Bridging Source and Target Word Embeddings for Neural Machine Translation

Shaohui Kuang[†] **Junhui Li**[†] **Deyi Xiong**[†]
[†]School of Computer Science and Technology,
Soochow University, Suzhou, China
shkuang@stu.suda.edu.cn, {lijunhui, dyxiong}@suda.edu.cn

Abstract

Neural machine translation systems encode a source sequence into a vector from which a target sequence is generated via a decoder. Different from the traditional statistical machine translation, source and target words are not directly mapped to each other in translation rules. They are at the two ends of a long information channel in the encoder-decoder neural network, separated by source and target hidden states. This may lead to translations with inconceivable word alignments. In this paper, we try to bridge source and target word embeddings so as to shorten their distance. We propose three strategies to bridge them: 1) a source state bridging model that moves source word embeddings one step closer to their counterparts, 2) a target state bridging model that explores relevant source word embeddings for target state prediction, and 3) a direct link bridging model that directly connects source and target word embeddings so as to minimize their discrepancy. Experiments and analysis demonstrate that the proposed bridging models are able to significantly improve quality of both translation and word alignments.

Introduction

Neural machine translation (NMT) is an end-to-end approach to machine translation that has achieved remarkable success over the state-of-the-art of statistical machine translation (SMT) on various language pairs (Bahdanau, Cho, and Bengio 2015; Cho et al. 2014; Sutskever, Vinyals, and Le 2014; Luong and Manning 2015). In NMT, the sequence-to-sequence (seq2seq) model learns word embeddings for both source and target words synchronously. However, as shown in Figure 1, source and target word embeddings locate at the two ends of a long information channel. The mapping and connection between them will gradually become obscure due to the separation of source-side hidden states (i.e., $h_1, \ldots, h_T$, and consequently c_t) and target-side hidden states (i.e., s_t). As a result, with the absence of direct interaction between source and target word pairs, the seq2seq model in NMT may produce translations that contain ridiculous word alignments.

Different from SMT, NMT adopts an attention model to align every target word to input words. The attention model estimates a probability distribution over all input words for each target word. Word alignments with attention weights can be easily deduced from such distributions. While it is yet to be seen how good word alignments can be obtained and how much they will benefit NMT, we find that translations of NMT sometimes contain surprisingly unfavorable word alignments. Figure 2 (a) shows a Chinese-to-English translation example of NMT. In this example, the NMT seq2seq model incorrectly aligns the target side *eos* to 下旬/*late* with a high attention weight (i.e., 0.80 in this example) due to the failure of capturing the word similarity between source word 下旬 and target *eos*. Statistics on our development data show that as high as 50% of target side *eos* do not properly align to source side *eos*. It is worth to note that, as 本/*this* and 月/*month* are not translated in this example, inappropriate alignment of target side *eos* is likely responsible for under-translation in NMT as the decoding process ends once a target *eos* is generated. Figure 2 (b) shows another example of unintelligible word translation, where source word 冬

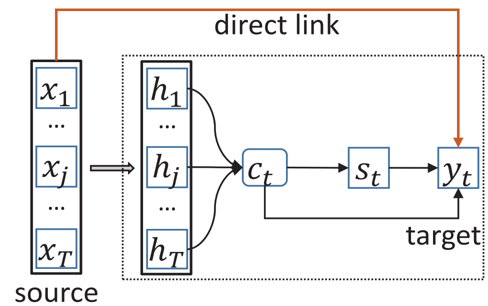

Figure 1: Relation of source and target word embeddings in NMT. Bridging them with a direct link (above) is our key motivation.

Figure 2: NMT translation examples that contain unintelligible word alignments.

奥会/*winter olympics* is mistakenly translated into a target comma "," and 载誉/*honers* into *have*. Usually unintelligible word translations happen along with bad word alignments which result in inadequate translation.

In this paper we try to shorten the information channel between source and target word embeddings. Bridging them with a direct link as illustrated in Figure 1 is our ultimate goal. In doing so, we hope that the seq2seq model is able to learn more desirable word alignments between source and target words by directly comparing their embeddings. On this basis, we propose several simple yet effective strategies to bridge word embeddings on the source and target side. We categorize these strategies into three bridging models in terms of how close the source and target word embeddings are along the information channel shown in Figure 1.

