Front. Comput. Sci., 2021, 15(1): 151302
https://doi.org/10.1007/s11704-019-9164-3

RESEARCH ARTICLE

# Improving neural sentence alignment with word translation

**Ying DING, Junhui LI, Zhengxian GONG (✉), Guodong ZHOU**

School of Computer Science and Technology, Soochow University, Suzhou 215006, China



**Abstract** Sentence alignment is a basic task in natural language processing which aims to extract high-quality parallel sentences automatically. Motivated by the observation that aligned sentence pairs contain a larger number of aligned words than unaligned ones, we treat word translation as one of the most useful external knowledge. In this paper, we show how to explicitly integrate word translation into neural sentence alignment. Specifically, this paper proposes three cross-lingual encoders to incorporate word translation: 1) Mixed Encoder that learns words and their translation annotation vectors over sequences where words and their translations are mixed alternatively; 2) Factored Encoder that views word translations as features and encodes words and their translations by concatenating their embeddings; and 3) Gated Encoder that uses gate mechanism to selectively control the amount of word translations moving forward. Experimentation on NIST MT and Opensubtitles Chinese-English datasets on both non-monotonicity and monotonicity scenarios demonstrates that all the proposed encoders significantly improve sentence alignment performance.

**Keywords** sentence alignment, word translation, mixed encoder, factored encoder, gated encoder

## 1 Introduction

Sentence alignment, aiming to find semantically equivalent sentence pairs in given bitexts, remains an essential and challenging component in construction of parallel corpora, which are fundamental to various multilingual natural language processing applications such as machine translation [1, 2], multilingual word representation [3], and cross-lingual information retrieval [4, 5]. Most traditional sentence alignment approaches [6] mainly depend on manually designed features (e.g., length ratios and word pairs), and thus suffer from the sparsity problem due to the language ambiguity. As neural networks recently show its powerful capability of modeling distributed representations, neural sentence alignment starts to shed light in this literature [7–10]. For example, Gregoire and Langlais [7] resort to sentence modeling which maps an input sentence into a fixed-length vector and then predict if two sentences are aligned by their sentence vectors. Considering that sentence-level representation fails to capture word alignment details which are promising evidences for sentence alignment, studies in [8–10] propose word-level approaches which calculate the similarity between word pairs to capture fine-grained word-level information. In an aligned sentence pair, words in one sentence usually have proper translations in the other. However, recent studies [11, 12] have shown that even with the complicated attention mechanism, neural models still fail to capture a large portion of word translation details, even though it is capable of learning certain word translations from parallel corpus.

In this paper, we address how to explicitly incorporate word translation, as external knowledge to enhance word distributed representation to improve sentence alignment performance. As shown in Fig. 1, the sentence pair ($Src$, $Trg$)'s word translation, i.e., $Src\_WT$ and $Trg\_WT$, can be viewed as extra useful knowledge for sentence alignment. Different from the representative approach in Arthur et al. [13] which constrains word prediction in decoding process with pre-prepared word translation tables, we take word translations as external inputs and let the encoder automatically learn useful information. Therefore, we propose and compare three different encoders: *Mixed Encoder*, *Factored Encoder* and *Gated Encoder* to incorporate word translations into the neural sentence alignment model. Experimentation on Chinese-English NIST MT dataset and Opensubtitles dataset demonstrates that all the three cross-lingual encoders can effectively improve the quality of sentence alignment.

## 2 Related work

The study of sentence alignment can be traced back to the 1990s. Gale and Church [14] use the length statistics of bilingual sentences which is based on the idea that the closer two sentence are in length, the more likely they are to align. However as shown with Chen [15] and Wu [16], sentence-length based approaches are not robust and severely dependent on the language pair involved. In 2000s, Moore [17] proposes a multi-pass procedure to search the best alignment. It first uses sentence-length model to generate a set of sentence pairs from

| | | | | | |
|---|---|---|---|---|---|
| Src | 来自 | 空中 | 的 | 战争 | 威胁 |
| Src_WT | from | air | of | war | thereat |
| Trg | war | threats | from | the | sky |
| Trg_WT | 战争 | 威胁 | 从 | 的 | 天空 |

**Fig. 1** An example of the model's inputs. *WT: word translations as the extra input in this paper

Received May 10, 2019; accepted September 10, 2019

E-mail: zhxgong@suda.edu.cn

the input bi-text strictly for training IBM translation model 1 [18] and re-aligns the bi-text using both sentence-length model and the generated word-correspondence model. Braune and Fraser [19] develop an unsupervised and two-step clustering approach, which slightly modifies Moore's alignment model to find a model-optimal alignment, namely 1-0/0-1 and 1-1, and then merges those correspondences into large alignment like 1-many/many-1. Then, meeting the challenge of noise data, Ma [20] proposes a robust and lexicon-based parallel text sentence aligner - Champollion. It borrows the idea of *tf-idf* to compute the similarity of two sentences and then uses dynamic programming algorithm to produce final alignment. To reduce running time of Champollion, Li et al. [21] first split the bi-texts into small aligned fragments and then align them one by one. Quan et al. [22] present a semi-supervised learning approach to non-monotonic sentence alignment by incorporating both monolingual and bilingual consistency.

Although there have been a lot of studies on sentence alignment of unsupervised and semi-supervised approaches, most of them are based on hand-crafted features. Due to the strong capability of automatically learning feature representations through neural network, the researches on supervised approaches with neural network recently start to emerge. Gregoire and Langlais [7] propose a deep neural network approach at sentence-level to detect translation equivalence between sentences in bi-texts. They use a shared bi-directional recurrent neural network (BiRNN) encoder to encode a sentence into a continuous vector representation, and then estimate aligning probability of a sentence pair by feeding their vectors into a fully connected layer with a sigmoid layer. Grover and Mitra [8] create a similarity matrix between the words of a sentence pair using cosine similarity measure, then dynamically pool the similarity matrix into a fixed-dimension matrix and finally use Convolutional Neural Network (CNN) to estimate if the pair is aligned or not. Similarly, Ding et al. [9] propose a word-pair relevance network to extract parallel sentences at word-level. They use BiRNN to encode a sentence pair, then three similarity measures are adopted to capture the semantic interaction between word pairs, and finally they transform semantic interaction by max pooling into a vector and adopt a multilayer perceptron to predict whether the sentence pair is aligned or not. To our best knowledge, we are not aware of studies that explore external knowledge for neural sentence alignment.

Nevertheless, the approaches of leveraging external knowledge have recently been proposed in neural machine translation (NMT) to improve translation quality through different perspectives. Word translation is an important linguistic resource and is very helpful. Most related studies [13,23–26] focus on decoders to guild NMT decoding model in favor of pre-obtained target words. Syntax is another type of useful external knowledge and has been widely explored in NMT, including both source-side [27–30] and target-side [31,32]. Our approach–*Mixed Encoder*, *Factored Encoder* and *Gated Encoder* are motivated by the models in Li et al. [28], Sennrich and Haddow [30], Han et al. [33], but we focus on introducing word translation rather than syntax into encoder of sentence alignment model.

