Convolution-Enhanced Bilingual Recursive Neural Network for Bilingual Semantic Modeling

Jinsong Su[1], **Biao Zhang**[1], **Deyi Xiong**[2]*, **Ruochen Li**[1], **Jianmin Yin**[3]
Xiamen University, Xiamen, China 361005[1]
Soochow University, Suzhou, China 215006[2]
Weifang Beida Jade Bird Huaguang Information Technology Co., Ltd, Weifang, China 261205[3]
jssu@xmu.edu.cn, zb@stu.xmu.edu.cn
dyxiong@suda.edu.cn, lrc_n@stu.xmu.edu.cn, jimyin@vip.sina.com

Abstract

Estimating similarities at different levels of linguistic units, such as words, sub-phrases and phrases, is helpful for measuring semantic similarity of an entire bilingual phrase. In this paper, we propose a convolution-enhanced bilingual recursive neural network (ConvBRNN), which not only exploits word alignments to guide the generation of phrase structures but also integrates multiple-level information of the generated phrase structures into bilingual semantic modeling. In order to accurately learn the semantic hierarchy of a bilingual phrase, we develop a recursive neural network to constrain the learned bilingual phrase structures to be consistent with word alignments. Upon the generated source and target phrase structures, we stack a convolutional neural network to integrate vector representations of linguistic units on the structures into bilingual phrase embeddings. After that, we fully incorporate information of different linguistic units into a bilinear semantic similarity model. We introduce two max-margin losses to train the ConvBRNN model: one for the phrase structure inference and the other for the semantic similarity model. Experiments on NIST Chinese-English translation tasks demonstrate the high quality of the generated bilingual phrase structures with respect to word alignments and the effectiveness of learned semantic similarities on machine translation.

1 Introduction

Recently, adapting deep neural networks to statistical machine translation (SMT) is of growing interest due to their superior capacity against conventional lexical models in feature learning and representation (Yang et al., 2013; Liu et al., 2013; Li et al., 2013; Devlin et al., 2014; Liu et al., 2014; Setiawan et al., 2015). As phrases are the basic translation units in many SMT systems, one line of research among these studies is to learn the semantic similarity of bilingual phrases for translation selection in SMT (Zhang et al., 2014a; Gao et al., 2014; Cho et al., 2014; Su et al., 2015; Hu et al., 2015).

Typically, these bilingual semantic similarity models learn source and target phrase representations with some bilingual constraints (Gao et al., 2014; Hu et al., 2015; Zhang et al., 2014a). In spite of their success, they often suffer from two problems. Firstly, it is difficult for them to recover the semantic hierarchy (binary tree structure) of a bilingual phrase. In this respect, Su et al. (2015) improve tree construction by incorporating word alignments into their objective function. Unfortunately, they still employ the recursive autoencoder (RAE) as the underlying model to build tree structures of phrases according to the minimum reconstruction error. As a result, word alignments are not fully exploited for phrase structure generation. Secondly, the previous bilingual semantic similarity models are incapable of leveraging representations at different levels of linguistic units, such as words, sub-phrases and phrases. They usually represent a phrase (a sequence of words) with a single, fixed vector. However, as demonstrated in attention-based neural machine translation (Bahdanau et al., 2014), one vector is not semantically sufficient to encode a sequence of words preserving representations at different levels of linguistic units may be beneficial.

*Corresponding author.

Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers, pages 3071–3081, Osaka, Japan, December 11-17 2016.

To solve these problems, we propose a convolution enhanced bilingual recursive neural network (ConvBRNN), which exploits word alignments to guide the generation of phrase structures and then integrates embeddings of different linguistic units on the phrase structures into bilingual semantic modeling. Specifically, we develop a new recursive neural network, in which the composition criterion for tree construction is the degree of consistency to word alignments rather than the reconstruction error. Furthermore, we propose a variant of the tree-based convolutional neural network (Mou et al., 2015) to fully access all embeddings on the phrase structures, which can be used to produce better phrase representations (see Section 3.2). All these make ConvBRNN more suitable for the subsequent bilingual semantic modeling, where a bilinear model is introduced to interact and compare the source and target phrase representations in terms of the degree of semantic equivalence. To train our model, we introduce two max-margin losses: one for the bilingual semantic structure inference and the other for the semantic similarity model, both of which are derivable.

We conduct experiments on large-scale corpus to examine the effectiveness of ConvBRNN on bilingual phrase structure learning and semantic similarity estimation. Experiment results on NIST MT06 and MT08 datasets show that our system achieves significant improvements over baseline methods. We further analyze the generated bilingual phrase structures and semantic scores, both of which indicate that ConvBRNN indeed learns information from word alignments that is beneficial for bilingual semantic representations.

Our major contributions lie in the following three aspects:

- We develop a new recursive neural network with an alignment-based semantic composition metric to generate word-alignment-consistent bilingual phrase structures.
- we develop a variant of tree-based convolutional neural model, which utilizes all embeddings on a phrase structure rather than the embedding of the entire phrase to model bilingual semantics.
- We carry out a series of experiments and demonstrate that our model is superior to baselines in terms of both the learned phrase structures and semantic similarities.

