv preprint arXiv:1603.07252*.

[Cho et al.2014] 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 Empirical Methods in Natural Language Processing (EMNLP)*, pages 1724–1734, Doha, Qatar, October. Association for Computational Linguistics.

[Chung et al.2014] Junyoung Chung, Caglar Gulcehre, KyungHyun Cho, and Yoshua Bengio. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. *arXiv preprint arXiv:1412.3555*.

[Frantzi et al.2000] Katerina Frantzi, Sophia Ananiadou, and Hideki Mima. 2000. Automatic recognition of multi-word terms:. the c-value/nc-value method. *International Journal on Digital Libraries*, 3(2):115–130.

[Gu et al.] Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. Incorporating copying mechanism in sequence-to-sequence learning. In *ACL2016*. Association for Computational Linguistics.

[Gulcehre et al.2016a] Caglar Gulcehre, Sungjin Ahn, Ramesh Nallapati, Bowen Zhou, and Yoshua Bengio. 2016a. Pointing the unknown words. *arXiv preprint arXiv:1603.08148*.

[Gulcehre et al.2016b] Caglar Gulcehre, Marcin Moczulski, Misha Denil, and Yoshua Bengio. 2016b. Noisy activation functions. *arXiv preprint arXiv:1603.00391*.

[He et al.2015] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. 2015. Deep residual learning for image recognition. *arXiv preprint arXiv:1512.03385*.

[Meng et al.2014] Fandong Meng, Deyi Xiong, Wenbin Jiang, and Qun Liu. 2014. Modeling term translation for document-informed machine translation. In *In Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing*, pages 546–556, Doha, Qatar.

[Meng et al.2015] Fandong Meng, Zhengdong Lu, Mingxuan Wang, Hang Li, Wenbin Jiang, and Qun Liu. 2015. Encoding source language with convolutional neural network for machine translation. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 20–30, Beijing, China, July. Association for Computational Linguistics.

[Ren et al.2009] Zhixiang Ren, Yajuan Lü, Jie Cao, Qun Liu, and Yun Huang. 2009. Improving statistical machine translation using domain bilingual multiword expressions. In *Proceedings of the Workshop on Multiword Expressions: Identification, Interpretation, Disambiguation and Applications*, pages 47–54. Association for Computational Linguistics.

[Schuster and Paliwal1997] Mike Schuster and Kuldip K Paliwal. 1997. Bidirectional recurrent neural networks. *Signal Processing, IEEE Transactions on*, 45(11):2673–2681.

[Sennrich et al.2015] Rico Sennrich, Barry Haddow, and Alexandra Birch. 2015. Improving neural machine translation models with monolingual data. *arXiv preprint arXiv:1511.06709*.

[Tu et al.2016] Zhaopeng Tu, Zhengdong Lu, Yang Liu, Xiaohua Liu, and Hang Li. 2016. Modeling coverage for neural machine translation.

[Yin et al.2015] Jun Yin, Xin Jiang, Zhengdong Lu, Lifeng Shang, Hang Li, and Xiaoming Li. 2015. Neural generative question answering. *arXiv preprint arXiv:1512.01337*.