## 3 Problem definition

Sentence alignment accepts a bitext consisting of a set of source language sentences $X = \{\mathbf{x}^1, \ldots, \mathbf{x}^i, \ldots, \mathbf{x}^M\}$, and a set of target language sentences $Y = \{\mathbf{y}^1, \ldots, \mathbf{y}^j, \ldots, \mathbf{y}^N\}$ as input. Monotonic alignment follows the monotonicity assumption that aligned sentences in bitexts appear in a similar sequential order in two languages without crossings in general [34]. On the contrast, non-monotonic alignment allows the sentence pairs in $X$ and $Y$ to cross arbitrarily. Figure 2(a) illustrates a monotonic alignment with no crossing correspondences in the bipartite graph while Fig. 2(b) has non-monotonic alignment with scrambled pairs. In non-monotonic alignment, we can find that the type of many-to-many alignment is much more complicated, so we will not consider this type but assume that each sentence may align to only one or zero sentence in the other language, i.e., 1-1 and 1-0/0-1.

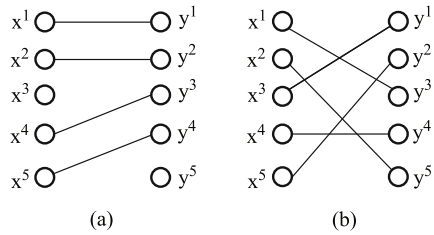


**Fig. 2** Illustration of (a) monotonic and (b) non-monotonic alignment, with a line correspondence between two bilingual sentence

For monotonic alignment, it is relatively straightforward to identify all types of alignments using the dynamic programming algorithms [20], even for many-to-many. We compute a lattice $F(i, j)$ representing the similar score from the beginning of the document to the $i$th source sentence and $j$th target sentence, then the lattice can be calculated using a recurrence relation as follow:

$$F(i,j) = \max \begin{cases} F(i-1,j) + sim(i,\phi) \\ F(i,j-1) + sim(\phi,j) \\ F(i-1,j-1) + sim(i,j) \\ F(i-1,j-2) + sim(i,j-1) \\ F(i-2,j-1) + sim(i-1,j) \\ F(i-1,j-3) + sim(i,j-2) \\ F(i-3,j-1) + sim(i-2,j) \end{cases} \quad (1)$$

where *sim* is the probability provided by our model, and the rows in the Eq. (1) correspond to 1-0, 0-1, 1-1, 1-2, 2-1, 1-3 and 3-1 alignment, respectively. Specifically, $\phi$ in $sim(i, \phi)$ denotes *NULL*, i.e., the $i$th source sentence is not aligned with any target sentence and we simply set $sim(i, \phi) = -0.01$ as Champollion [20]. Note that for NIST MT monotonic test sets, we only extract 1-1 and 1-0/0-1 alignments.

Then, for non-monotonic alignment, let matrix $F \in \mathbf{R}^{M \times N}$ represents the correspondence relation between $X$ and $Y$, where $F_{ij}$ is a real score to measure the likelihood of matching the $i$th sentence $\mathbf{x}^i$ in $X$ against the $j$th sentence $\mathbf{y}^j$ in $Y$, i.e., the probability of $\mathbf{x}^i$ and $\mathbf{y}^j$ being aligned. And alignment matrix $A \in \{0,1\}^{M \times N}$ is defined to produce the final alignment where $A_{ij} = 1$ for a correspondence between $\mathbf{x}^i$ and $\mathbf{y}^j$ and $A_{ij} = 0$ otherwise, then we use a heuristic search for local optimization [35], which consists of two steps:

- Pick $F_{ij}$ such that $F_{ij} \geqslant 0.5$ and is ranked the greatest score in the similarity matrix $F$. Set $A_{ij}$ to 1, and $F_{i^*,j^*}$ $(1 \leqslant i^* \leqslant M, 1 \leqslant j^* \leqslant N)$ to 0 since $\mathbf{x}^i$ and $\mathbf{y}^j$ are aligned.
- Repeat above step until all entries in $F$ are less than 0.5.

On the basis, given the alignment matrix $A$, it is easy to obtain all 1-0/0-1, and 1-1 alignments from it.

Given a sentence pair $(\mathbf{x}^i, \mathbf{y}^j)$, in next section we describe our neural sentence alignment model which returns the probability of the sentence pair being aligned, i.e., $sim(i, j)$ score for monotonic alignment, and $F_{ij}$ score for non-monotonic alignment.

## 4 Neural sentence alignment with word translation

In this section, we will first describe our method to obtain word translation, then present the neural sentence alignment model, and finally detail our approaches that incorporate word translation.

### 4.1 Learning word translation

For each source word $x_i$, we simply obtain its translation $d_i$ from a bilingual dictionary by looking for its translation with the highest probability, as defined below:

$$d_i = \arg\min_{w \in V_t} P_{dic}(w|x_i), \tag{2}$$

where $V_t$ is the vocabulary of the target language and $P_{dic}(w|x_i)$ is the lexical translation probability from source word $x_i$ to target word $w$. And each target word is processed in the same way.

To obtain the bilingual dictionary, we get word alignment results by running Giza++ [36] on the two training data in two directions (source → target, target → source), respectively. Then word translation probability could be computed from word alignments. Specifically, a special token *NULL* indicates the word does not have a corresponding translation.

### 4.2 Neural sentence alignment

We use the neural sentence alignment model proposed in Ding et al. [9] as our baseline. Given a sentence pair, the baseline uses two bi-directional RNNs to encode the pair, one for the source sentence and the other for the target one.[1)] In order to explicitly incorporate word translation, we propose three different cross-lingual encoders while keep the rest part of the model unchanged. Figure 3 shows the architecture of the renewed model which consists of the following main sublayers:

- Cross-lingual Encoder: models input sentence pair and their corresponding word translation sequences. The encoder serves as the basis of the subsequent network sublayers (see details in Section 4.3).
- Word-pair relevant networks (WPRN): captures the relevance score for every word pair $(x_i, y_j)$ through its hidden state pair $(h_{x_i}, h_{y_j})$ from different perspectives. Specially, we measure the pair's relevance scores from following multiple views:
  - Cosine similarity: $\cos(h_{x_i}, h_{y_j})$ is defined as:

$$\cos(h_{x_i}, h_{y_j}) = \frac{h_{x_i} \cdot h_{y_j}}{||h_{x_i}||||h_{y_j}||}. \tag{3}$$

    The cosine similarity measures the similarity of two representations with the angle between them.
  - Bilinear model: $b(h_{x_i}, h_{y_j})$ is defined as:

$$b(h_{x_i}, h_{y_j}) = {h_{x_i}}^{\mathrm{T}} M h_{y_j}, \tag{4}$$

    where $M \in \mathbf{R}^{d_h \times d_h}$ is a weight matrix. The bilinear model is a simple but efficient way to incorporate the strong linear interactions between two representations [38, 39].
  - Single Layer Network: $s(h_{x_i}, h_{y_j})$ is defined as:

$$s(h_{x_i}, h_{y_j}) = u^{\mathrm{T}} f(V[h_{x_i}, h_{y_j}] + b), \tag{5}$$

    where $u \in \mathbf{R}^k$, $V \in \mathbf{R}^{k \times 2d_h}$, $b \in \mathbf{R}^k$ are parameters to be learned, and $f$ is a non-linear function applied element-wise, e.g., *tanh* in this paper. $k$ is a hyper-parameter we can set arbitrarily. The single layer network captures the nonlinear interactions between two representations [40].

