Learning Bilingual Phrase Representations with Recurrent Neural Networks

Hideya Mino hideya.mino@nict.go.jp
Andrew Finch andrew.finch@nict.go.jp
Eiichiro Sumita eiichiro.sumita@nict.go.jp
National Institute of Information and Communications Technology, 3-5 Hikaridai, Seika-cho, Soraku-gun, Kyoto 619-0289, JAPAN

Abstract

We introduce a novel method for bilingual phrase representation with Recurrent Neural Networks (RNNs), which transforms a sequence of word feature vectors into a fixed-length phrase vector across two languages. Our method measures the difference between the vectors of source- and target-side phrases, and can be used to predict the semantic equivalence of source and target word sequences in the phrasal translation units used in phrase-based statistical machine translation. Our experiments show that the proposed method is effective in a bilingual phrasal semantic equivalence determination task and a machine translation task.

1 Introduction

In recent years, continuous vector representations of word, phrase, and sentence which alleviate issues of sparsity have successfully been used in a number of natural language processing tasks. Language models with continuous word representations (Bengio et al., 2003; Mikolov et al., 2010; Mikolov, 2012) based on neural networks have outperformed the previous state-of-the-art approaches. These language models map each word to a dense, low-dimensional, real-valued vector, and estimate the probability of words in a continuous space. Representations for phrases have been used in the context of Statistical Machine Translation (SMT). Zou et al. (2013) used phrasal representations for computing the distance between phrase pairs and added a feature based on this distance into the log-linear model of a phrase-based SMT system (Koehn et al., 2003). Their method learned bilingual word representations, and subsequently obtained the phrase-level representations by simply averaging word vectors. Continuous representations for phrases or sentences with neural networks – such as a feed-forward, recursive, or recurrent neural networks – have also been used in SMT. A phrase representation model using a feed-forward neural network for phrase-based SMT was proposed by Schwenk (2007, 2012), and achieved significant BLEU score improvements. Since the model directly projects feature vectors not from words but from phrases or sentences onto a continuous vector space, the representations can contain more global semantic information.

In this paper, we propose a new method to learn bilingual phrase representations for phrase-based SMT using two Recurrent Neural Networks (RNNs) (source- and target-side) combined in a simple linear architecture. We follow the idea of Cho et al. (2014) that the last hidden state of the RNN is a summary representation of the whole input phrase, and the summary representations with the same meaning are trained to be the same vector representation. The procedure is similar to that used in the RNN of Cho et al. (2014), which predicts the next word

in the sequence with a conditional probability. In contrast to this, our model uses an objective with a similarity distance instead of a conditional probability, and learns to minimize the error distance. Furthermore, we developed a novel extension of the model that uses an autoencoder, which is an architecture trained to provide a latent representation of its input by means of a nonlinear encoder and an associated decoder. The objective involves three kinds of errors: a next symbol error for predicting the next word in a phrase, a semantic error for the comparison of the summary phrase representations, and a reconstruction error for the autoencoder. The prediction error represents how well the intermediate hidden states can predict the next word in a sequence. The semantic error represents the dissimilarity between the final hidden states on the source- and target-side. The reconstruction error represents how well the hidden states represent the words in a phrase.

We introduce a bilingual phrase similarity feature derived from our proposed method as a new feature into the log-linear model of a phrase-based SMT system applied to a English-Japanese translation task, and confirm the effectiveness of our method on this task. The results of the experiments show that our model is able to indicate the effective phrase pairs for machine translation. In this paper, though our model is symmetric and does not differentiate between source- and target-side, we use the following notation: the left-side RNN is referred to as the source-side and the right-side RNN (we use overbars on the symbols to differentiate it) is referred to as the target-side.

2 Related work

In this section, we review recent work on neural network phrase representation models.

Continuous phrase representation models with a feed-forward neural network were studied in Schwenk (2007, 2012). The models estimated translation probabilities for unseen phrases with a continuous vector space of phrases. Le et al. (2012) proposed a similar approach to score phrase pairs using fixed-size inputs and outputs. Devlin et al. (2014) proposed a neural network joint model (NNJM) as an extension of the NNLM (Bengio et al., 2003). The NNJM calculates the target-side word probability by using a target-side language model in combination with a context from the source-side. The NNJM requires a maximum length for the source-side phrases. These approaches employ feed-forward neural networks and are constrained to operate on phrases of limited length.