1) **Source state bridging model:** The basic idea is to move source word embeddings towards target word embeddings along the information channel. Our first strategy is to just move one step closer to the target side. We concatenate each source word embedding with the source hidden state at the same position so that the attention model can directly explore source word embeddings to produce word alignments.
2) **Target state bridging model:** More radically, we want to incorporate relevant source word embeddings into the prediction of the next target hidden state. In particular, we select the most appropriate source words according to the attention weights and make them directly interact with target hidden states.
3) **Direct link bridging model:** The final goal is to bridge source and target word embeddings with a direct link. In this model, we optimize the training objective towards minimizing the distance between target word embeddings and their most relevant source word embeddings selected according to the attention model.

Experiments on Chinese-English translation with extensive analysis demonstrate that bridging word embeddings of two sides can achieve better word alignments. The translation quality is also significantly improved.

Attention-based NMT

As a background and a baseline, in this section, we briefly describe the NMT model. Without loss of generality, we take the NMT architecture proposed by Bahdanau, Cho, and Bengio (2015) in this paper, which is established on an encoder-decoder neural network.

Encoder

The encoder uses bidirectional recurrent neural networks (Bi-RNN) to encode a source sentence with a forward and backward RNN. The forward RNN reads a source sentence $x = (x_1, x_2, ..., x_T)$ from left to right and outputs a hidden state sequence $(\overrightarrow{h_1}, \overrightarrow{h_2}, ..., \overrightarrow{h_T})$ while the backward RNN reads sentence in an inverse direction, and outputs a backward hidden state sequence $(\overleftarrow{h_1}, \overleftarrow{h_2}, ..., \overleftarrow{h_T})$. The context-dependent word representations of the source sentence h_j (also known as word annotation vectors) are the concatenation of hidden states $\overrightarrow{h_j}$ and $\overleftarrow{h_j}$ in two directions, which contain contextual information.

Decoder

The decoder is also an RNN that predicts a target word y_t via a multi-layer perceptron (MLP) neural network. The prediction is based on the decoder RNN hidden state s_t, the previous predicted word y_{t-1} and a source-side context vector c_t. The hidden state s_t of decoder at time t and conditional probability of the next word y_t are computed as follows:

$$s_t = f(s_{t-1}, y_{t-1}, c_t) \tag{1}$$

$$p(y_t|y_{<t}; x) = g(y_{t-1}, s_t, c_t) \tag{2}$$

Attention Model

In the attention model, the context vector c_t is calculated as a weighted sum over source annotation vectors $(h_1, h_2, ..., h_T)$:

$$c_t = \sum_{j=1}^{T_x} \alpha_{tj} h_j \tag{3}$$

$$\alpha_{tj} = \frac{exp(e_{tj})}{\sum_{k=1}^{T} exp(e_{tk})} \tag{4}$$

$$e_{tj} = a(s_{t-1}, h_j) \tag{5}$$

where α_{tj} is the attention weight of each hidden state h_j computed by the attention model, and a is a feed forward neural network with a single hidden layer.

The dl4mt tutorial [1] has presented an improved implementation of the attention-based NMT system, which feeds the previous word y_{t-1} to the attention model and computes e_{tj} as follows:

$$e_{tj} = a(\widetilde{s}_{t-1}, h_j), \tag{6}$$

where $\widetilde{s}_{t-1} = GRU(s_{t-1}, y_{t-1})$. The hidden state of the decoder is updated as follows:

$$s_t = GRU(\widetilde{s}_{t-1}, c_t) \tag{7}$$

we use the dl4mt tutorial implementation as our baseline, which is referred to as RNNSearch* in this paper.

Objective Function of Training

We train the NMT model by minimizing the negative log-likelihood on a set of training data $\{(x^n, y^n)\}_{n=1}^{N}$:

$$L(\theta) = -\frac{1}{N} \sum_{n=1}^{N} \sum_{t=1}^{T_y} logp(y_t^n | y_{<t}^n, x^n) \tag{8}$$

[1] https://github.com/nyu-dl/dl4mt-tutorial/tree/master/session2

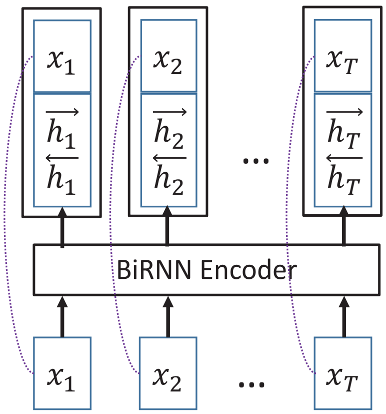

Figure 3: Architecture of NMT with the source state bridging model.