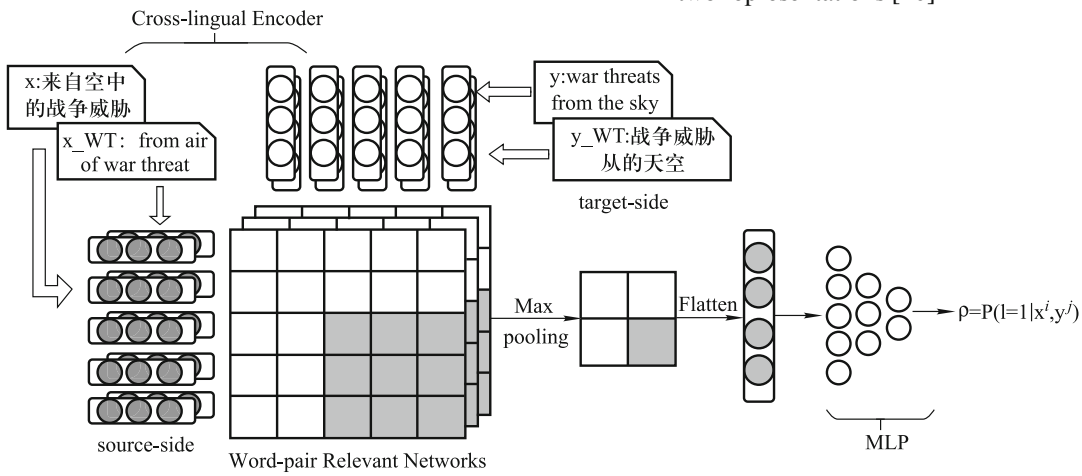


**Fig. 3** The architecture of the proposed approach. "*_WT" denotes the corresponding word translations

[1)] For the activation function of an RNN, in this paper we uses the gated recurrent unit (GRU) proposed by Cho et al. [37]

- Max pooling: adopts a max-pooling strategy to partition the similarity score matrix generated by WPRN, into a set of non-overlapping sub-regions, each of which outputs the maximum value.[2] Assuming that the max pooling size is $3 \times k_1 \times k_2$, the output is thus a matrix with the size of $\mathbf{R}^{\lceil \frac{m}{k_1} \rceil \times \lceil \frac{n}{k_2} \rceil}$, and then we flatten the matrix of pooling scores into a vector[3].
- Multi-layer perceptron (MLP): uses two successive full connection hidden layers to get a more abstractive representation and then connect to the output layer. For the task of classification, the outputs are probabilities of binary classes, computed by a *sigmoid* function after the fully-connected layer.

### 4.3 Cross-lingual Encoders

Most conventional neural network models regard a sentence as a sequence of words, which easily ignores external knowledge and fails to effectively capture various useful information.

To leverage external knowledge, especially the use of word translation, we propose three cross-lingual encoders to model the input words and their translations from multiple ways. For simplicity, we take a source sentence and its word translation sequence as an example to illustrate the encoders while a target sentence and its word translation sequence can be encoded in the same way. Given a source sentence $\mathbf{x} = (x_1, \ldots, x_i, \ldots, x_m)$ and its word translation sequence $\mathbf{d} = (d_1, \ldots, d_i, \ldots, d_m)$, the goal is to inform the model of each word and its potential translation so that it not only encodes the information of words (e.g., $x_i$) and their surroundings, but also encodes their possible translations (e.g., $d_i$). In the following, we will present the details of these cross-lingual encodes (Sections 4.3.1–4.3.3) and Figs. 4–6 show their structures. Among them, $x_i$ and $d_i$ represent source word and its possible translation, $Ex_i$ and $Ed_i$ represent corresponding embedding, $\overrightarrow{h_i}$ and $\overleftarrow{h_i}$ represent the forward hidden state and the backward hidden state of the $i$th word, respectively.

#### 4.3.1 Mixed Encoder

Inspired by the model proposed by Li et al. [28], Fig. 4 illustrates the structure of *Mixed Encoder* which encode source words and their translations in a mixed way. The input sequence is alternatively mixed with both source words and their translation, i.e., $(x_1, d_1, \ldots, x_i, d_i, \ldots, x_m, d_m)$, and only the annotation vectors of source words, i.e., $([\overrightarrow{h_{x_1}}, \overleftarrow{h_{x_1}}], \ldots, [\overrightarrow{h_{x_i}}, \overleftarrow{h_{x_i}}], \ldots, [\overrightarrow{h_{x_m}}, \overleftarrow{h_{x_m}}])$ are fed to the next layer. Even though the annotation vectors of translations (e.g., $[\overrightarrow{h_{d_i}}, \overleftarrow{h_{d_i}}]$) are not directly fed to the next layer, the error signal is back propagated along the word sequence and allows the annotation vectors of word translation being updated accordingly.

#### 4.3.2 Factored Encoder

Similar to the related studies which integrate source-side features into encoder [30], we regard word translation $d_i$ as an external feature of source word $x_i$ and directly concatenate their corresponding word embeddings $Ex_i$ and $Ed_i$. As shown in Fig. 5, *Factored Encoder* encodes a source word and its translation vertically, treating them equally. Then, the encoder reads the concatenated word embedding sequence as input.