The use of recursive neural networks addresses the fixed-size issue by using a tree structure of phrases and sentences. The recursive neural network maps features from subsequences of a phrase to a continuous vector on each node of the tree recursively. Li et al. (2013) described an ITG reordering classifier which predicted phrase reorderings in SMT that was able to exploit syntactic and semantic information, Zhang et al. (2014, 2015) proposed bilingually-constrained recursive autoencoders, which generated phrasal embeddings for machine translation by learning to minimize the semantic distance between translation equivalents, and maximizing the distance between non-translation pairs.

In contrast to the work on recursive networks, it is also possible to create continuous phrase representations with RNNs. Here, simpler models are possible that do not need take the tree structure of their input into account. Kalchbrenner and Blunsom (2013) proposed recurrent continuous translation models based on recurrent language models (RLMs), which predict target words from an unbounded history of both source and target words with a conditional probability. In their implementation convolutional neural networks were used to model the source-side. Cho et al. (2014) proposed a gated recurrent unit which adaptively remembers and forgets its state based on the input signal to the unit. This model was used to score each phrase pair in the phrase table for SMT.

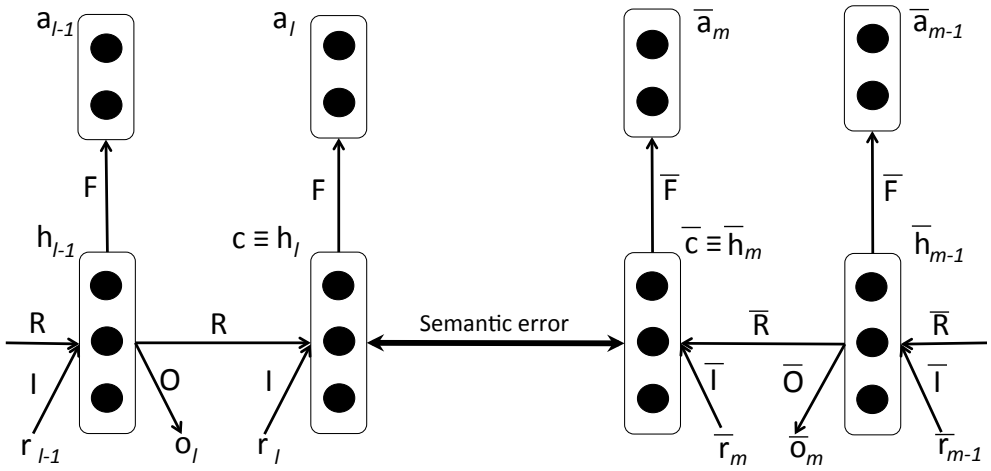

Figure 1: Bilingual Phrase Representation Model Architecture

3 Bilingual Phrase Representation Model

3.1 Phrase Representations

Our bilingual phrase representation model comprises two RNNs: one for source phrases, and the other for target phrases. Each RNN reads a sequence of word representations, and transforms it into a fixed-length vector that holds the semantic content of the whole input sequence. We call this vector the phrasal representation. Then, the model identifies phrase pairs with the same meaning on both source- and target-side, by computing the similarity distance between the respective source and target side phrase representations.

Figure 1 shows the framework of the bilingual phrase representation model, where $\mathbf{r}_k(0 \leq k \leq l)$ and $\bar{\mathbf{r}}_k(0 \leq k \leq m) \in \mathbb{R}^{n\times 1}$ are word representations of the phrases $\mathbf{r}$ and $\bar{\mathbf{r}}$ in a phrase pair $(\mathbf{r}, \bar{\mathbf{r}})$, $\mathbf{h}_k(0 \leq k \leq l)$ and $\bar{\mathbf{h}}_k(0 \leq k \leq m) \in \mathbb{R}^{q\times 1}$ are hidden layers, $\mathbf{o}_k(1 \leq k \leq l)$ and $\bar{\mathbf{o}}_k(1 \leq k \leq m) \in \mathbb{R}^{n\times 1}$ are output layers, $\mathbf{a}_k(1 \leq k \leq l)$ and $\bar{\mathbf{a}}_k(1 \leq k \leq m) \in \mathbb{R}^{n\times 1}$ are autoencoder layers, $\mathbf{c}$ and $\bar{\mathbf{c}} \in \mathbb{R}^{q\times 1}$, which are also the last hidden layers, are the summary layers which contain the summary representations of the phrases $\mathbf{r}$ and $\bar{\mathbf{r}}$, and four types of transformation matrices: $\mathbf{I}$ and $\bar{\mathbf{I}} \in \mathbb{R}^{q\times n}$, the input vocabulary transformation matrices, $\mathbf{F}$ and $\bar{\mathbf{F}} \in \mathbb{R}^{n\times q}$, the autoencoder transformation matrices, $\mathbf{R}$ and $\bar{\mathbf{R}} \in \mathbb{R}^{q\times q}$ the recurrent transformation matrices, and $\mathbf{O}$ and $\bar{\mathbf{O}} \in \mathbb{R}^{n\times q}$, the output transformation matrices. The parameter q indicates the size of the summary representations. In Figure 1, $n := 2$ and $q := 3$ for the purposes of illustration.