2 Related Work

A straightforward approach to learning bilingual phrase representations is to adapt monolingual phrase models with bilingual supervisions. For example, Li et al. (2013) encode reordering orientations into RAE-generated embeddings. To utilize the semantic equivalence constraint between source and target phrases, Gao et al. (2014) use a feedforward neural network to model phrase embeddings and try to maximize their semantic similarity, while Zhang et al. (2014a) introduce a bilingually-constrained RAE. Furthermore, Hu et al. (2015) incorporate context information to disambiguate translation selection. Very recently, neural machine translation trains a unified encoder-decoder (Sutskever et al., 2014; Bahdanau et al., 2014) neural network for translation, where an encoder maps the input sentence into a fixed-length vector, and a decoder generates a translation from the encoded vector.

Unlike the work mentioned above, our model mainly explore word alignments to guide the generation of bilingual phrase structures. The most relevant work to ours is the model proposed by Su et al. (2015), where they treat word alignments as a constraint to the RAE model. However, as discussed in Section 1, the composition criterion in RAE (i.e. reconstruction errors) does not allow us to fully benefit from word alignments. Therefore, we introduce a new composition criterion based on word alignment consistency. The proposed recursive neural network works in a way similar to that in (Socher et al., 2011b) except for our specific bilingual supervision. Zhang et al. (2014b) also propose a recursive neural network. However, their model mainly focuses on the composition in machine translation process (namely, *swap* or *monotone*), which is different from ours.

Additionally, our model also adapts convolutional neural network (Kalchbrenner et al., 2014; Kim, 2014) to extract semantic information encoded in phrase structures. Our model is related to the tree-based convolution (Mou et al., 2015). The differences are 1) that we treat the whole tree structure as the window for convolution; and 2) that the underlying phrase structure for a sentence is generated automatically in our model, instead of taking from a given constituency or dependency tree. Besides, the exploration of the semantic embeddings at different levels of granularity is firstly investigated in

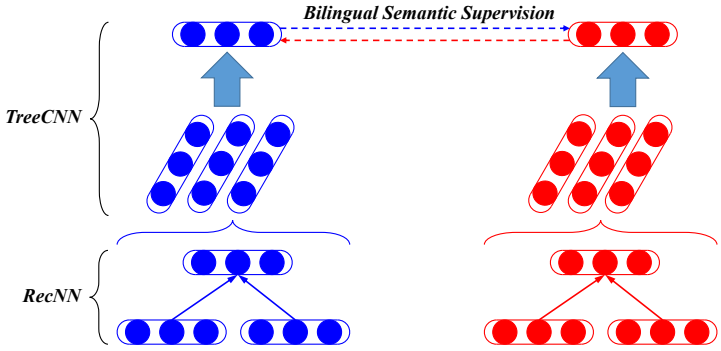

Figure 1: An illustration of the convolution-enhanced bilingual recursive neural network.

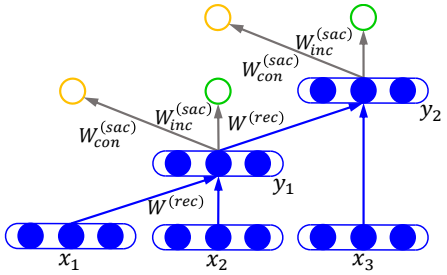

Figure 2: An illustration of the proposed RecNN. We use a yellow/green circle to represent the preference score of a node to be an SAC/non-SAC node.

(Socher et al., 2011a), where they compute an interaction matrix from which discriminative features are dynamically extracted for paraphrase identification. He et al. (2015) and Yin et al. (2015b) further extend this idea to convolutional neural network. Although our method is partially inspired by them, we implement it in a completely different manner.

3 Convolution-Enhanced Bilingual Recursive Neural Network

This section elaborates the proposed ConvBRNN model, of which network structure is shown in Figure 1. We begin with the generation of phrase structures via a recursive neural network. We then elaborate how to perform convolution upon the generated phrase structures. After that, we describe our bilingual semantic similarity model. Finally, we provide a detailed illustration on the training of ConvBRNN.

3.1 Recursive Neural Network for Generating Phrase Structures

To generate phrase structures, the conventional RAE usually composes neighboring nodes based on their reconstruction errors, which we argue are insufficient to model bilingual semantics. In SMT, one important auxiliary for a bilingual phrase is its word alignments, which contain some useful guidance signals for the bilingual structure construction, as discussed in Section 1. To make better use of these signals, we introduce the following recursive neural network (*RecNN*).