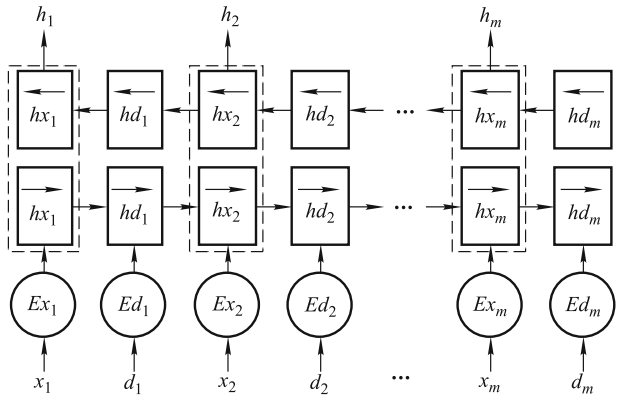


**Fig. 4** Mixed Encoder

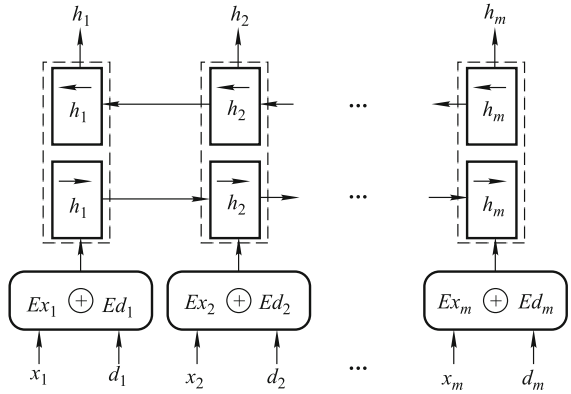


**Fig. 5** Factored Encoder

#### 4.3.3 Gated Encoder

Both encoders mentioned above make full use of word translation $d_i$, while rather than fully utilizing word translation, the *Gated Encoder* selectively controls the amount of word translation fed to the encoder. As shown in Fig. 6, $Ex'_i$ represents the word embedding of the input to encoder as position $i$ and is defined as:

$$Ex'_i = Ex_i + g \circ Ed_i, \tag{6}$$

where $\circ$ is an element wise multiplication, $g$ is gate expressing how much the amount of translation word embedding $Ed_i$ should be incorporated into the encoder. We define $g$ as follows:

$$g = \sigma(W_x Ex_i + W_d Ed_i + b), \tag{7}$$

where $W_x \in \mathbf{R}^{1\times d}$, $W_d \in \mathbf{R}^{1\times d}$ and $b \in \mathbf{R}$ are parameters to be learned, $\sigma$ denotes the logistic sigmoid function. *Gated Encoder* selectively incorporates the amount of translation's information through the gated mechanism, which is potentially helpful in cases where word translation might be incorrect.

[2] We also tried other pooling strategies, like average pooling and sum pooling. However, our preliminary experimental results show that the max-pooling strategy outperforms the others on the development set

[3] Note that in max pooling, the pooling size's first dimension (e.g., 3 in this paper) is the number of similarity measures

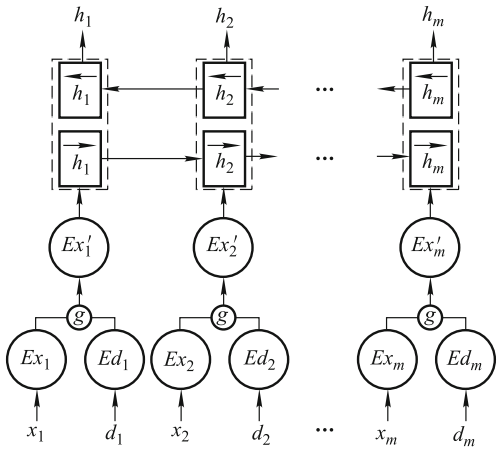

**Fig. 6** Gated Encoder

### 4.4 Training

Given training sentence pairs $(X, Y) = \{\mathbf{x}^i, \mathbf{y}^i | 1 \leqslant i \leqslant N\}$ and their true labels $L = \{l^i | 1 \leqslant i \leqslant N, l^i \in \{0, 1\}\}$, the training objective is to minimize the cross entropy, defined as:

$$L(X, Y, L; \Theta) = \sum_{i=1}^{N} \left( l^i \log(\hat{p}^i) + (1 - l^i) \log(1 - \hat{p}^i) \right), \qquad (8)$$

where $\hat{p}^i$ is the predicted probability of label 1 for sentence pair $(\mathbf{x}^i, \mathbf{y}^i)$.

## 5 Experimentation

To our best knowledge, there is no official corpus for sentence alignment. Therefore, we evaluate our approaches on NIST MT Chinese-English translation dataset and Opensubtitles 2018 Chinese-English dataset.

### 5.1 Experimental settings

#### 5.1.1 Datasets

The NIST MT training set consists of 1.25M sentence pairs extracted from LDC corpora[4)], with 27.9M Chinese words and 34.5M English words respectively. All the parallel sentence pairs are naturally viewed as positive samples in training. Besides, we also construct negative examples of the same size. That to say, for each source sentence, we randomly choose a target sentence from the target side and obtain a negative example. We use NIST MT 02 (878 sentence pairs) as development set and NIST MT 03, 04 and 05 (919, 1788 and 1082 sentence pairs) as test sets. Note that the source and target sentences in the above dataset are originally 1-1 mapping. In order to obtain 1-0/0-1 alignments, we randomly delete 90 sentences on the source side and 60 sentences on the target side[5)]. Consequently, the bilingual texts are aligned in monotonic way. In addition, sentence order in the above sets is scrambled to obtain NIST MT non-monotonic test sets. Table 1 shows the statistics of the NIST MT sets.

The Opensubtitles training set consists of 1,400 documents randomly selected from Opensubtitles2018, including 1.11M sentence pairs, 15.3M Chinese words and 19.3M English words. Similarly, for every source sentence we randomly choose a target sentence from the target side of the same document and obtain a negative example. The Opensubtitles development set (OSD) and test sets (OST) are also randomly selected from Opensubtitles2018, including 1 document and 8 documents respectively, not included in the training set. Since the test sets contain 1-0/0-1, 1-1, 1-2/2-1 and 1-3/3-1 alignments and the non-monotonic alignment of them is much more complicated, we only evaluated as monotonic alignment for simplicity. Table 2 shows the statistics of the Opensubtitles sets.

**Table 1** Numbers of sentences and alignments on NIST MT development set and test set

| Dataset | #Src | #Trg | 1-0/0-1 | 1-1 |
|---|---|---|---|---|
| nist02 | 788 | 818 | 138 | 734 |
| nist03 | 829 | 859 | 144 | 772 |
| nist04 | 1698 | 1728 | 146 | 1640 |
| nist05 | 992 | 1022 | 140 | 937 |
| All (Test) | 3519 | 3609 | 430 | 3349 |

**Table 2** Numbers of sentences and alignments on Opensubtitles development set and test set

| Dataset | #Src | #Tgt | 1-0/0-1 | 1-1 | 1-2/2-1 | 1-3/3-1 |
|---|---|---|---|---|---|---|
| OSD | 170 | 180 | 24 | 121 | 24 | 3 |
| OST | 4225 | 3806 | 207 | 2840 | 576 | 104 |

#### 5.1.2 Parameters

In order to train the neural network models effectively, we limit the maximum sentence length of source and target side to 50. And the most frequency of 30K words in Chinese and English are selected as the source and target vocabularies, covering approximately 98.4% and 99.0% of NIST MT source and target training sentences respectively, and 96.3% and 99.0% of Opensubtitles training set. All the out-of-vocabulary words are mapped to a special token *UNK*.