Each RNN minimizes the error distance over continuous word representations of each phrase by being trained to predict the inputs, the next inputs, and the summary representation of the whole input sequence which is shared between both source- and target-sides. Hence, the hidden layer activation vectors $\mathbf{H} = \{\mathbf{h}_0, \mathbf{h}_1, ..., \mathbf{h}_l\}$ and $\bar{\mathbf{H}} = \{\bar{\mathbf{h}}_0, \bar{\mathbf{h}}_1, ..., \bar{\mathbf{h}}_m\}$ contain information of the previous input words and the next word in the input sequence. The summary layers $\mathbf{c}$ and $\bar{\mathbf{c}}$, which are the last hidden layer of each RNN, contain semantic information in common with each phrase. Any language-specific information is weakened by optimizing to jointly predict source and target. The source-side RNN learns according to following steps: first, the activations in the hidden layers $\mathbf{H}$ are calculated from the word representations recursively

as follows:

$$\mathbf{h}_k = \begin{cases} \sigma\left(\mathbf{I} \cdot \mathbf{r}_0\right) & k = 0 \\ \sigma\left(\mathbf{R} \cdot \mathbf{h}_{k-1} + \mathbf{I} \cdot \mathbf{r}_k\right) & 1 \leq k < l \end{cases} \tag{1}$$

where σ is a nonlinear function such as $\tanh$. $\mathbf{r}_0$ and $\bar{\mathbf{r}}_0$ are the representations of the source and target start symbols. When there are no constraints on the hidden layers $\mathbf{H}$, the RNNs are able to minimize the error distance by making $\mathbf{H} \rightarrow [0]$, which is undesirable. To prevent such behavior, the hidden layers $\mathbf{H}$ are normalized to have unit length. Then, the output layers $\mathbf{o}$ that predict the next word of the input, the summary layer $\mathbf{c}$ that predicts the target-side summary $\bar{\mathbf{c}}$, and the autoencoder layers $\mathbf{a}$ that predict the vector $\mathbf{r}_k$ (representing the source word at position k) are calculated from the hidden layer activations $\mathbf{H}$ in Equation (1) as follows:

$$\mathbf{o}_k = \sigma(\mathbf{O} \cdot \mathbf{h}_{k-1}) \quad (1 \leq k \leq l) \tag{2}$$

$$\mathbf{c} = \mathbf{h}_l \tag{3}$$

$$\mathbf{a}_k = \sigma(\mathbf{F} \cdot \mathbf{h}_k) \quad (1 \leq k \leq l) \tag{4}$$

Bias values for Equations (1), (2), (3), and (4) are included in the computation. To avoid overfitting, we trained each layer using the *dropout* method (Srivastava et al., 2014). There are three kinds of prediction error, which we denote: the next symbol E_o error, the semantic error E_c, and the reconstruction error E_a. These were calculated by using Euclidean distance:

$$E_o(\mathbf{r}|\mathbf{o};\theta) = \frac{1}{2l}\sum_{k=1}^{l} ||\mathbf{r}_k - \mathbf{o}_k||^2 \tag{5}$$

$$E_c(\bar{\mathbf{c}}|\mathbf{c};\theta) = \frac{1}{2}||\bar{\mathbf{c}} - \mathbf{c}||^2 \tag{6}$$