As shown in Figure 2, the input to our RecNN is a list of ordered d-dimensional vectors x=(x_1, x_2, x_3), each of which can be retrieved from a word embedding matrix $\mathbb{L} \in \mathbb{R}^{d \times |V|}$ via its corresponding word index. Here $|V|$ is the size of the vocabulary. Given two neighboring children c_1 and c_2, we compose them into a parent node n (For example, in Figure 2, if we set c_1=x_1 and c_2=x_2, then n=y_1) and produce its semantic vector p_n through a non-linear transformation:

$$p_n = f(W^{(rec)}[c_1; c_2] + b^{(rec)}) \tag{1}$$

where $[c_1; c_2] \in \mathbb{R}^{2d}$ is the concatenation of c_1 and c_2, $W^{(rec)} \in \mathbb{R}^{d \times 2d}$ and $b^{(rec)} \in \mathbb{R}^d$ is the parameter matrix and bias term respectively, and $f(\cdot)$ is an element-wise activation function such as $\tanh(\cdot)$, which is used throughout our experiments. As discussed in Section 1, the previous RAE-style models (Zhang

et al., 2014a; Su et al., 2015) adopt reconstruction error to measure how well p_n represents its children c_1 and c_2, which, however, is not a good solution to directly fully exploit word alignments for bilingual semantic modeling. To fully exploit different levels of bilingual semantic constraints within phrase pairs, we design a new semantic composition metric based on word alignments. As word alignments are shared across the source and target language, they are suitable to act as a desirable bridge for modeling bilingual semantics.

To achieve this goal, we first use the *structural alignment consistency* (SAC) (Su et al., 2015) that is the basis of our model to classify resultant nodes of semantic compositions into two categories. Specifically, if the node n covers a sub-phrase, and there exists a sub-phrase in the other language such that these two sub-phrases are consistent with word alignments (Och and Ney, 2003), we say n satisfies the structural alignment consistency, and it is referred to as an SAC node, otherwise, it is a non-SAC node.

Then, we introduce two functions $Score_{con}(n)$ and $Score_{inc}(n)$ to measure the preference strength of node n to be an SAC or a non-SAC node, respectively

$$Score_{con}(n) = W_{con}^{(sac)}\, p_n, \quad Score_{inc}(n) = W_{inc}^{(sac)}\, p_n \tag{2}$$

where $W_{con}^{(sac)} \in \mathbb{R}^{1\times d}$ and $W_{inc}^{(sac)} \in \mathbb{R}^{1\times d}$ are parameter matrices. Furthermore, we calculate the final semantic composition score of node n as follows

$$Score_{sc}(n) = \frac{exp(Score_{con}(n))}{exp(Score_{inc}(n))} \tag{3}$$

Obviously, the larger $Score_{con}(n)$ is than $Score_{inc}(n)$, the larger $Score_{sc}(n)$ should be.

We traverse each possible semantic composition of neighboring children and calculate its semantic composition score, and finally select the composition with the largest score. This combination process on neighboring children repeats at each node until the structure and embedding of the entire bilingual phrase are generated. To obtain the optimal binary tree and phrase representation for x, we minimize the following objective function formulated as follows:

$$\begin{aligned} E_{align}(x) = & \sum_{n\in T_{con}(x)} max\{0, 1 - Score_{con}(n) + Score_{inc}(n)\} \\ & + \sum_{n\in T_{inc}(x)} max\{0, 1 - Score_{inc}(n) + Score_{con}(n)\} \end{aligned} \tag{4}$$

where $T_{con}(x)$ and $T_{inc}(x)$ denote the SAC and non-SAC node sets in the binary tree of x, respectively. It should be noted especially that we use different max-margin loss functions for different types of nodes. On the one hand, we simultaneously maximize the $Score_{con}(*)$ and minimize the $Score_{inc}(*)$ of SAC nodes. On the other hand, we take an opposite approach to deal with non-SAC nodes. In this way, the node type (SAC/non-SAC) with word alignment information performs as a guidance signal to encourage the generation of word-alignment-consistent phrase structures.

3.2 Convolutional Neural Network for Learning Phrase Representations

Given a generated phrase structure, a straightforward way to obtain phrase representation is to extract the embedding of the root node of the phrase structure, as implemented in the conventional RAE. However, a major limitation of this method is the neglect of lower-level linguistic units, e.g. words and sub-phrases. To alleviate this problem, we stack a variant of tree-based convolutional neural network (*TreeCNN*) to incorporate all the embeddings inside the phrase structure.

Upon the generated structure $T(x)$ of an input phrase x, we first perform postorder traversal to extract embeddings of all nodes, and then concatenate them column-wisely into a matrix $M \in \mathbb{R}^{d\times|n|}$, where $|n|$ is the number of nodes in $T(x)$. Note that the node number varies with different phrases. In this way, the representations at different levels are interlaced along the rows of M, which facilitates the upcoming window-based convolution. To construct our TreeCNN, we take the matrix M as the input layer. Figure