We use pre-trained 50-dimensional Chinese-English bilingual word embeddings provided by Zou et al. [41], and update them in training process. In the cross-lingual encoder layer, we use GRU for the activation function and set the size of hidden state as 150. In the WPRN layer, we set $k$ in single layer network as 2. Consequently, the WPRN layer will obtain tensors with size $3 \times 50 \times 50$. In the max pooling layer, the max-pooling size is the set $3 \times 3 \times 3$, resulting the flatten vector size as 289 (i.e., $\lceil \frac{50}{3} \rceil * \lceil \frac{50}{3} \rceil$). In the MLP layer, the sizes of the neighboring hidden sub-layers are 2000 and 1000, respectively.

#### 5.1.3 Evaluation

We adopt precision (P), recall (R) and F-measure (F1) as evaluation metrics to evaluate each type of alignments, as well as Micro-averaged scores of precision, recall and F-measure (Micro-P/R/F1) to measure the overall performance on all alignments.

### 5.2 Experimental results

#### 5.2.1 Results on non-monotonic alignment

Table 3 shows the performance of NIST MT test set on non-monotonic alignment. From this table, we have the following observations:

[4)] The dataset includes LDC2002E18, LDC2003E07, LDC2003E14, Hansards portion of LDC2004T07, LDC2004T08 and LDC2005T06

[5)] The deleted sentences number can be set as arbitrary number, just to increase the alignment of 1-0/0-1

- Word translation is helpful for sentence alignment. Clearly, the three cross-lingual encoders, *Mixed Encoder*, *Factored Encoder* and *Gated Encoder*, significantly improve all type of alignments. For instances, our best approach *Gated Encoder* outperforms the baseline with 3.2 F1 scores over all alignments.
- The three cross-lingual encoders behave differently. *Gated Encoder* achieves the biggest improvement, followed by *Mixed Encoder* and *Factored Encoder*.
- Comparing to 1-1 alignment, the lower performance of 1-0/0-1 alignment provides more room for improvement. For example, *Gated Encoder* achieves 9.1 F1 scores improvement on 1-0/0-1 alignment over the baseline, much bigger than the 2.2 F1 scores improvement on 1-1 alignment.

**Table 3** The non-monotonic sentence alignment performance on the NIST MT test set

| | 1-0/0-1 | | | 1-1 | | | Micro | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| Baseline | 66.4 | 83.0 | 73.8 | 97.0 | 95.4 | 96.2 | 92.7 | 94.0 | 93.3 |
| Factored | 68.5 | 89.5 | 77.6 | 98.0 | 96.1 | 97.0 | 94.6 | 96.3 | 95.4 |
| Mixed | 76.2 | 89.3 | 82.2 | 98.8 | 97.7 | 98.2 | 95.8 | 96.7 | 96.3 |
| Gated | **76.7** | **90.2** | **82.9** | **98.9** | **97.8** | **98.4** | **96.0** | **96.9** | **96.5** |

### 5.2.2 Results on monotonic alignment

Most sentence aligners are designed for monotonic alignment. Here we also compare our approaches against three popular traditional aligners as follows:

- Moore [17]: A fast and accurate sentence aligner for bilingual corpora which combines sentence-length-based method and word-correspondence-based method.
- Gargantua [19]: An unsupervised and language-pair independent aligner of symmetrical and asymmetrical parallel corpora.
- Champollion [20]: A lexicon-based sentence aligner designed for robust alignment of potential noisy parallel text.

Table 4 presents the monotonic alignment performance on NIST MT test set. Although all the traditional aligners obtain favorable performance, our baseline achieves as high as 99.3 F1 score, clearly better than the traditional aligners, suggesting supervised learning is necessary to achieve high performance. We also note that the almost perfect performance of the baseline gives no space for cross-lingual encoders to achieve further improvement.

**Table 4** The monotonic sentence alignment performance on the NIST MT test set

| | 1-0/0-1 | | | 1-1 | | | Micro | | |
|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 |
| Baseline | **100.0** | **92.1** | **95.9** | **99.5** | **100.0** | **99.8** | **99.6** | **99.1** | **99.3** |
| Factored | 99.8 | 91.8 | 95.6 | 99.5 | 99.9 | 99.7 | 99.5 | 99.0 | 99.2 |
| Mixed | **100.0** | **92.1** | **95.9** | **99.5** | **100.0** | **99.8** | **99.6** | **99.1** | **99.3** |
| Gated | **100.0** | **92.1** | **95.9** | **99.5** | **100.0** | **99.8** | **99.6** | **99.1** | **99.3** |
| Moore | 53.8 | 89.3 | 67.1 | 98.8 | 94.6 | 96.6 | 90.6 | 94.0 | 92.3 |
| Gargantua | 43.5 | 79.8 | 56.3 | 97.0 | 91.9 | 94.4 | 86.4 | 90.5 | 88.4 |
| Champollion | 32.7 | 59.5 | 42.2 | 91.1 | 86.3 | 88.7 | 79.6 | 83.3 | 81.4 |

Fortunately, the priority of our cross-lingual encoders also persists when they are evaluated on the out-domain and realistic Opensubtitles test set, as shown in Table 5.

From Tables 4 and 5, we observe that:

- Sentence alignment on the Opensubtitles is more challenging than that on the NIST test set. This is not surprising since the former originates from dialogue while the latter originates from news.
- For monotonic alignment, the advantage of cross-lingual encoders over the baseline is shrunken, given the fact that monotonic alignment is less challenging to non-monotonic one.
- Among different types of alignment, 1-0/0-1 alignments are the hardest to recognize, followed by 1-2/2-1 and 3-1/1-3 alignments while 1-1 alignments are the easiest.

## 5.3 Experimental analysis

In this subsection, we analyze the influence of word translation on sentence alignment from different perspectives.

### 5.3.1 Comparison of encoders

The proposed three cross-lingual encoders shown in Figs. 4–6 integrate the information of word translation. Here we compare the proposed encoders as well as the conventional RNN encoder in following three perspectives.