$$E_a(\mathbf{r}|\mathbf{a};\theta) = \frac{1}{2l}\sum_{k=1}^{l} ||\mathbf{r}_k - \mathbf{a}_k||^2 \tag{7}$$

where $\theta = \{\mathbf{I}, \mathbf{R}, \mathbf{F}, \mathbf{O}\}$ is the set of source-side parameters to be learned, together with the bias parameters. Equation (5) represents the sum of the next symbol error distance between each input $\mathbf{r}_k$ in the source-side phrase and the output prediction $\mathbf{o}_k$. The output from the last hidden layer $\mathbf{h}_l$ is the summary representation $\mathbf{c}$ ($\mathbf{c} \equiv \mathbf{h}_l$). A shared semantic representation of both source and target is required, and therefore $\mathbf{c}$ and $\bar{\mathbf{c}}$ are trained jointly using error signals based on the distance between them. The error is calculated using the semantic error defined in Equation (6). Equation (7) is the sum of the reconstruction error distance between each input $\mathbf{r}_k$ in the source-side phrase and the autoencoder's reconstruction $\mathbf{a}_k$. The autoencoder is used for learning representations of words (Chandar A P et al., 2014), phrases (Zhang et al., 2015), and sentences (Socher et al., 2011; Li et al., 2013). The target-side errors were calculated in the same manner. The objective function J is the sum of the total error distance from the source and target RNNs, and is represented by using Equations (5), (6), and (7) as:

$$\begin{aligned} J &= \alpha E_o(\mathbf{r}|\mathbf{o};\theta) + E_c(\bar{\mathbf{c}}|\mathbf{c};\theta) + \beta E_a(\mathbf{r}|\mathbf{a};\theta) + \lambda\|\theta\| \\ &\quad + \alpha E_o(\bar{\mathbf{r}}|\bar{\mathbf{o}};\bar{\theta}) + E_c(\mathbf{c}|\bar{\mathbf{c}};\bar{\theta}) + \beta E_a(\bar{\mathbf{r}}|\bar{\mathbf{a}};\bar{\theta}) + \lambda\|\bar{\theta}\| \\ &= 2 \cdot E_c(\bar{\mathbf{c}}|\mathbf{c};\theta,\bar{\theta}) \\ &\quad + \alpha(E_o(\mathbf{r}|\mathbf{o};\theta) + E_o(\bar{\mathbf{r}}|\bar{\mathbf{o}};\bar{\theta})) \\ &\quad + \beta(E_a(\mathbf{r}|\mathbf{a};\theta) + E_a(\bar{\mathbf{r}}|\bar{\mathbf{a}};\bar{\theta})) \\ &\quad + \lambda(\|\theta\| + \|\bar{\theta}\|) \end{aligned} \tag{8}$$

Data	Training			Development			Test			Monolingual	
	sent	word types		sent	word types		sent	word types		sent	
		en	ja		en	ja		en	ja	en	ja
IWSLT 2007	40K	9.5K	10K	0.5K	1.2K	1.3K	0.5K	0.8K	0.9K	-	-
NTCIR-10	720K	119K	85K	2.0K	5.0K	4.4K	0.5K	2.4K	2.1K	41M	81M

Table 1: Data sets

where $\bar{\theta}$ is the set of target-side parameters, and α and β are the hyper-parameters for the balance of each error. We also use an L_1 regularization term in the objective function. In Equation (8), we group the semantic error E_c terms of source- and target-side (which use the summary vectors $\mathbf{c}$ and $\bar{\mathbf{c}}$) into one term, and arrange the terms according to error type.

The parameters θ and $\bar{\theta}$ are optimized to minimize Equation (8) using the AdaGrad stochastic adaptive subgradient algorithm (Duchi et al., 2011; Green et al., 2013):

$$\theta_i = \theta_{i-1} - \eta \frac{\partial J}{\partial \theta_{i-1}} G_i^{-1/2} \tag{9}$$

$$G_i = G_{i-1} + \frac{\partial J}{\partial \theta_{i-1}}^2 \tag{10}$$

where η is the learning rate, i is the number of the training iterations, and G is the sum of the squares of the past gradients. θ and $\bar{\theta}$ are learned and updated in every iteration through the training data of phrase-pairs. The number of training iterations was determined using development data.