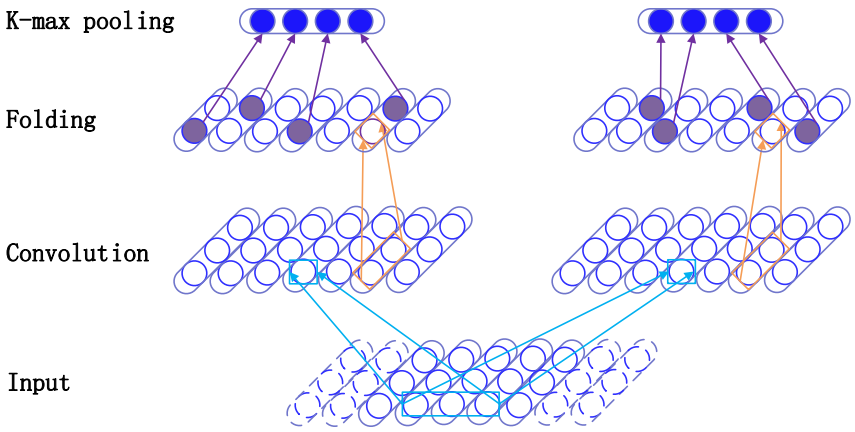

Figure 3: Illustration of the proposed TreeCNN with window size 3, pooling size 2 and filter number 2. We use a light blue, orange and purple color to indicate the convolution, folding and pooling operation, respectively. The nodes with dashed circles represent zero-padded embeddings for wide convolution.

3 shows the architecture of our TreeCNN, which consists of three different layers: *convolution*, *folding* and *k-max pooling*.

Convolution Layer This layer iteratively convolves an h-sized sliding window on M, and uses a filter F to summarize the information inside the window. Since the length of phrases in the translation model is usually not long, we pad the matrix M with h-1 zero embeddings on both sides and adopt the wide convolution (Kalchbrenner et al., 2014) (see the dashed circles in Figure 3). To discover semantic information at a finer granularity, we further construct *per-dimension filters* $F^{[r]}$ ($1 \leq r \leq d$) (He et al., 2015) to convolve the embeddings in the r-th row of M.

Formally, applying the per-dimension filter $F^{[r]}$ on M produces an output vector $C^{[r]} \in \mathbb{R}^{|n|+h-1}$ where the i-th entry ($1 \leq i \leq |n| + h - 1$) is computed as follows:

$$C_i^{[r]} = (W_{F^{[r]}})^T M_{i:i+h-1}^{[r]} \tag{5}$$

where $W_{F^{[r]}} \in \mathbb{R}^h$ is the parameter vector of $F^{[r]}$. This procedure is illustrated in Figure 3 with a light blue color. By applying all per-dimension filters to traverse all windows of matrix M, we can obtain a feature map $C \in \mathbb{R}^{d \times (|n|+h-1)}$. It encodes complex dependencies across different levels of linguistic units and contains linguistic properties implied in each dimension, which, nevertheless, makes different dimensions independent of each other. Next we will introduce a folding layer to exploit these dimensions simultaneously.

Folding Layer This layer bridges the gap across different dimensions through averaging each nonoverlapping neighboring rows in the convoluted feature map C. Specifically, for each row index r ($1 \leq r \leq \lfloor \frac{d}{2} \rfloor$), the output can be computed as follows (shown in the orange color in Figure 3):

$$A^{[r]} = (C^{[2r-1]} + C^{[2r]})/2 \tag{6}$$

where $A^{[r]} \in \mathbb{R}^{|n|+h-1}$ is the r-th row of $A \in \mathbb{R}^{\lceil \frac{d}{2} \rceil \times (|n|+h-1)}$. Different from previous work, we allow dimension size d to be odd. In this case, we simply append the last row of C onto A.

After the above operation, each element of A captures complex dependencies across both rows and columns of M. To mingle these dependencies, we further perform a non-linear transformation following Yin et al. (2015a):

$$U = f(A + b_{[:]}) \tag{7}$$

where $b \in \mathbb{R}^{\lceil \frac{d}{2} \rceil}$ is the bias term that is shared across different columns, and the subscript $[:]$ indicates a column-wise *broadcasting* operation. It should be noted that the column dimension of U (i.e. $|n|+h-1$) differs for different phrases. This raises a key problem: how can we transform the variable-length matrix

U into a fixed-length vector. In order to deal with this problem, we further stack a K-max Pooling layer (Kalchbrenner et al., 2014).

K-max Pooling Layer This layer extracts the top-k values over each row of U so as to: 1) preserve rich semantic information of a sentence; and 2) eliminate the variance in the column dimension of U (shown in the purple color in Figure 3). In doing so, we obtain a phrase vector representation with the dimension size $\lceil \frac{d}{2} \rceil \cdot k$. Notice that k in this layer is predefined. Although we can use the dynamic version of k-max pooling to stack more convolution, folding and pooling layers (Kalchbrenner et al., 2014), we do not take this strategy due to the trade-off between performance and cost. Theoretically, more layers should capture much deeper semantic information. We leave this for our future research.

So far we have described how we apply the wide convolution, folding layer and k-max pooling layer onto an input phrase matrix to obtain a fixed-length phrase representation. Inspired by studies on convolutional networks for object recognition, we introduce L filters to produce multiple feature maps, which are used to capture semantics of input phrases. Finally, we concatenate the vector representations derived from L filters to obtain the final phrase representation $p \in \mathbb{R}^{\lceil \frac{d}{2} \rceil \cdot k \cdot L}$.