From parameter perspective, *Mixed Encoder* does not require any additional parameters since although its input sentence length becomes twice longer, the encoder parameters are shared to convert both $Ex_i$ to $h_{x_i}$ and $Ed_i$ to $h_{d_i}$. And *Factored Encoder* simply concatenates $Ex_i$ and $Ed_i$ that only requires an extra parameter matrix $W_d \in \mathbf{R}^{d\times h}$ to convert $d$-dimension $Ed_i$ into $h$-dimension $h_{d_i}$. While for *Gated Encoder*, due to the use

**Table 5** The monotonic sentence alignment performance on the Opensubtitles test set

| | 1-0/0-1 | | | 1-1 | | | 1-2/2-1 | | | 1-3/3-1 | | | Micro | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Baseline | 36.5 | 41.1 | 38.6 | 91.0 | 92.1 | 91.6 | 69.7 | 68.2 | 68.9 | 69.3 | 58.7 | 63.5 | 83.9 | 84.7 | 84.3 |
| Factored | 42.4 | 40.6 | 41.5 | 90.8 | 91.8 | 91.3 | 70.4 | 69.1 | 69.8 | 68.7 | 65.4 | 67.0 | 84.6 | 84.7 | 84.7 |
| Mixed | **48.6** | 43.5 | 45.9 | **91.6** | 92.7 | 92.1 | 70.9 | 71.2 | 71.1 | 71.4 | **72.1** | **71.8** | 85.7 | 86.1 | 85.9 |
| Gated | 46.2 | **50.2** | **48.1** | 91.4 | **93.0** | **92.2** | **73.7** | **71.2** | **72.4** | **75.6** | 65.4 | 70.1 | **85.7** | **86.5** | **86.1** |
| Moore | 5.0 | 8.1 | 9.5 | 76.2 | 65.6 | 70.5 | - | - | - | - | - | - | 35.1 | 54.5 | 42.7 |
| Gargantua | 8.9 | 27.1 | 13.4 | 67.8 | 75.6 | 71.5 | 56.1 | 30.2 | 39.3 | 61.3 | 18.3 | 28.1 | 57.9 | 64.3 | 61.0 |
| Champollion | 3.2 | 5.3 | 4.0 | 52.2 | 48.3 | 50.2 | 20.7 | 8.7 | 12.2 | 8.8 | 12.5 | 10.4 | 40.3 | 38.8 | 39.5 |

of gate mechanism shown in Eq. (7), it requires extra $(2 * d + 1)$ parameters.

From the network perspective, the degrees of making use of word translation in the three encoders are like this: *Factored Encoder* > *Mixed Encoder* > *Gated Encoder*. Given a word sequence $\mathbf{x} = (x_1, \ldots, x_i, \ldots, x_m)$ and its word translation sequence $\mathbf{d} = (d_1, \ldots, d_i, \ldots, d_m)$, our goal is to learn word annotation vectors $\mathbf{h} = (h_1, \ldots, h_i, \ldots, h_m)$ which encode not only the information of words and their surroundings, but also translation information. *Factored Encoder* makes full use of word translation sequence $d$ and treat it equally as word sequence $x$ while *Mixed Encoder* learns word annotation $h$ with context containing word translation sequence. Differently, *Gated Encoder* uses gate mechanism to selectively control the amount of word translation information moving forward into the encoder. This is important to alleviate the impact of error propagation from word translation.

From the flexibility perspective, *Mixed Encoder* is more flexible and extensible than the other two. In the *Factored Encoder* or *Gated Encoder*, there must be one and only one translation word associated to every source word. Therefore, it would be hard to model other scenarios with phrase translations, such as 纽约 to *New York*, 女 发言人 to *spokeswoman*, and so on. However, in the *Mixed Encoder*, original words and translation words are sequenced and they are also applied to scenarios when a source word has no word translation or has multiple translation words. Therefore, phrase translations can be naturally handled by the *Mixed Encoder* to sequence the source phrase and its translation, e.g., 女 发言人 *spokeswoman*.[6] Moreover, rather than covering all source words, the *Mixed Encoder* is extensible to focus on source words with special interest (e.g., low frequency words, named entities).

### 5.3.2 Accuracy of word translation

In this paper, word translation is used as external knowledge to enhance the word semantic modeling. Normally, correct word translation can enhance the semantic information while incorrect one may affect it. While it is hard to precisely estimate the accuracy of word translation, we access the accuracy from word alignment perspective. Given a sentence pair $(x, y)$ and word $x_i$ in $x$, we obtain its translation from the word alignment to sentence $y$. Specifically, if $x_i$ aligns to multiple target words, we select the one with the highest lexical translation probability as $x_i$'s *gold* translation. Therefore, we are able to estimate if the given $x_i$'s word translation is correct or not. On the NIST MT development set, the accuracy of Chinese word translation is 52.1%, while that of English is 49.3%. Note that the evaluation criterion is very strict and will view two words of semantic equivalence as wrong translation to each other. As shown in Table 6, the word translation accuracy of Chinese and English are 45.4% and 70.0% (i.e, 5/11 and 7/10), respectively. However, wife-(妻子, 夫人), children-(孩子, 儿童), 是-(are, is) are all polysemous pairs.

### 5.3.3 Analysis on opensubtitles

Sentence alignment on Opensubtitles is more difficult than the NIST dataset, Table 7 presents several typical examples to illustrate the challenges.

**Table 6** Example of aligned sentences $(Src, Trg)$ and their word translations $(Src\_WT, Trg\_WT)$. Here, semantically equivalent words are highlighted in the same color

| | |
|---|---|
| Src | 他的妻子 和两个孩子是美国公民。 |
| Trg | his wife and two children are all us citizens. |
| Src_WT | he of wife and two NULL children is us citizens. |
| Trg_WT | 他夫人 及两儿童是所有美国市民。 |

**Table 7** Example of aligned sentences $(Src, Trg)$ extracted from Opensubtitiles2018 test sets while our approaches fail to recognize them. The literal translations are provided below each $Src$ sentence for easy understanding

| | |
|---|---|
| Src | 你好 。 |
| | hello . |
| Trg | good evening . |
| Prob | Gated: 0.14 Mixed: 0.21 Factored: 0.10 |
| | (a) |
| Src | 那就好 。 |
| | that's_good . |
| Trg | all right, then . |
| Prob | Gated: 0.24 Mixed: 0.36 Factored: 0.47 |
| | (b) |
| Src | 真是 省话 一哥 |
| | really save words brother |
| Trg | never one to waste words |
| Prob | Gated: 0.39 Mixed: 0.38 Factored: 0.23 |
| | (c) |

Firstly, the document-level information is important for Opensubtitle's sentence alignment, because all sentences come from conversations and document information can provide context to assist alignment. In the first example (a), the target sentence provides the temporal background, and the surface source meaning is "hello" which is different from the target. Lacking of the document context is one of reasons that our approaches recognize it incorrectly. Secondly, the unknown words are one of the sources of sentence alignment errors, especially for Opensubtitlie dataset whose sentences are usually short in length. Take the second sentence pair (b) as example, "那就好" is an unknown word in vocabulary and there are no other useful words in source sentence, which makes the prediction quite difficult. Finally, it is challenging to capture the correct semantic of proverbs, poems or four-characters words in Chinese. Like "省话一哥" in the last example (c) is a popular phrase in Chinese.

### 5.3.4 Cross domain evaluation

Experimental results in Section 2 demonstrate that our proposed encoders significantly improve sentence alignment on the test sets which are in the same domain of the training sets. To see if the encoders benefit test sets of a different domain, we evaluate the encoders trained on the NIST MT data against the Opensubtitles test set.