3.2 Word representations

Word representations, in which words are represented as real-valued vectors (Bengio et al., 2003; Mikolov et al., 2013), serve as the inputs to our model. The word representations $\mathbf{r}$ are calculated as:

$$\mathbf{r}_i = \mathbf{L}\mathbf{u}_i \in \mathbb{R}^{n \times 1} \tag{11}$$

where n is the number of dimensions of the vector, $\mathbf{L} \in \mathbb{R}^{n \times |V|}$ is a word embedding matrix, $|V|$ is the vocabulary size, and $\mathbf{u}_i$ is a binary vector which is zero in all positions except for the i^{th} index. Given a phrase which is a sequence of l words, each word has a vocabulary index i into the columns of the word embedding matrix $\mathbf{L}$. The i^{th} column of the embedding matrix is the word's representation vector. The matrix $\mathbf{L}$ is pre-trained by training a neural network on unlabeled monolingual data. In our experiments, we trained the matrices $\mathbf{L}$ and $\bar{\mathbf{L}}$ for source and target word representations using the Word2Vec toolkit (Mikolov et al., 2013). The size of the word representation vector n is usually determined empirically.

4 Experiments

We conducted two experiments with the Bilingual Phrase Representation Model: a phrase-pair extraction task and a phrase-based SMT task.

4.1 Data and model parameters

Both experiments were conducted on two English-Japanese (en-ja) corpora. One was from IWSLT 2007 (Fordyce, 2007) which is in the domain of spoken travel conversation and the other was a patent translation corpus from NTCIR-10 (Goto et al., 2013). The Japanese sentences were tokenized using KyTea (Neubig et al., 2011).

Table 1 provides statistics on each corpus. The "sent" column indicates the number of sentence pairs, and the "word types" columns of "en" and "ja" indicate the number of English and Japanese unique words. The "Monolingual" column indicates the size of the monolingual data for the training of the word representations described in Section 3.2. For IWSLT 2007, we used the training data for the training of the word representations. For NTCIR-10, we used about 723K sentence pairs belonging to the *physics* domain, which contains the most documents among the domains, according to International Patent Classification (IPC) code [1]. 720K sentence pairs from the documents published between 1993 to 2005 were used as the training data, and 2.0K and 0.5K sentence pairs randomly sampled from the 2006 and 2007 documents were used as the development and test data respectively. We also used the 2006 and 2007 documents to extract the similar phrases. Furthermore, we used the English and Japanese monolingual corpus in NTCIR-10 for the training of word representations.

For the extraction of phrase pairs, we used MGIZA++ (Gao and Vogel, 2008) and grow-diag-final-and heuristics of the Moses toolkit (Koehn et al., 2007). To facilitate effective learning, we used only phrase pairs that contained content words (i.e. had at least one noun or verb word in the phrase) and had a high translation probability (a threshold on the source-given-target conditional probability was used). We extracted phrase pairs from the training, development and test data. The training phrase-pairs were used for training the neural network models. The development phrase-pairs were used to control the training of the models. The model was trained for 4,000 iterations, and estimated parameters θ and $\bar{\theta}$ by evaluating the highest accuracy of the top-1 phrase pair extracted with the development data. The evaluation of the accuracy was performed as follows: for each source-side phrase in a 100-phrase pair development set, the system was requested to choose the top-n candidate target word sequences from the target-side 100 phrases. The minimum error distance defined in Eq. (8) was used to produce the top-n list. The test phrase-pairs were used for the experiment on the phrase-pair extraction. Consequently, we extracted about 316K phrase pairs from IWSLT 2007 and 23M phrase pairs from the NTCIR-10 training set. Due to the computation time, we randomly selected 10K phrase pairs for IWSLT 2007 and 100K phrase pairs for NTCIR-10 as the training set from the full set of phrase pairs. For the development set, we randomly selected another 300 phrase pairs. We also extracted 100 phrase pairs from the test data for the experiment on phrase-pair extraction.

For the parameters of the model, the input and output vector-size n was set to 200. The summary vector-size q was also set to 200. The activation function σ was $\tanh$. The dropout rate was 0.9 for the hidden layers and 0.5 for the other layers. The learning rate η was set to 0.01 for the experiments with IWSLT 2007 and 0.02 for the experiments with NTCIR-10. The regularization rate λ was set to 0.01. The hyperparameters α and β in Equation (8) were set to 0.01. All weight parameters θ and $\bar{\theta}$ were randomly initialized, and all bias parameters were initialized to zero.