3.3 Bilingual Semantic Supervision

Through the above procedures, we obtain the semantic representations of bilingual phrase (f, e), denoted by p_f and p_e. To measure the semantic similarity of f and e, we introduce two transformation matrixes $W_f^{(sem)} \in \mathbb{R}^{d_{sem} \times (\lceil \frac{d_s}{2} \rceil \cdot k \cdot L)}$ and $W_e^{(sem)} \in \mathbb{R}^{d_{sem} \times (\lceil \frac{d_t}{2} \rceil \cdot k \cdot L)}$ to project their semantic representations p_f and p_e into a common semantic space:

$$p'_f = f(W_f^{(sem)} p_f + b^{(sem)}), \;\; p'_e = f(W_e^{(sem)} p_e + b^{(sem)}) \tag{8}$$

where p'_f and p'_e are transformed representations of f and e, d_s/d_t is the dimension size of phrase representation in the source/target semantic space, d_{sem} is the that of the common semantic space. Although we distinguish the transformation matrices for the source and target language, we share the same bias term $b^{(sem)}$ for both languages. The advantage of this is that our model will learn to encode bilingual semantics into these transformation matrices, rather than biases.

Then, we further stack a bilinear model over the transformed representations to compute the semantic similarity score $Sim(f, e)$:

$$Sim(f, e) = {p'_f}^T W_{bi}^{(sem)} p'_e \tag{9}$$

where $W_{bi}^{(sem)} \in \mathbb{R}^{d_{sem} \times d_{sem}}$ is a squared matrix of parameters to be learned. Intuitively, each element in $W_{bi}^{(sem)}$ represents an interaction between p'_f and p'_e, which is used to capture the semantic correspondence within f and e.

To make the semantic scores of translation equivalents as large as possible while scores of non-translation pairs as small as possible, we introduce the following max-margin loss for (f, e):

$$\begin{aligned} E_{sem}(f, e) = & \; max\{0, 1 - Sim(f, e) + Sim(f, e^-)\} \\ & + max\{0, 1 - Sim(f, e) + Sim(f^-, e)\} \end{aligned} \tag{10}$$

where f^-/e^- is a bad translation that replaces the words in f/e with randomly chosen source/target language words.

3.4 Model Training

As described above, there are two types of errors involved for the phrase pair (f, e): (1) structural alignment error $E_{align}(f, e)$ that estimates how well the generated structures of f and e comply with word alignments, and (2) semantic error $E_{sem}(f, e)$ that measures how well the learned phrase embeddings of f and e are semantically equivalent.

Given a training corpus $D = \{(f, e)\}$, the final objective of ConvBRNN is formulated as follows:

$$J_{ConvBRNN}(\theta) = \frac{1}{|D|} \sum_{(f,e) \in D} \{\alpha E_{align}(f, e) + (1 - \alpha) E_{sem}(f, e)\} + R(\theta) \tag{11}$$

where $E_{align}(f,e)$ is the sum of $E_{align}(f)$ and $E_{align}(e)$, the hyper-parameter α is used to balance the effects of $E_{align}(f,e)$ and $E_{sem}(f,e)$, and $R(\theta)$ is a regularization term.

Parameters θ are divided into four sets[1]: (1) θ_L: the word embedding matrix (Section 3.1); (2) θ_{RT}: the structure parameters of RecNN (Section 3.1) and TreeCNN (Section 3.2); (3) θ_{wa}: the parameters for structural alignment consistency (Section 3.1); (4) θ_{sem}: the parameters for semantic similarity (Section 3.3). Following previous work (Zhang et al., 2014a; Su et al., 2015), we assign each parameter set a unique weight for regularization:

$$R(\theta) = \frac{\lambda_L}{2}\|\theta_L\|^2 + \frac{\lambda_{RT}}{2}\|\theta_{RT}\|^2 + \frac{\lambda_{wa}}{2}\|\theta_{wa}\|^2 + \frac{\lambda_{sem}}{2}\|\theta_{sem}\|^2 \quad (12)$$

We apply L-BFGS to tune parameters based on gradients over the joint error, as implemented in (Socher et al., 2011c). Word vector embeddings θ_L are initialized with the toolkit Word2Vec[2] on a large scale unlabeled data. Other parameters are randomly initialized according to a normal distribution (μ = 0,σ = 0.01). With the trained model parameters, we can easily obtain the dense semantic vectors for bilingual phrases. During translation, we incorporate the derived phrasal similarity feature into the standard log-linear framework (Och and Ney, 2002) of SMT for translation selection.

4 Experiment

We conducted experiments on NIST Chinese-English translation task to validate the effectiveness of ConvBRNN.

System Overview Our baseline decoder is a state-of-the-art phrase-based translation system equipped with a maximum entropy based reordering model, which adopts three bracketing transduction grammar rules (Wu, 1997; Xiong et al., 2006). We compared the proposed model with two models: (1) the bilingual correspondence model (*BCorrRAE*) proposed by Su et al. (2015); (2) the proposed model without the convolutional neural network (*ConvBRNN-CNN*), which simply treats the embedding of root node of the phrase structure as the semantic representation of the whole phrase, instead of the convoluted one. Other components of ConvBRNN-CNN are the same as those in the ConvBRNN model.