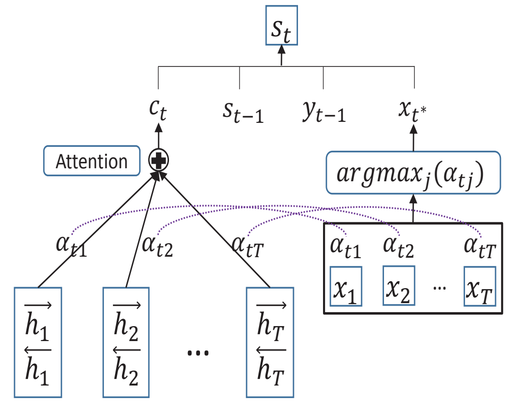

Figure 4: Architecture of NMT with the target state bridging model.

Word Embedding Bridging Models

The conventional NMT models opt to generate discrepancies between word embeddings of the source and target sides due to a long information channel between them. To narrow such discrepancies, we focus on building additional links between word embeddings of two sides so as to shorten distances between them. As illustrated in Figure 1, there may exist many ways to connect x to y_t. As representatives, we propose three bridging models to capture the collocation of y and its corresponding x. With respect to the distance of the source and target word embeddings along the information channel, we group our approaches into 1) source state bridging, 2) target state bridging and 3) direct link bridging.

Source State Bridging Model

Figure 3 illustrates the source state bridging model. In the model, the encoder reads a word sequence equipped with word embeddings and generates a word annotation vector for each position. Then we simply concatenate the word annotation vector with its corresponding word embedding as the final annotation vector. For example, the final annotation vector h_j for the word x_j in Figure 3 is $[\overrightarrow{h_j}; \overleftarrow{h_j}; x_j]$, where the first two sub-items $[\overrightarrow{h_j}; \overleftarrow{h_j}]$ are the source-side forward and backward hidden states and x_j is the corresponding word-embedding. In this way, the word embeddings will not only be used to compute attention weights (Equation 5), but also be along with hidden states (Equation 3) to form the weighted source context vector and consequently be used to predict target words.

Target State Bridging Model

While the above source state bridging method uses weighted word embeddings, we now explore only relevant source word embeddings for bridging, rather than all of them. This is partially inspired by word alignments in SMT where words on two sides are determinatively aligned. In particular, we explicitly calculate the main aligned source word and use its word embedding to predict the target hidden state for the coming word. Figure 4 illustrates the target state bridging method, where the inputs for computing the hidden state s_t of the decoder are additionally augmented by x_{t^*}. Accordingly we replace Equation (1) with the following formulation:

$$s_t = f(s_{t-1}, y_{t-1}, c_t, x_{t^*}) \tag{9}$$

where x_{t^*} is the word embedding of the selected source word which has the highest attention weight:

$$t^* = \arg\max_j(\alpha_{tj}) \tag{10}$$

Direct Link Bridging Model

Unlike the above two bridging methods which use source word embeddings to predict target words in a direct or indirect way, next we try to bridge the two types of word embeddings with a direct link. This is done by proposing an auxiliary objective function to narrow the discrepancy between word embeddings of the two sides. Figure 5 presents our direct link bridging method with an auxiliary objective function. More specifically, our goal is to let the learned word embeddings on the two sides be transformable. That is to say, if a target word e_i aligns to a source word f_j, we learn a transformation matrix W with the hope that the discrepancy of Wx_i and y_j tends to be zero. Accordingly, we update the objective function of training from Equation 8 to the following objective function:

$$\widetilde{L}(\theta) = -\frac{1}{N}\sum_{n=1}^{N}\sum_{t=1}^{T_y}\{\log p(y_t^n|y_{<t}^n, x^n) - \|Wx_{t^*}^n - y_t^n\|^2\} \tag{11}$$

where the term $\|Wx_{t^*} - y_t\|^2$ measures and penalizes the difference between target word y_t and its aligned source word x_{t^*}, i.e., the one with the highest attention weight, as computed in Equation 10. Similar to Mi, Wang, and Ittycheriah (2016), we view the two parts of loss in Equation 11 equally important.