4.2 Experimental design

4.2.1 Phrase-pair Extraction

We did two sub-experiments for the phrase-pair extraction: the evaluation of the accuracy and the extraction of phrases with similar meaning. The accuracy was calculated with the test phrase-pairs. In order to mitigate the issue of the training process terminating in a local minimum, we evaluated the accuracy at each iteration on four data sets: three different development data sets (DEV.1, DEV.2, and DEV.3) and a fourth (closed) set which was the test data itself. This resulted in four different models, each defined by the estimated parameters θ and $\bar{\theta}$ at the iteration that gave rise to the highest accuracy on the respective data set. Each of the development data sets contained 100 phrase pairs sampled randomly without replacement from the full

[1] *SECTION G* of IPC code indicates the *physics* domain

		DEV.1	DEV.2	DEV.3	closed
IWSLT 2007	BPRM (without autoencoder)				
	1-best	0.03	0.06	0.04	0.10
	10-best	0.23	0.25	0.24	0.32
	BPRM				
	1-best	0.03	0.03	0.03	0.08
	10-best	0.27	0.28	0.28	0.28
NTCIR-10	BPRM (without autoencoder)				
	1-best	0.20	0.17	0.18	0.21
	10-best	0.45	0.43	0.44	0.48
	BPRM				
	1-best	0.20	0.24	0.20	0.24
	10-best	0.43	0.41	0.45	0.47

Table 2: Accuracy of the phrase-pair extraction: 1-best and 10-best on three development sets

300-pair development set.

The data for extracting the similar phrases was obtained by searching for English phrases that were close to their Japanese counterpart phrases in NTCIR-10. We used the English phrases in unseen sentences published in 2006 and 2007 and the Japanese phrases from sentences randomly selected from the training sets. We calculated the error distance between the Japanese and the English phrases with the model terminated using the accuracy on DEV.1. To limit the number of the English phrase candidates, we only used the English sentences that were similar to the Japanese sentences. The similarity was assessed using the number of lemmatized words in the Japanese sentences, that could be translated to lemmatized words in the English sentences by using a Japanese-English dictionary.

4.2.2 Phrase-based SMT

The phrase-based SMT experiments were performed with the two models, in which the parameters θ and $\bar{\theta}$ were estimated on DEV.1 and DEV.2 of the phrase-pair extraction experiment, using the phrase-pairs extracted using the Moses toolkit. We added the inverse of the error distance used in the ranking experiments, as a feature into the log-linear model of the Moses decoder. The 5-gram language models were built using the SRILM toolkit (Stolcke, 2002) with modified Kneser-Ney smoothing (Chen and Goodman, 1996). For word and phrase alignments, we used MGIZA++ and grow-diag-final-and heuristics. To tune the weights with respect to the BLEU score (Papineni et al., 2002), we used n-best batch MIRA (Cherry and Foster, 2012). The *distortion limit* parameter was set to 10. We evaluated each model on BLEU using the NIST's *mteval-v13a.pl* [2] script. Statistical significance testing of the BLEU differences was performed using paired bootstrap resampling (Koehn, 2004).

For both experiments, we tested with two models. The first was the proposed Bilingual Phrase Representation Model (BPRM). The second was the same BPRM model with the autoencoder layer removed.

5 Results and Analysis

Tables 2 and 3 present the results of the phrase pair extraction task, and Table 4 presents the results of the phrase-based SMT task.

[2] ftp://jaguar.ncsl.nist.gov/mt/resources/mteval-v13a.pl

English Phrase in training data	English Phrases from unseen data
as was described above	as has been described above
a series of operations are	and the sequence of operations
each processors	each of the processors
an exposing device	the exposure device / the exposure apparatus

Table 3: Examples of the new phrase pairs extracted with BPRM in NTCIR-10

		DEV.1	DEV.2
IWSLT 2007	PBMT	46.55	
	+BPRM (without autoencoder)	47.75	47.65
	+BPRM	**48.19**	**48.08**
NTCIR-10	PBMT	32.13	
	+BPRM (without autoencoder)	32.14	32.35
	+BPRM	32.49	32.29

Table 4: BLEU scores on the IWSLT 2007 task (PBMT denotes phrase-based SMT)