All translation systems used the log-linear framework. The adopted sub-models include: (1) rule translation probabilities in two directions, (2) lexical weights in two directions, (3) targets-side word number, (4) phrase number, (5) language model score, (6) the score of maximal entropy based reordering model, (7) the semantic similarities of phrase pairs. We performed minimum error rate training to tune the optimal feature weights on the development set (Och and Ney, 2003).

Experiment Setup Our training corpus contains 1.0M sentence pairs (25.2M Chinese words and 29M English words) that are from the FBIS corpus and Handsards part of LDC2004T07 corpus. We ran GIZA++[3] on the training data in two directions and applied the "*grow-diag-final-and*" heuristic rule to obtain word alignments. We trained a 5-gram language model on the Xinhua portion of the GIGAWORD corpus using *SRILM* Toolkit[4] with modified Kneser-Ney Smoothing. We chose the 2005 NIST MT evaluation test data as the development set, and the 2006, 2008 NIST MT evaluation test data as the test sets. We used case-insensitive BLEU-4 metric (Papineni et al., 2002) to evaluate translation quality, and conducted paired bootstrap sampling (Koehn, 2004) for significance test.

Network Training To train ConvBRNN, we applied forced decoding (Wuebker et al., 2010) on the training corpus to extract high-quality bilingual phrases for model training. We tuned the optimal hyper-parameters via *random search* method (Bergstra and Bengio, 2012) to minimize the joint error on a small portion of our training data. Finally, we set $d_s = d_t = d_{sem} = 50$, $h = 5$, $L = 10$, $k = 3$, $\alpha = 0.116$, $\lambda_L = 2.14e^{-7}$, $\lambda_{RT} = 2.43e^{-5}$, $\lambda_{wa} = 7.33e^{-5}$ and $\lambda_{sem} = 4.03e^{-6}$, the L-BFGS iteration number N_{iter}=100. To train BCorrRAE, we used the same training data and method for hyper-parameter optimization.

[1]Note that the source and target languages have different four sets of parameters.

[2]https://code.google.com/p/word2vec/

[3]http://www.statmt.org/moses/giza/GIZA++.html

[4]http://www.speech.sri.com/projects/srilm/download.html

Method	MT06	MT08	AVG
Baseline	29.66	21.52	25.59
BCorrRAE	30.94	23.33	27.14
ConvBRNN-CNN	31.16^{+}	23.39^{+}	27.28
ConvBRNN	$\mathbf{31.48}^{+*}$	$\mathbf{23.89}^{+*}$	**27.69**

Table 1: Experiment results on the MT 06/08 test sets, where we highlight the best result in bold. **AVG** = average BLEU scores on test sets; "+": significantly better than *Baseline* ($p < 0.01$); "$*$": significantly better than *BCorrRAE* ($p < 0.05$);

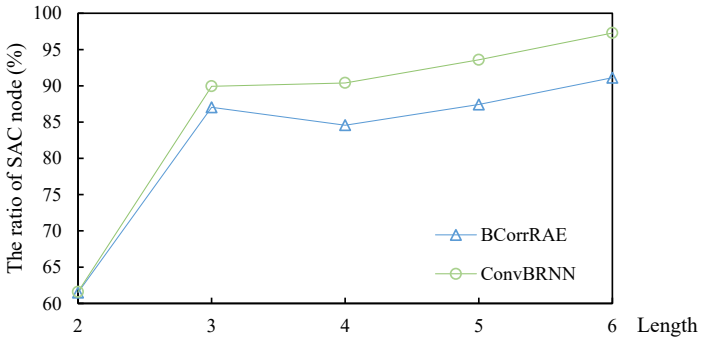

Figure 4: The ratio of *SAC* node specific to the length of covered source phrase. We limit the maximal length of source and target phrase to be 7, and the results when length is 1 and 7 are not shown because they are the same for both models.

4.1 Translation Results

The first experiment checks whether the learned bilingual semantic similarity is able to improve the translation quality. Table 1 summarizes the detailed results. We can observe that our ConvBRNN model significantly improves translation quality in terms of BLEU score on all test sets. Overall, ConvBRNN obtains a gain of up to *2.1* BLEU points on average over the Baseline. Particularly, on the MT08 data set, the improvements over the Baseline can be up to *2.37* BLEU points.

The integration of bilingual correspondence helps BCorrRAE gain *1.55* BLEU points on average over the Baseline. With the recursive neural network for phrase structure generation, ConvBRNN-CNN performs slightly better than BCorrRAE. By integrating the tree-based convolution network for phrase representation learning, our ConvBRNN achieves further improvements over BCorrRAE, which is significant at $p<0.05$. For this result, the reasons may be the following two points: 1) bilingual phrase structures generated by ConvBRNN are more close to the actual semantic structures of phrases; 2) the ConvBRNN model encodes different levels of linguistic units inside phrase structures into final phrase representations. These two points are not adequately considered in BCorrRAE.