Note that:

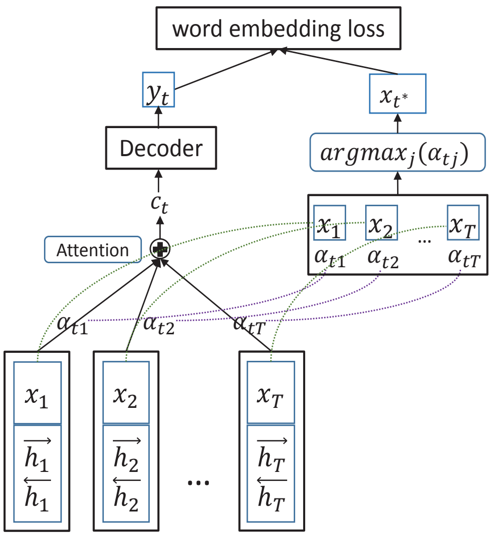

Figure 5: Architecture of NMT with the direct link bridging model.

- Our direct link bridging model is an extension of the source state bridging model, where the source word embeddings are part of final annotation vector of the encoder. We have also tried to put the auxiliary object function directly on the NMT baseline model, however, our empirical study shows that the combined objective consistently worsens the translation quality. We blame this on that the learned word embeddings on two sides by the baseline model are too heterogeneous to be constrained.
- Rather than using a concrete source word's embedding $x^n_{t^*}$ in Equation 11, we could also use a weighted sum of source word embeddings, i.e., $\sum_j \alpha_{tj} h_j$. However, our preliminary experiments show that the performance gap between these two methods is very small. Therefore, we use $x^n_{t^*}$ to calculate the new training objective as shown in Equation 11 in all experiments.

Experimentation

We have presented our approaches to bridging word embeddings of the source and target side into NMT model. In this section, we conducted a series of experiments on Chinese-English translation to examine the effectiveness of the proposed method.

Experimental Settings

We used Chinese-English bilingual corpora that contain 1.25M sentence pairs extracted from LDC corpora, with 27.9M Chinese words and 34.5M English words respectively.[2] We chose NIST06 dataset as our development set, and the NIST02, NIST03, NIST04, NIST08 datasets as our test sets.[3] We used the case-insensitive 4-gram NIST BLEU score as our evaluation metric (Papineni et al. 2002) and the script 'mteval-v11b.pl' to compute BLEU scores. We also report TER scores on our dataset (Snover et al. 2006).[4]

For efficient training of the neural networks, we limited the source and target vocabularies to the most frequent 30k words in both Chinese and English, covering approximately 97.7% and 99.3% of the two corpora respectively. All the out-of-vocabulary words were mapped to a special token *UNK*. The dimension of word embedding was 620 and the size of the hidden layer was 1000. All other settings were the same as in Bahdanau, Cho, and Bengio (2015). The maximum length of sentences that we used to train NMT model in our experiments was set to 50 for both the Chinese and English side.

We compared our proposed models against the following two systems:

- cdec (Dyer et al. 2010): an open source hierarchical phrase-based SMT system (Chiang 2007) with default configuration and a 4-gram language model trained on the target portion of the training data. [5]
- **RNNSearch***: the attention-based NMT system with slight changes taken from dl4mt tutorial. It improves the attention model by feeding the lastly generated word. For the activation function f of an RNN, we use the gated recurrent unit (GRU) (Chung et al. 2014). Dropout was applied only on the output layer and the dropout (Hinton et al. 2012) rate was set to 0.5. We used the stochastic gradient descent algorithm with mini-batch and Adadelta (Zeiler 2012) to train the NMT models. The mini-batch was set to 80 sentences and decay rates ρ and ε of Adadelta were set to 0.95 and 10^{-6}. Additionally, during decoding, we used the beam-search algorithm and set the beam size to 10. The model parameters were selected according to the maximum BLEU points on the development set.

For the NMT with direct link bridging model, we used a simple pre-training strategy to train the additional embedding transformation parameter W in Equation 11. The reason of using pre-training strategy is that the embedding loss requires well-trained word alignment at the starting point.

Experimental Results

Table 1 shows the translation performances measured in BLEU and TER scores. Clearly, all the proposed NMT models that bridge source and target embeddings improve the translation accuracy over all test sets.