5.1 Phrase-pair Extraction

In Table 2, the results of the experiments on NTCIR-10 data show higher levels of accuracy than the experiments on IWSLT 2007 data. The reason for this may be indicated in Table 1 which shows that the number of the word types on IWSLT 2007 is smaller than on NTCIR-10, and the proportion of words that appear multiple times in the corpus was 70% for IWSLT 2007 and 30% for NTCIR-10. The set of phrases from the IWSLT 2007 data is likely to contain many similar phrases, thereby making the discrimination more difficult. The differences between DEV.1, DEV.2 and DEV.3 were small. It is likely that there is no local minimum problem in these three models. Table 3 shows examples of phrase pairs in the training data and the English phrases which were extracted from the English monolingual documents with BPRM, illustrating the kinds of semantically similar phrases our model is capable of identifying.

5.2 Phrase-based SMT

The results of the phrase-based SMT experiments on IWSLT 2007 data show that the proposed method was able to improve machine translation quality. The statistical significance tests between PBMT and the other models shows a significant improvement on both DEV.1 and DEV.2 at $p < 0.05$. Although the results were not statistically significant, the full BPRM approach achieved higher BLEU scores than the BPRM without the autoencoder. Therefore we believe it is likely that the autoencoder is effective for the improvement of translation quality. For NTCIR-10, the improvements in performance were smaller than on the IWSLT 2007 data set.

In terms of computational time for training the model, training with 10K phrase pairs on IWSLT 2007 took about 40 seconds for one iteration, and training with 100K phrase pairs on NTCIR-10 took about 3 minutes for one iteration. Training was performed on an 8-core 2.00GHz Intel Xeon CPU.

In summary, our model was capable of identifying phrase-pairs with semantically source and target word sequences, and this knowledge could be exploited to yield an respectable improvement in machine translation quality.

6 Conclusion

In this paper, we proposed a Bilingual Phrase Representation Model which learns phrase representations by using source- and target-side Recurrent Neural Networks. We demonstrated the effectiveness of the proposed model on an English-Japanese corpus on two tasks: phrase-pair extraction and statistical machine translation. Future avenues of research include investigating hyper-parameter tuning for the objective function, and discovering a method to select appropriate initial values of the weights which were set randomly in this work.

Acknowledgements

We are deeply grateful to Taro Watanabe, Atsushi Fujita and anonymous reviewers for their suggestions and insightful comments on the early version of this paper.

References

Bengio, Y., Ducharme, R., Vincent, P., and Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, 3:1137–1155.

Chandar A P, S., Lauly, S., Larochelle, H., Khapra, M., Ravindran, B., Raykar, V. C., and Saha, A. (2014). An autoencoder approach to learning bilingual word representations. In Ghahramani, Z., Welling, M., Cortes, C., Lawrence, N., and Weinberger, K., editors, *Advances in Neural Information Processing Systems 27*, pages 1853–1861. Curran Associates, Inc.

Chen, S. F. and Goodman, J. (1996). An empirical study of smoothing techniques for language modeling. In *Proceedings of the 34th Annual Meeting of the Association for Computational Linguistics*, pages 310–318, Santa Cruz, California, USA. Association for Computational Linguistics.

Cherry, C. and Foster, G. (2012). Batch tuning strategies for statistical machine translation. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 427–436, Montréal, Canada. Association for Computational Linguistics.

Cho, K., van Merrienboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., and Bengio, Y. (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. Association for Computational Linguistics.

Devlin, J., Zbib, R., Huang, Z., Lamar, T., Schwartz, R., and Makhoul, J. (2014). Fast and robust neural network joint models for statistical machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1370–1380, Baltimore, Maryland. Association for Computational Linguistics.

Duchi, J., Hazan, E., and Singer, Y. (2011). Adaptive subgradient methods for online learning and stochastic optimization. *J. Mach. Learn. Res.*, 12:2121–2159.

Fordyce, C. S. (2007). Overview of the 4th international workshop on spoken language translation iwslt 2007 evaluation campaign. In *In Proceedings of IWSLT 2007*, pages 1–12, Trento, Italy.

Gao, Q. and Vogel, S. (2008). Parallel implementations of word alignment tool. In *Software Engineering, Testing, and Quality Assurance for Natural Language Processing*, pages 49–57, Columbus, Ohio. Association for Computational Linguistics.

