Word Position Aware Translation Memory for Neural Machine Translation

Qiuxiang He[1], Guoping Huang[2], Lemao Liu[2], and Li Li[1*]

[1] School of Computer and Information Science, Southwest University, Chongqing, 400715, China
[2] Tencent AI Lab, Tencent, Shenzhen, 518000, China
hqxiang@email.swu.edu.cn, donkeyhuang@tencent.com, lemaoliu@gmail.com, lily@swu.edu.cn

Abstract. The approach based on translation pieces is appealing for neural machine translation with a translation memory (TM), owing to its efficiency in both computation and memory consumption. Unfortunately, it is incapable of capturing sufficient contextual translation leading to a limited translation performance. This paper thereby proposes a simple yet effective approach to address this issue. Its key idea is to employ the word position information from a TM as additional rewards to guide the decoding of neural machine translation (NMT). Experiments on seven tasks show that the proposed approach yields consistent gains particularly for those source sentences whose TM is very similar to themselves, while maintaining similar efficiency to the counterpart of translation pieces.

Keywords: Word position · Translation memory · Neural machine translation.

1 Introduction

A translation memory (TM) provides the most similar source-target sentence pairs to the source sentence to be translated, and it yields more reliable translation results particularly for those matched segments between a TM and the source sentence [9]. Therefore, a TM has been widely used in machine translation systems. For example, various research work has been devoted to integrating TM into statistical machine translation (SMT) [4, 6, 12]. As an evolutional shift from SMT to the advanced neural machine translation (NMT), there are increasingly interests in employing TM information to improve the NMT results.

Li et al. and Farajian et al. proposed a fine tuning approach in [5, 2] to train a sentence-wise local neural model on top of a retrieved TM, which was further used for testing a particular sentence. Despite its appealing performance, the fine-tuning for each testing sentence leads to the low latency in decoding. On the contrary, in [3] and [13], the standard NMT model was augmented by additionally encoding a TM for each testing sentence. The proposed model was trained to optimize for testing all source sentences. Although these approaches

[3, 13] are capable of capturing global context from a TM, its encoding of a TM with neural networks requires intensive computation and considerable memory, because a TM typically encodes much more words than those encoded by a standard NMT model.

Thankfully, a simple approach was proposed in [14], which was efficient in both computation and memory. Rather than employing neural networks for TM encoding, they represent a TM for each sentence as a collection of translation pieces consisting of weighted n-grams in a TM, whose weights are added into NMT probabilities as rewards. Unfortunately, because translation pieces capture very local context in a TM, this approach can not generate good translations when a TM is very similar to the testing sentence: in particular, the translation quality is far away from perfect even if the reference translation of the source sentence is included in the training set as argued by [13].

To address the above issue, this paper proposes a word position aware TM approach which captures more contextual information in a TM while maintaining similar efficiency to [14]. Our intuition is that: when translating a source sentence, if a word y is at the position i of a target sentence in a TM, and the word y should be in the output, then the position of y in the output should be not far away from i.

To put this intuition into practice, we design two types of position rewards according to the normal distribution and then integrate them into NMT with translation pieces. We apply our approach to Transformer, a strong NMT system [11]. Extensive experiments on seven translation tasks demonstrate the proposed method delivers substantial BLEU improvements over Transformer and it further consistently and significantly outperforms the approach in [14] over 1 BLEU score on average, while our running speed is almost the same as that in [14].

2 Background

2.1 NMT

In this paper, we use the state-of-the-art NMT model, Transformer [11], as our baseline. Suppose $\mathbf{x} = \left\langle x_1, \dots, x_{|\mathbf{x}|} \right\rangle$ is a source sentence with length $|\mathbf{x}|$ and $\mathbf{y} = \left\langle y_1, \dots, y_{|\mathbf{y}|} \right\rangle$ is the corresponding target sentence of $\mathbf{x}$ with length $|\mathbf{y}|$. Generally, for a given $\mathbf{x}$, Transformer aims to generate a translation $\mathbf{y}$ according to the conditional probability $P(\mathbf{y}|\mathbf{x})$ defined by neural networks:

$$P(\mathbf{y}|\mathbf{x}) = \prod_{i=1}^{|\mathbf{y}|} P(y_i|\mathbf{y}_{<i}, \mathbf{x}) \tag{1}$$

where $\mathbf{y}_{<i} = \langle y_1, \dots, y_{i-1} \rangle$ denotes a prefix of $\mathbf{y}$ with length $i{-}1$. To expand each factor $P(y_i|\mathbf{y}_{<i}, \mathbf{x})$, Transformer bases on the encoder-decoder framework similar to the standard sequence-to-sequence learning in [1].