Parameters The three proposed models introduce new parameters in different ways. The target state bridging model introduces 1.8M additional parameters for catering x_{t^*} in calculating target side state, as in Equation 9. The source stage bridging model augments source hidden states from 2,000 dimension to 2,620, requiring 3.7M additional parameters to accommodate the appended hidden states. Based

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

[3] http://itl.nist.gov/iad/mig/test/mt/

[4] http://www.cs.umd.edu/~snover/tercom/

[5] https://github.com/redpony/cdec

	Model	NIST06	NIST02	NIST03	NIST04	NIST08	Avg
BLEU	cdec	34.00	35.81	34.70	37.15	25.28	33.23
	RNNSearch*	35.92	37.88	36.21	38.83	26.30	34.81
	Source bridging	36.79‡	38.71‡	37.24‡	40.28‡	27.40‡	35.91
	Target bridging	36.69	39.04‡	37.63‡	40.41‡	**27.98**‡	36.27
	Direct link	**36.97**‡	**39.77**‡	**38.02**‡	**40.83**‡	27.85‡	**36.62**
TER	cdec	58.29	59.65	59.28	58.12	61.54	59.64
	RNNSearch*	59.56	57.79	59.25	57.88	64.22	59.78
	Source bridging	58.13	**56.25**	57.33	56.32	62.13	**58.01**
	Target bridging	58.01	56.27	57.76	56.33	**62.12**	58.12
	Direct link	**57.20**	56.68	**57.29**	**55.62**	62.49	58.02

Table 1: BLEU and TER scores on the NIST Chinese-English translation tasks. The BLEU scores are case-insensitive. Avg means the average scores on all test sets. "‡": statistically better than RNNSearch* ($p < 0.01$). Higher BLEU (or lower TER) scores indicate better translation quality.

on the source state bridging model, the direct link bridging model requires extra 0.4M parameters (i.e., the transformation matrix W in Equation 11 is $620 * 620$), resulting 4.1M additional parameters over the baseline. Comparing to the baseline model which has 74.8M parameters, the sizes of extra parameters in our proposed models are small.

Comparison with the baseline systems Table 1 shows that all NMT systems outperform the SMT system. This is very consistent with other studies on Chinese-to-English translation (Mi, Wang, and Ittycheriah 2016; Tu et al. 2016; Li et al. 2017).

Moreover, all the three proposed NMT models outperform the baseline NMT model RNNSearch*. With respect to BLEU scores, we observe a consistent performance trend that the target state bridging model works better than the source state bridging model, while the direct link bridging model achieves the best accuracy over all test sets with the only exception of NIST MT 08. On all test sets, the direct link bridging model outperforms RNNSearch* by 1.81 BLEU points and outperforms the other two improved NMT models by 0.4∼0.6 BLEU points. Though all models are not tuned on TER score, our three models perform favorably well with similar average improvement (e.g., about 1.70 TER points) over the baseline model.

Analysis

As the proposed direct link bridging system achieves the best performance, we further look at the RNNSearch* system and the direct link bridging system to explore more on how bridging source and target word embeddings helps in translation.

Analysis on Word alignment

Due to the fact that our improved model explicitly bridges source and target word embeddings in NMT model, we expect to observe better word alignment quality. We examine word alignment quality from the following three aspects.

Better *eos* Translation As a special symbol, target side *eos* has a critical impact on controlling the length of translation. Assumingly, a correct translation of target *eos* is extractly from *eos* of the source side, indicating the end of translation. Table 2 illustrates the percentages of target side *eos* that are translated from the source side *eos*. It shows that our improved model substantially achieves better translation of *eos*.

System	Percentage (%)
RNNSearch*	49.82
Direct link	81.30

Table 2: Percentage of target side *eos* translated from source side *eos* on the development set.

Better Word Translation To have a better understanding of word translation, we group words in translation by their part-of-speech (POS) tags and examine their aligned source words. [6] Table 3 shows the translation matrix. For example, in RNNSearch* 64.95% of verbs in translation originate from verbs in the source side. This is adjusted to 66.29% in our improved model. We observe from the table that more target words in our improved model align to source words with the same POS tags.

Better Word Alignment Next we report word alignment quality on a manually aligned dataset. We carried out experiments of the word alignment task on the evaluation dataset from Liu and Sun (2015), which contains 900 manually aligned Chinese-English sentence pairs. We forced the decoder to output reference translations, so as to get automatic alignments between input sentences and their reference translations. To evaluate alignment performance, we report the alignment error rate (AER) (Och and Ney 2003) and soft AER (SAER) (Tu et al. 2016) in Table 4.