Table 9 shows the monotonic sentence alignment performance achieved by the encoders trained on the NIST MT data. From it, we observe that except the *Factored Encoder*, the other two encoders also benefit test set of another domain. It is also not surprising that the results in Table 9 are lower that the coun-

[6] For an aligned phrase pair containing multiple words on two sides, it can also be heuristically converted into a sequence

**Table 8** Example of sentence alignment predictions. Note that the first two sentence pairs are aligned while the last one is unaligned. "Src/Trg_WT" indicates that the word translations of the source/target sentence. "Prob" indicates the predicted probabilities of *Src* and *Trg* being aligned by different models

| | |
|---|---|
| Src1 | 欧盟 轮值 主席 荷兰 在 中国 \| 欧盟 高峰 会后, 作 以上 表示。 |
| Src_WT1 | eu follows chairman netherlands in china ? eu peak meeting, for over said. |
| Trg1 | the eu dutch presidency made the remarks following a china-eu summit. |
| Trg_WT1 | 的 欧盟 荷兰 总统 了 的 话 下 一 欧 首脑。 |
| Prob1 | Gated: 0.99 Mixed: 0.96 Factored: 0.62 Baseline: 0.26 |
| Src2 | 深圳 将 在 部分 楼宇 和 小区 强制 推行 "中 水" 应用 |
| Src_WT2 | shenzhen will in part building and unit mandatory implementation "of water" application |
| Trg2 | shenzhen to enforce the use of " reclaimed water " in some office buildings and residential quarters |
| Trg_WT2 | 深圳 *NULL* 执行的使用 *NULL* ┌ 填水 ┌ 在一些事务楼宇及住宅方面 |
| Prob2 | Gated: 0.98 Mixed: 0.95 Factored: 0.73 Baseline: 0.72 |
| Src3 | 中方的基本出发点是维护朝鲜半岛的和平与稳定, 实现半岛无核化。 |
| Src_WT3 | side of basic starting is safeguarding dprk peninsula of peace with stability, achieve peninsula no nuclear to. |
| Trg3 | national reconciliation is an important condition for sustained peace and stability in a region after conflicts. |
| Trg_WT3 | 国家和解是一重要情况为持续和平及稳定在一地区后冲突。 |
| Prob3 | Gated: 0.03 Mixed: 0.05 Factored: 0.41 Baseline: 0.55 |

**Table 9** The monotonic sentence alignment performance on the Opensubtitles test set when the encoders are trained on the NIST MT data

| | 1-0/0-1 | | | 1-1 | | | 1-2/2-1 | | | 1-3/3-1 | | | Micro | | |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 | P | R | F1 |
| Baseline | 23.8 | 39.1 | 29.6 | 84.8 | 86.1 | 85.4 | 59.9 | 55.0 | 57.4 | 64.3 | 51.9 | 57.5 | 75.5 | 77.7 | 76.6 |
| Factored | 23.3 | 31.9 | 26.9 | 85.6 | 83.5 | 84.6 | 54.4 | 55.9 | 55.1 | 51.9 | 53.9 | 52.9 | 75.0 | 75.6 | 75.3 |
| Mixed | 25.6 | 38.6 | 30.8 | **86.8** | 86.1 | **86.4** | 57.4 | 56.1 | 56.7 | 60.0 | 57.7 | 58.8 | 76.7 | 78.0 | 77.3 |
| Gated | **28.4** | **40.1** | **33.3** | 85.7 | **86.9** | 86.3 | **62.0** | **58.0** | **60.0** | **66.7** | **57.7** | **61.9** | **77.5** | **79.0** | **78.2** |

terparts in Table 5, suggesting that there exists improvement room for cross domain sentence alignment.

#### 5.3.5 Case study

In Table 8, we list three sentence pairs from test sets as examples to illustrate the advantages of our approaches. The first two examples are aligned sentence pairs and the last is unaligned.

For the sentence pair (*Src1, Trg1*), the baseline incorrectly predicts it as unaligned with a low probability while our approaches all give this pair with high probabilities (i.e., 0.99, 0.96 and 0.62). Even containing semantically equivalent pairs, like (欧盟, eu), (荷兰, dutch), the baseline still fails to recognize them. However, with the help of word translations, our approaches correctly predict the the sentence pair as aligned. Although all approaches correctly predict the second sentence pair (*Src2, Trg2*) as aligned, they give considerable distinguishing probabilities, with *Gated Encoder* gives the highest confidence, followed by *Mixed Encoder* and *Factored Encoder*. The third sentence pair (*Src3, Trg3*) is unaligned, despite of the existence of several semantically equivalent word pairs, such as (和平, peace), (和, and) and (稳定, stability). For this pair, the baseline mistakenly predicts it as aligned while our approaches, especially the *Gated Encoder* is almost 97% confident that this pair is unaligned.

Analysis of above examples shows that word translation is helpful to improve the performance of sentence alignment. On the one hand, our three cross-lingual encoders are more likely to make correct prediction than the baseline. On the other hand, *Gated Encoder* obtains the most reasonable alignment probabilities, followed by *Mixed Encoder* and *Factored Encoder*.

## 6 Conclusion and future work

In this paper, we explore approaches of incorporating word translation into neural sentence alignment models. Specifically, we propose three cross-lingual encoders to capture translation information. Among them, *Mixed Encoder* and *Factored Encoder* treat words equally with their translations and respectively model word translation in horizontal and vertical way while *Gated Encoder* automatically controls the amount of translation information through the gate mechanism. Experimental results on NIST MT and Opensubtitles Chinese-English datasets shows that word translation is useful for sentence alignment and the proposed cross-lingual encoders yield improvements over strong baseline.

For future work, we will further explore other external resources, such as part of speech and syntax, to improve sentence alignment performance. Given that many language pairs are of low-resource, we will test our approach on them and examine whether the sentence alignment quality is good enough if using small scale training data to learn bilingual dictionary.

**Acknowledgements** We thank the anonymous reviewers for their insightful comments and suggestions. This work was supported by the National Natural Science Foundation of China (Grant Nos. 61876120, 61673290).

## References

1. Bahdanau D, Cho K, Bengio Y. Neural machine translation by jointly learning to align and translate. In: Proceedings of International Conference on Learning Representations. 2015
2. Vaswani A, Shazeer N, Parmar N, Uszkoreit J, Jones L, Gomez A, Kaiser L, Polosukhin I. Attention is all you need. In: Proceedings of the 31st Conference on Neural Information Processing Systems. 2017, 6000–6010
3. Hermann K M, Blunsom P. Multilingual models for compositional dis-

tributed semantics. In: Proceedings of the 52nd Annual Meeting of the Association for Computational Linguistics. 2014, 58–68