More specifically, in encoding x, an encoder is composed of L layers of neural networks. During decoding process, the Transformer is also composed of L layers

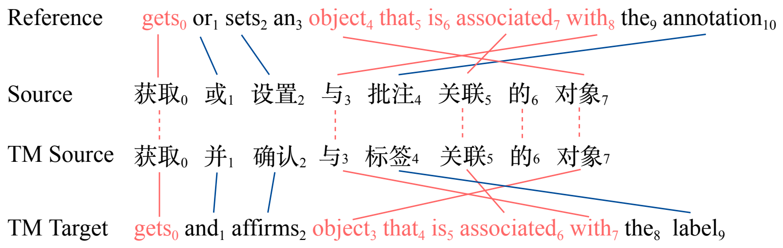
Reference gets$_0$ or$_1$ sets$_2$ an$_3$ object$_4$ that$_5$ is$_6$ associated$_7$ with$_8$ the$_9$ annotation$_{10}$
Source 获取$_0$ 或$_1$ 设置$_2$ 与$_3$ 批注$_4$ 关联$_5$ 的$_6$ 对象$_7$
TM Source 获取$_0$ 并$_1$ 确认$_2$ 与$_3$ 标签$_4$ 关联$_5$ 的$_6$ 对象$_7$
TM Target gets$_0$ and$_1$ affirms$_2$ object$_3$ that$_4$ is$_5$ associated$_6$ with$_7$ the$_8$ label$_9$

Fig. 1. An example of translation pieces in translation memory. The red part is employed to extract translation pieces, such as "gets", "object", "object that", "object that is", "object that is associated" and "that" etc.

of neural networks as mentioned in [11]. The factory $P(y_i|\mathbf{y}_{<i}, \mathbf{x})$ can be defined as following:

$$P(y_i|\mathbf{y}_{<i}, \mathbf{x}) = \mathrm{softmax}\left(\phi(h_i^{D,L})\right) \tag{2}$$

where $h_i^{D,L}$ indicates the i_{th} hidden unit at L_{th} layer under the encoder-decoder framework, and ϕ is a linear network to project the hidden unit to a vector with dimension of the target vocabulary size.

The standard decoding algorithm for NMT is beam search. Namely, at each time step i, we keep n-best hypotheses. The probability of a complete hypothesis is computed as following:

$$\log P(\mathbf{y}|\mathbf{x}) = \sum_{i=1}^{|\mathbf{y}|} \log P(y_i|\mathbf{y}_{<i}, \mathbf{x}) \tag{3}$$

2.2 Translation Pieces

For a source sentence $\mathbf{x}$ to be translated, we use an off-the-shelf search engine to retrieve a set of source sentences along with corresponding translations from translation memory (TM), and then get the TM list $\{(\mathbf{x}^m, \mathbf{y}^m)|m \in [1, M]\}$. Then, we calculate the similarity between $\mathbf{x}$ and $\mathbf{x}^m$ as following [3]:

$$\mathrm{sim}(\mathbf{x}, \mathbf{x}^m) = 1 - \frac{dist(\mathbf{x}, \mathbf{x}^m)}{\max(|\mathbf{x}|, |\mathbf{x}^m|)} \tag{4}$$

where $dist(\cdot)$ denotes the edit-distance and $|\mathbf{x}|$ denotes the word-based length of $\mathbf{x}$.

Following [14], we firstly collect translation pieces from the TM list. Specifically, translation pieces (up to 4-grams) are collected from the retrieved target sentences $\mathbf{y}^m$ as possible translation pieces $G_{\mathbf{x}}^m$ for $\mathbf{x}$, using word-level alignments to select n-grams that are related to $\mathbf{x}$ and discard others. For example, in Fig. 1, the red part of the retrieved TM target sentence is employed to extracted translation pieces for the source sentence, such as "gets", "object" and "object

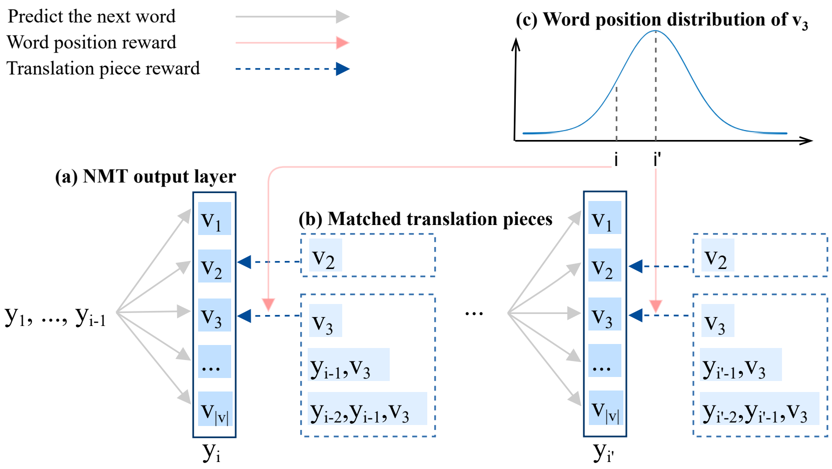

Fig. 2. Adding word position rewards into the NMT output layer. v refers to a word in the target vocabulary, and i' refers to the expected position of word v_3 according to TM. Therefore, the position reward at time i' is larger than that at time i.