Goto, I., Chow, K., Lu, B., Sumita, E., and Tsou, B. K. (2013). Overview of the patent machine translation task at the NTCIR-10 workshop. In *Proceedings of the 10th NTCIR Conference on Evaluation of Information Access Technologies, NTCIR-10, National Center of Sciences, Tokyo, Japan, June 18-21, 2013*.

Green, S., Wang, S., Cer, D., and Manning, C. D. (2013). Fast and adaptive online training of feature-rich translation models. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 311–321, Sofia, Bulgaria. Association for Computational Linguistics.

Kalchbrenner, N. and Blunsom, P. (2013). Recurrent continuous translation models. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1700–1709, Seattle, Washington, USA. Association for Computational Linguistics.

Koehn, P. (2004). Statistical significance tests for machine translation evaluation. In Lin, D. and Wu, D., editors, *Proceedings of EMNLP 2004*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Koehn, P., Hoang, H., Birch, A., Callison-Burch, C., Federico, M., Bertoldi, N., Cowan, B., Shen, W., Moran, C., Zens, R., Dyer, C., Bojar, O., Constantin, A., and Herbst, E. (2007). Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Koehn, P., Och, F. J., and Marcu, D. (2003). Statistical phrase-based translation. In *In Proceedings of HLT-NAACL*, pages 48–54, Edmonton, Canada.

Le, H.-S., Allauzen, A., and Yvon, F. (2012). Continuous space translation models with neural networks. In *Proceedings of the 2012 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 39–48, Montréal, Canada. Association for Computational Linguistics.

Li, P., Liu, Y., and Sun, M. (2013). Recursive autoencoders for ITG-based translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 567–577, Seattle, Washington, USA. Association for Computational Linguistics.

Mikolov, T. (2012). *Statistical language models based on neural networks*. PhD thesis, Brno University of Technology.

Mikolov, T., Karafit, M., Burget, L., Cernock, J., and Khudanpur, S. (2010). Recurrent neural network based language model. In Kobayashi, T., Hirose, K., and Nakamura, S., editors, *INTERSPEECH*, pages 1045–1048. ISCA.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., and Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 26: 27th Annual Conference on Neural Information Processing Systems 2013. Proceedings of a meeting held December 5-8, 2013, Lake Tahoe, Nevada, United States.*, pages 3111–3119.

Neubig, G., Nakata, Y., and Mori, S. (2011). Pointwise prediction for robust, adaptable japanese morphological analysis. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies*, pages 529–533, Portland, Oregon, USA. Association for Computational Linguistics.

Papineni, K., Roukos, S., Ward, T., and Zhu, W.-J. (2002). Bleu: a method for automatic evaluation of machine translation. In *Proceedings of 40th Annual Meeting of the Association for Computational Linguistics*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Schwenk, H. (2007). Continuous space language models. *Comput. Speech Lang.*, 21(3):492–518.

Schwenk, H. (2012). Continuous space translation models for phrase-based statistical machine translation. In *Proceedings of COLING 2012: Posters*, pages 1071–1080, Mumbai, India. The COLING 2012 Organizing Committee.

Socher, R., Pennington, J., Huang, E. H., Ng, A. Y., and Manning, C. D. (2011). Semi-supervised recursive autoencoders for predicting sentiment distributions. In *Proceedings of the 2011 Conference on Empirical Methods in Natural Language Processing*, pages 151–161, Edinburgh, Scotland, UK. Association for Computational Linguistics.

Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. (2014). Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15:1929–1958.

Stolcke, A. (2002). Srilm an extensible language modeling toolkit. In *In Proceedings of the Seventh International Conference on Spoken Language Processing*, pages 901–904.

Zhang, J., Liu, S., Li, M., Zhou, M., and Zong, C. (2014). Bilingually-constrained phrase embeddings for machine translation. In *Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 111–121, Baltimore, Maryland. Association for Computational Linguistics.

Zhang, J., Liu, S., Li, M., Zhou, M., and Zong, C. (2015). Towards machine translation in semantic vector space. *ACM Trans. Asian Low-Resour. Lang. Inf. Process.*, 14(2):9:1–9:26.

Zou, W. Y., Socher, R., Cer, D., and Manning, C. D. (2013). Bilingual word embeddings for phrase-based machine translation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1393–1398, Seattle, Washington, USA. Association for Computational Linguistics.