4.2 Result Analyses

In order to know how the ConvBRNN model improves the performance of the SMT system, we study the bilingual phrases of our model from the following two respects:

First, we investigate the ability of our model in generating word-alignment-consistent bilingual phrase structures. For this, we extracted phrase pairs from our translation model filtered by NIST test sets and computed the percentage of SAC nodes (Section 3.1) specific to the length of covered source phrase. Following the previous work (Su et al., 2015), we define this percentage as the ratio of the number of SAC nodes to that of all nodes.

Figure 4 reports the ratio values. The ConvBRNN model consistently outperforms the BCorrRAE model. Additionally, as the length grows, the ratio gap between two models becomes larger, with a gain of up to absolute 6%. This indicates that word alignments are more efficiently exploited by our

Source Phrase	BCorrRAE	ConvBRNN
wǒ rènwéi zhè zhǒng	((((*i think*) *this*) *is*) *a*) (((*i think*) *that*) *was*) ((*i regard*) *this*)	((*i think*) ((*this type*) *of*)) ((*i think*) ((*this kind*) *of*)) ((*i find*) ((*that kind*) *of*))
biǎoshì qiángliè bùmǎn	(*strong* ((*opposition against*) *the*)) ((((*expressed strong*) *opposition*) *to*) *the*) (*voice* (*my* (*strong* (*opposition against*))))	((*strongly* (*dissatisfied with*)) *the*) (((*voice* (*my* (*strong opposition*))) *against*) *the*) ((*express* (*strong dissatisfaction*)) *at*)
jiānjué zh̄ichí zhèngfǔ	((*resolutely* (*support the*)) *government*) ((*firm* (*supporter* (*of our*))) *government*) ((*staunchly* (*support* (*the Chinese*))) *government*)	((*staunchly support*) ((*the Chinese*) *government*)) ((*resolutely support*) (*the government*)) ((*firm supporter*) (*of* (*our government*)))

Table 2: Semantically similar target phrases in the training set for example source phrases. The brackets indicate the learned binary tree structure.

ConvBRNN model to generate word-alignment-consistent bilingual phrase structures.

Second, we study whether ConvBRNN can extract meaningful information for semantic similarity from the learned phrase structures. We show some source phrases in Table 2 with their most semantically similar translations learned by BCorrRAE and ConvBRNN in the training corpus. We find that both models are able to distinguish semantic equivalents from non-translation pairs. However, in contrast to BCorrRAE, ConvBRNN prefers diverse expressions. For example, "zhǒng" can be translated into "*type*" or "*kind*", and "bùmǎn" also has two candidate translations "*opposition*" and "*dissatisfaction*". Therefore, during translation, the decoder has many candidate translations for the same source phrase, which we argue is one of the reasons for our success.

We also provide phrase structures in Table 2. We observe that the semantic compositions in BCorrRAE are relatively meaningless because they often do not respect the linguistic phenomena. For example, BCorrRAE prefers branching structures in the same composition direction, such as "(*voice* (*my* (*strong* (*opposition against*))))". Besides, BCorrRAE is more likely to produce undesirable nodes covering high-frequency sub-phrases. For instance, the target phrase "*firm supporter of our government*" has different structures learned by BCorrRAE and ConvBRNN: "((*firm* (*supporter* (*of our*))) *governmen-t*))" (BCorrRAE) and "((*firm supporter*) (*of* (*our government*)))" (ConvBRNN). Obviously, the phrase structure learned by ConvBRNN is more syntactically meaningful. This again demonstrates the advantage of ConvBRAE over BCorrRAE in exploiting word alignments for learning better bilingual phrase structures.

5 Conclusion and Future Work

In this paper, we have presented a convolution-enhanced bilingual recursive neural network to learn bilingual semantic similarity. We first introduce a recursive neural network which directly exploits word alignments to generate word-alignment-consistent bilingual phrase structures. Based on these structures, we further employ a variant of tree-based convolutional neural network to produce bilingual phrase embeddings by summarizing embeddings at different levels of lingual units. Experiment results and analyses on machine translation demonstrate the effectiveness of our model.

In the future, we would like to explore more different selection functions in Eq. (3) for our model due to its importance for the generation of bilingual phrase structures. Besides, as discussed in Section 3.2, we will further enhance the proposed model by trying more effective components, such as dynamic version of k-max pooling, multi-layer convolutions.

Acknowledgements

The authors were supported by National Natural Science Foundation of China (Grant Nos. 61303082, 61403269 and 61672440), Natural Science Foundation of Fujian Province (Grant No. 2016J05161), Natural Science Foundation of Jiangsu Province (Grant No. BK20140355) and CCF Opening Project of Chinese Information Processing (Grant No. CCF2015-01-01). We also thank the anonymous reviewers for their insightful comments.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2014. Neural machine translation by jointly learning to align and translate. In *Proc. of ICLR*.

James Bergstra and Yoshua Bengio. 2012. Random search for hyper-parameter optimization. *Journal of Machine Learning Research*, pages 281–305.