Table 4 shows that bridging source and target word embeddings improves the alignment quality as expected by maintaining a certain relationship between the word embeddings of the two sides.

[6] We use Stanford POS tagger (Toutanova et al. 2003) to get POS tags for source sentences and target translations.

System	Target POS Tag	Source POS Tag				
		V	N	CD	JJ	AD
RNNSearch*	V	64.95	-	-	-	12.09
	N	7.31	39.24	-	-	-
	CD	-	33.37	53.40	-	-
	JJ	-	26.79	-	14.67	-
Direct Link	V	66.29	-	-	-	10.94
	N	7.19	39.71	-	-	-
	CD	-	32.25	56.29	-	-
	JJ	-	26.12	-	15.22	-

Table 3: Translation matrix in percentage. To omit the subtle differences among verbs (e.g., VV, VC and VE in Chinese, and VB, VBD, VBP, etc. in English), we merge all verbs into V. Similarly, we merge all nouns into N. CD for Cardinal numbers, JJ for adjectives or modifiers, AD for adverbs. These POS tags exists in both Chinese and English. For simplicity, we present the two most aligned source POS tags for each target POS tag.

System	SAER	AER
RNNSearch*	62.68	47.61
Direct link	59.72	44.71

Table 4: Evaluation of alignment quality. The lower the score is, the better alignment quality.

Analysis on Long Sentence Translation

Following Bahdanau, Cho, and Bengio (2015), we group sentences of similar lengths together and compute BLEU scores. Figure 6 presents the BLEU scores over different lengths of input sentences. It shows that our improved system outperforms RNNSearch* over sentences with all different lengths. It also shows that the performance drops substantially when the length of input sentences increases. This performance trend over the length is consistent with the findings in (Cho et al. 2014; Tu et al. 2016; Li et al. 2017). We also observe that the NMT systems perform surprisingly bad on sentences over 50 in length, especially compared to the performance of SMT system (i.e., cdec). We think that the bad behavior of NMT systems towards long sentences (e.g., length of 50) is due to the following two reasons: (1) the maximum source sentence length limit is set as 50 in training, making the learned models not ready to translate sentences over the maximum length limit; (2) NMT systems tend to stop early for long input sentences.

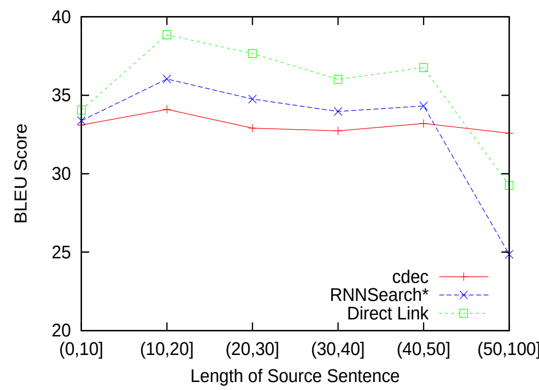

Figure 6: BLEU score of generated translations with respect to the lengths of the input sentences.

Analysis on Over Translation and Under Translation

To test the conjecture that better translation of *eos* opts to end the translation of a sentence properly, we analyze the performance of our best model with respect to over translation and under translation, both of which are notoriously hard for NMT.

To estimate the over translation generated by NMT system, we follow Li et al. (2017) and report the ratio of over translation (ROT):

$$ROT = \frac{\sum_{w_i} t(w_i)}{|w|} \tag{12}$$

where $|w|$ is the number of words in consideration, $t(w_i)$ is the number of times of over translation for word w_i which is computed as follows:

$$t(w) = |e| - |uniq(e)| \tag{13}$$

where $|e|$ is the number of words in w's translation e, while $|uniq(e)|$ is the number of unique words in e.

Table 5 displays ROTs grouped by some typical POS tags. We find that both systems have higher ROT in POS tags of NN and NR than other POS tags. The observation is is consistent with Li et al. (2017) since the two POS tags usually have more unknown words, which tend to be over translated. It also shows that our improved system alleviates the over translation issue by 15%: ROT drops from 5.28% to 4.49%.

While it is hard to obtain an accurate estimation on under translation, we simply report 1-gram BLEU score that measures how many words in translation appear in reference, roughly indicating how many source words are translated. Table 6 presents the average 1-gram BLEU scores on our test datasets. It is reasonable for cdec system having the highest 1-gram BLEU score since under translation infrequently exists in SMT model. It also shows that our improved system has higher score than RNNSearch*, suggesting less under translation.