4. Nie J Y, Simard M, Isabelle P, Durand R. Cross-language information retrieval based on parallel texts and automatic mining of parallel texts from the web. In: Proceedings of the 22nd Annual International ACM SIGIR Conference on Research and Development in Information Retrieval. 1999, 74–81
5. Martino G D S, Romeo S, Barroón-Cedeno A, Joty S, Marquez L, Moschitti A, Nakov P. Cross-language question re-ranking. In: Proceedings of the 40th International ACM SIGIR Conference on Research and Development in Information Retrieval. 2017, 1145–1148
6. Wu D. Alignment. Handbook of Natural Language Processing. CRC Press. 2010
7. Gregoire F, Langlais P. A deep neural network approach to parallel sentence extraction. 2017, arXiv preprint arXiv:1709.09783
8. Grover J, Mitra P. Bilingual word embeddings with bucketed CNN for parallel sentence extraction. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics–Student Research Workshop. 2017, 11–16
9. Ding Y, Li J, Zhou G. Word-pair relevance network for sentence alignment. Journal of Chinese Information Processing, 2019
10. Ding Y, Li J H, Gong Z X, Zhou G D. Word-pair relevance modeling with multi-view neural attention mechanism for sentence alignment. Journal of Computer Science and Technology, 2019
11. Liu L, Utiyama M, Finch A, Sumita E. Neural machine translation with supervised attention. In: Proceedings of the 26th International Conference on Computational Linguistics: Technical Papers. 2016, 3093–3102
12. Mi H, Wang Z, Ittycheriah A. Supervised attentions for neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016, 2283–2288
13. Arthur P, Neubig G, Nakamura S. Incorporating discrete translation lexicons into neural machine translation. In: Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing. 2016, 1557–1567
14. Gale W A, Church K W. A program for aligning sentences in bilingual corpora. In: Proceedings of the Meeting on the Association for Computational Linguistics. 1991, 177–184
15. Chen S F. Aligning sentences in bilingual corpora using lexical information. Computer Knowledge & Technology, 1993, 46(3): 9–16
16. Wu D. Aligning a parallel English-Chinese corpus statistically with lexical criteria. Computer Science, 1994, 4(4): 80–87
17. Moore R C. Fast and accurate sentence alignment of bilingual corpora. In: Processing of the 5th Conference of the Association for Machine Translation in the Americas. 2002, 135–144
18. Brown P F, Pietra V J D, Pietra S A D, Mercer R L. The mathematics of statistical machine translation: parameter estimation. Computational linguistics, 1993, 19(2): 263–311
19. Braune F, Fraser A. Improved unsupervised sentence alignment for symmetrical and asymmetrical parallel corpora. In: Processings of the 23rd International Conference on Computational Linguistics. 2010, 81–89
20. Ma X. Champollion: a robust parallel text sentence aligner. In: Processings of the 5th International Conference on Language Resources and Evaluation. 2006, 489–492
21. Li P, Sun M, Xue P. Fast-Champollion: a fast and robust sentence alignment algorithm. In: Proceedings of the 23rd International Conference on Computational Linguistics. 2010, 710–718
22. Quan X, Kit C, Song Y. Non-monotonic sentence alignment via semisupervised learning. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics. 2013, 622–630
23. Chatterjee R, Negri M, Turchi M, Federico M, Specia L, Blain F. Guiding neural machine translation decoding with external knowledge. In: Proceedings of the 2nd Conference on Machine Translation. 2017, 157–168
24. Nguyen T Q, Chiang D. Improving lexical choice in neural machine translation. 2017, arXiv preprint arXiv:1710.01329
25. Wang X, Tu Z, Xiong D, Zhang M. Translating phrases in neural machine translation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017, 1421–1431
26. Wang X, Lu Z, Tu Z, Li H, Xiong D, Zhang M. Neural machine translation advised by statistical machine translation. In: Proceedings of the 31st AAAI Conference on Artificial Intelligence. 2017, 3330–3336
27. Chen K, Wang R, Utiyama M, Liu L, Tamura A, Sumita E, Zhao T. Neural machine translation with source dependency representation. In: Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing. 2017, 2846–2852
28. Li J, Xiong D, Tu Z, Zhu M, Zhang M, Zhou G. Modeling source syntax for neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017, 688–697
29. Eriguchi A, Hashimoto K, Tsuruoka Y. Tree-to-sequence attentional neural machine translation. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics. 2016, 823–833
30. Sennrich R, Haddow B. Linguistic input features improve neural machine translation. In: Proceedings of the 1st Conference on Machine Translation. 2016, 83–91
31. Aharoni R, Goldberg Y. Towards string-to-tree neural machine translation. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Short Papers). 2017, 132–140
32. Chen H, Huang S, Chiang D, Chen J. Improved neural machine translation with a syntax-aware encoder and decoder. In: Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics. 2017, 1936–1945
33. Han D, Li J, Li Y, Zhang M, Zhou G. Explicitly modeling word translations in neural machine translation. ACM Transactions on Asian and Low-Resource Language Information Processing, 2019, 19(1): 1–17
34. Langlais P, Simard M, Veronis J. Methods and practical issues in evaluating alignment techniques. In: Processings of the 36th Annual Meeting of the Association for Computational Linguistics and the 17th International Conference on Computational Linguistics. 1998, 711–717
35. Kit C, Webster J J, Sin K K, Pan H, Li H. Clause alignment for bilingual Hong Kong legal texts: a lexicalbased approach. International Journal of Corpus Linguistics, 2004, 9(1): 29–52
36. Och F J, Ney H. A systematic comparison of various statistical alignment models. Computational Linguistics, 2003, 29(1): 19–51
37. Cho K, van Merrienboer B, Gulcehre C, Bahdanau D, Bougares F, Schwenk H, Bengio Y. Learning phrase representations using RNN encoder-decoder for statistical machine translation. In: Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing. 2014, 1724–1734
38. Sutskever I, Salakhutdinov R, Tenenbaum J B. Modelling relational data using bayesian clustered tensor factorization. In: Proceedings of the 22nd International Conference on Neural Information Processing Systems. 2009, 1821–1828
39. Jenatton R, Roux N L, Bordes A, Obozinski G. A latent factor model for highly multi-relational data. In: Proceedings of the 25th International Conference on Neural Information Processing Systems. 2012, 3167–3175
40. Collobert R, Weston J. A unified architecture for natural language processing: deep neural networks with multitask learning. In: Proceedings of the 25th International Conference on Machine Learning. 2008, 160–167
41. Zou W Y, Socher R, Cer D, Manning C D. Bilingual word embeddings for phrase-based machine translation. In: Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing. 2013, 1393–1398

Ying Ding received her BS degree in computer science from Huaiyin Normal University, China in 2016. She is now a Master student in computer science at Soochow University, China. Her current research interests include natural language processing, machine translation.

Zhengxian Gong received her PhD degree in computer science from Soochow University, China in 2014. She is an associate professor in Soochow University, China. Her main research interests include natural language processing, machine translation.

Junhui Li received his PhD degree in computer science from Soochow University, China in 2010. He is an associate professor in Soochow University, China. His main research interests include natural language processing, machine translation.

Guodong Zhou received his PhD degree in computer science from the National University of Singapore, Singapore in 1999. He is a distinguished professor in Soochow University, China. His research interests include natural language processing, information extraction and machine learning.