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 *Proc. of EMNLP*, pages 1724–1734.

Jacob Devlin, Rabih Zbib, Zhongqiang Huang, Thomas Lamar, Richard Schwartz, and John Makhoul. 2014. Fast and robust neural network joint models for statistical machine translation. In *Proc. of ACL*, pages 1370–1380.

Jianfeng Gao, Xiaodong He, Wen-tau Yih, and Li Deng. 2014. Learning continuous phrase representations for translation modeling. In *Proc. of ACL*, pages 699–709.

Hua He, Kevin Gimpel, and Jimmy Lin. 2015. Multi-perspective sentence similarity modeling with convolutional neural networks. In *Proc. of EMNLP*, pages 1576–1586.

Baotian Hu, Zhaopeng Tu, Zhengdong Lu, Hang Li, and Qingcai Chen. 2015. Context-dependent translation selection using convolutional neural network. In *Proc. of ACL*, pages 536–541.

Nal Kalchbrenner, Edward Grefenstette, and Phil Blunsom. 2014. A convolutional neural network for modelling sentences. In *Proc. of ACL*, pages 655–665.

Yoon Kim. 2014. Convolutional neural networks for sentence classification. In *Proc. of EMNLP*, pages 1746–1751.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proc. of EMNLP*, pages 388–395.

Peng Li, Yang Liu, and Maosong Sun. 2013. Recursive autoencoders for ITG-based translation. In *Proc. of EMNLP*, pages 567–577.

Lemao Liu, Taro Watanabe, Eiichiro Sumita, and Tiejun Zhao. 2013. Additive neural networks for statistical machine translation. In *Proc. of ACL*, pages 791–801.

Shujie Liu, Nan Yang, Mu Li, and Ming Zhou. 2014. A recursive recurrent neural network for statistical machine translation. In *Proc. of ACL*, pages 1491–1500.

Lili Mou, Hao Peng, Ge Li, Yan Xu, Lu Zhang, and Zhi Jin. 2015. Discriminative neural sentence modeling by tree-based convolution. In *Proc. of EMNLP*, pages 2315–2325.

Franz Josef Och and Hermann Ney. 2002. Discriminative training and maximum entropy models for statistical machine translation. In *Proc. of ACL*, pages 295–302.

Franz Josef Och and Hermann Ney. 2003. A systematic comparison of various statistical alignment models. *Computational Linguistics*, pages 19–51.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proc. of ACL*, pages 311–318.

Hendra Setiawan, Zhongqiang Huang, Jacob Devlin, Thomas Lamar, Rabih Zbib, Richard Schwartz, and John Makhoul. 2015. Statistical machine translation features with multitask tensor networks. In *Proc. of ACL-IJCNLP*, pages 31–41.

Richard Socher, Eric H. Huang, Jeffrey Pennin, Christopher D Manning, and Andrew Y. Ng. 2011a. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. In *Proc. of NIPS*, pages 801–809.

Richard Socher, Cliff C. Lin, Andrew Y. Ng, and Christopher D. Manning. 2011b. Parsing Natural Scenes and Natural Language with Recursive Neural Networks. In *Proc. of ICML*.

Richard Socher, Jeffrey Pennington, Eric H. Huang, Andrew Y. Ng, and Christopher D. Manning. 2011c. Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proc. of EMNLP*, pages 151–161.

Jinsong Su, Deyi Xiong, Biao Zhang, Yang Liu, Junfeng Yao, and Min Zhang. 2015. Bilingual correspondence recursive autoencoder for statistical machine translation. In *Proc. of EMNLP*, pages 1248–1258.

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

Dekai Wu. 1997. Stochastic inversion transduction grammars and bilingual parsing of parallel corpora. *Computational Linguistics, Volume 23, Number 3, September 1997*.

Joern Wuebker, Arne Mauser, and Hermann Ney. 2010. Training phrase translation models with leaving-one-out. In *Proc. of ACL*, pages 475–484.

Deyi Xiong, Qun Liu, and Shouxun Lin. 2006. Maximum entropy based phrase reordering model for statistical machine translation. In *Proc. of ACL*, pages 521–528.

Nan Yang, Shujie Liu, Mu Li, Ming Zhou, and Nenghai Yu. 2013. Word alignment modeling with context dependent deep neural network. In *Proc. of ACL*, pages 166–175.

Wenpeng Yin and Hinrich Schütze. 2015a. Convolutional neural network for paraphrase identification. In *Proc. of NAACL-HLT*, pages 901–911.

Wenpeng Yin and Hinrich Schütze. 2015b. Multigrancnn: An architecture for general matching of text chunks on multiple levels of granularity. In *Proc. of ACL-IJCNLP*, pages 63–73.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014a. Bilingually-constrained phrase embeddings for machine translation. In *Proc. of ACL*, pages 111–121.

Jiajun Zhang, Shujie Liu, Mu Li, Ming Zhou, and Chengqing Zong. 2014b. Mind the gap: Machine translation by minimizing the semantic gap in embedding space. In *Proc. of AAAI*, pages 1657–1664.