System	POS	**ROT**(%)
RNNSearch*	NN	8.63
	NR	12.92
	DT	4.01
	CD	7.05
	ALL	5.28
Direct Link	NN	7.56
	NR	10.88
	DT	2.37
	CD	4.79
	ALL	4.49

Table 5: Ratio of over translation (ROT) on test sets. Here NN for nouns excluding proper nouns and temporal nouns, NR for proper nouns, DT for determiner, and CD for cardinal number.

System	1-gram
cdec	77.09
RNNSearch*	72.70
Direct link	74.22

Table 6: 1-gram BLEU scores averaged on test sets.

Src	Transformation	Lexical Table
是	is	is
和	and	and
及	and	and
将	will	will
会	will	will
国	countries	countries
发展	development	development
经济	economic	economic
问题	**question**	**issue**
人民	people	people

Table 7: Top 10 most frequent source words and their closest translations obtained by embedding transformation in NMT and lexical translation table in SMT, respectively.

Analysis on Learned Word Embeddings

In our direct link bridging model, we introduced a transformation matrix to convert a source-side word embedding to its counterpart on the target side. Next we examine how good the transformation is. Given a source word x_i, we obtain its closest target word y^* via:

$$y^* = \arg\min_y(\|wx_i - y\|) \tag{14}$$

Table 7 lists the most frequent 10 source words and its corresponding closest target words. For better comparison, it also gives their most likely translations from lexical translation table in SMT. From the table, we observe that the closest target words obtained are very consistent with those of SMT lexical table, suggesting the capability of capturing similarity between source and target word embeddings in our improved model.

Related Work

Since the pioneer work of joint learning alignment and translation in NMT (Bahdanau, Cho, and Bengio 2015), many effective approaches have been proposed to further improve alignment quality through different perspectives.

Attention model plays a critic role in alignment quality, and thus attracts continuous attention. To obtain better focuses, Luong, Pham, and Manning (2015) propose global and local attention models. Cohn et al. (2016) extend the attentional model to include structural biases from word based alignment models, including positional bias, Markov conditioning, fertility and agreement over translation directions. On the contrast, we do not focus on the attention model, but achieve better alignment quality by letting NMT model capture favorable word pairs on two sides.

Recently there have also been studies towards leveraging word alignments from SMT models. Mi, Wang, and Ittycheriah (2016) and Liu et al. (2016) use pre-obtained word alignments to guild NMT attention model for learning favorable word pairs. Arthur, Neubig, and Nakamura (2016) leverage a pre-obtained word dictionary to constrain the prediction of target words. Despite having similar motivations of using favorable word pairs to benefit NMT translation, we do not use extra resources in our NMT model, but let the model itself learn similarity of word pairs from data. [7]

Besides, there also exist studies on learning cross-lingual embeddings for machine translation. In SMT, Mikolov, Le, and Sutskever (2013) first learn distributed representation of words from large monolingual data, then learn a linear mapping between vector spaces of languages. Gehring et al. (2017) introduce source word embeddings to predict target word. This is similar to our source state bridging model. However, in this paper we proposed more strategies to bridge source and target word embeddings following the same motivation.

Conclusion

We have presented three models to bridge source and target word embeddings for NMT, either on the source/target side or via a direct link. The three models try to connect source and target words and shorten the distance between them along the long information channel in the encoder-decoder network in order to convey source signals to the target side more directly. Experiments on Chinese-English translation shows that the proposed models can significantly improve translation quality. Further in-depth analyses demonstrate that our models are able to 1) learn better word alignments than the baseline NMT, 2) alleviate the notorious problems of over and under translation in NMT and 3) learn direct mappings between source and target words.

In the future, we will further explore more strategies to bridge the source and target side for sequence-to-sequence or tree-based neural machine translation. Additionally, we also want to apply our methods to other sequence-to-sequence tasks, e.g., neural conversation.

[7] Though the pre-obtained word alignment and pre-obtained word dictionary can be learned unsupervisedly from MT training data, these are still extra knowledge to NMT models.

References

[2016] Arthur, P.; Neubig, G.; and Nakamura, S. 2016. Incorporating discrete translation lexicons into neural machine translation. In *Proceddings of EMNLP 2016*, 7–12.

[2015] Bahdanau, D.; Cho, K.; and Bengio, Y. 2015. Neural machine translation by jointly learning to align and translate. In *Proceedings of ICLR 2015*.

[2007] Chiang, D. 2007. Hierarchical phrase-based translation. *computational linguistics* 33(2):201–228.

[2014] Cho, K.; van Merrienboer, B.; Caglar Gulcehre, D. B.; Bougares, F.; Schwenk, H.; and Bengio, Y. 2014. Learning phrase representations using rnn encoder-decoder for statistical machine translation. In *Proceedings of EMNLP 2014*, 1724–1734.

[2014] Chung, J.; Gulcehre, C.; Cho, K.; and Bengio, Y. 2014. Empirical evaluation of gated recurrent neural networks on sequence modeling. In *Proceedings of Deep Learning and Representation Learning Workshop in NIPS 2014*.

[2016] Cohn, T.; Hoang, C. D. V.; Vymolova, E.; Yao, K.; Dyer, C.; and Haffari, G. 2016. Incorporating structural alignment biases into an attentional neural translation model. In *Proceedings of NAACL 2016*, 876–885.

[2010] Dyer, C.; Weese, J.; Setiawan, H.; Lopez, A.; Ture, F.; Eidelman, V.; Ganitkevitch, J.; Blunsom, P.; and Resnik, P. 2010. cdec: A decoder, alignment, and learning framework for finite-state and context-free translation models. In *Proceedings of the ACL 2010 System Demonstrations*, 7–12. Association for Computational Linguistics.

[2017] Gehring, J.; Auli, M.; Grangier, D.; Yarats, D.; and Dauphin, Y. N. 2017. Convolutional sequence to sequence learning. *arXiv preprint arXiv:1705.03122*.

[2012] Hinton, G. E.; Srivastava, N.; Krizhevsky, A.; Sutskever, I.; and Salakhutdinov, R. R. 2012. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580*.

[2017] Li, J.; Xiong, D.; Tu, Z.; Zhu, M.; Zhang, M.; and Zhou, G. 2017. Modeling source syntax for neural machine translation. In *Proceedings of ACL 2017*.

[2015] Liu, Y., and Sun, M. 2015. Contrastive unsupervised word alignment with non-local features. In *Proceedings of AAAI 2015*, 2295–2301.

[2016] Liu, L.; Utiyama, M.; Finch, A.; and Sumita, E. 2016. Neural machine translation with supervised attention. In *In Proceddings of COLING 2016*.

[2015] Luong, M.-T., and Manning, C. D. 2015. Stanford neural machine translation systems for spoken language domains. In *Proceedings of the International Workshop on Spoken Language Translation*.

[2015] Luong, M.-T.; Pham, H.; and Manning, C. D. 2015. Effective approaches to attention-based neural machine translation. In *Proceedings of EMNLP 2015*.

[2016] Mi, H.; Wang, Z.; and Ittycheriah, A. 2016. Supervised attentions for neural machine translation. In *Proceddings of EMNLP 2016*.

[2013] Mikolov, T.; Le, Q. V.; and Sutskever, I. 2013. Exploiting similarities among languages for machine translation. *arXiv preprint arXiv:1309.4168*.

[2003] Och, F. J., and Ney, H. 2003. A systematic comparison of various statistical alignment models. *Computational linguistics* 29(1):19–51.

[2002] Papineni, K.; Roukos, S.; Ward, T.; and Zhu, W.-J. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of ACL 2002*, 311–318. Association for Computational Linguistics.

[2006] Snover, M.; Dorr, B.; Schwartz, R.; Micciulla, L.; and Makhoul, J. 2006. A study of translation edit rate with targeted human annotation. In *Proceddings of AMTA 2006*.

[2014] Sutskever, I.; Vinyals, O.; and Le, Q. V. 2014. Sequence to sequence learning with neural networks. In *Proceedings of NIPS 2014*, 3104–3112.

[2003] Toutanova, K.; Klein, D.; Manning, C.; and Singer, Y. 2003. Feature-rich part-of-speech tagging with a cyclic dependency network. In *Proceedings of HLT-NAACL 2003*, 252–259.

[2016] Tu, Z.; Lu, Z.; Liu, Y.; Liu, X.; and Li, H. 2016. Modeling coverage for neural machine translation. In *Proceedings of ACL 2016*, 76–85.

[2012] Zeiler, M. D. 2012. Adadelta: an adaptive learning rate method. *arXiv preprint arXiv:1212.5701*.