that" etc. While the black part of the TM target sentence is the unmatched piece that will not be collected. Formally, the translation pieces $G_{\mathbf{x}}$ from TM are represented as :

$$G_{\mathbf{x}} = \cup_{m=1}^{M} G_{\mathbf{x}}^{m} \tag{5}$$

where $G_{\mathbf{x}}^{m}$ denotes all weighted n-grams from $\langle \mathbf{x}^m, \mathbf{y}^m \rangle$ with n up to 4.

Secondly, we calculate a score for each $u \in G_{\mathbf{x}}$. The weighted score for each u measures how likely it is a correct translation piece for $\mathbf{x}$ based on sentence similarity between the retrieved source sentences $\{\mathbf{x}^m | m \in [1, M]\}$ and the input sentence $\mathbf{x}$ as following:

$$s_p(\mathbf{x}, u) = \max_{1 \leq m \leq M \wedge u \in G_{\mathbf{x}}^{m}} \text{sim}(\mathbf{x}, \mathbf{x}^m) \tag{6}$$

And then, as shown in Fig. 2(**a**)(**b**), an additional translation piece reward for the collected translation pieces will be added to NMT output layer according to:

$$R_p(y_i | \mathbf{y}_{<i}, \mathbf{x}) = \lambda \sum_{n=1}^{4} \delta\big(y_{i-n+1}^{i} \in G_{\mathbf{x}}, s_p(\mathbf{x}, u)\big) \tag{7}$$

where λ can be tuned on the development set and $\delta(cond, val)$ is computed as:

$$\delta(cond, val) = \begin{cases} 0 & \text{if } cond \text{ is } false \\ val & \text{if } cond \text{ is } true \end{cases} \tag{8}$$

Finally, based on Equation 2 and 7, the updated probability $P'(y_i | \mathbf{y}_{<i}, \mathbf{x})$ for the word y_i is calculated by:

$$P'(y_i | \mathbf{y}_{<i}, \mathbf{x}) = P(y_i | \mathbf{y}_{<i}, \mathbf{x}) \times e^{R_p(y_i | \mathbf{y}_{<i}, \mathbf{x})} \tag{9}$$

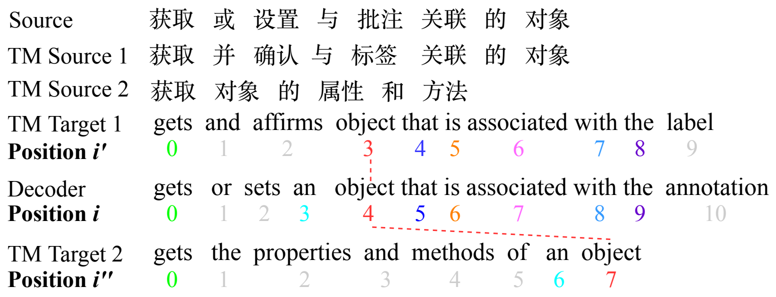
Source 获取 或 设置 与 批注 关联 的 对象
TM Source 1 获取 并 确认 与 标签 关联 的 对象
TM Source 2 获取 对象 的 属性 和 方法
TM Target 1 gets and affirms object that is associated with the label
Position i′ 0 1 2 3 4 5 6 7 8 9
Decoder gets or sets an object that is associated with the annotation
Position i 0 1 2 3 4 5 6 7 8 9 10
TM Target 2 gets the properties and methods of an object
Position i″ 0 1 2 3 4 5 6 7

Fig. 3. An example of word position relationship between translation memory and decoding step. Position i refers to the decoding step and i^* refer to the global position information according to TM. The same color position numbers (except gray) represent the position relationship between translation memory and each decoding step in the NMT output layer. For example, at decoding step 4, the positions of output word "object" are 3 and 7 in TM as shown in red.

In this section, we provide a brief summary of how to use retrieved translation pieces in TM for NMT. For more details, we refer readers to [14].

3 Word Positions Aware TM

In order to improve greatly the translation quality, we hope the NMT output majorly follows the target sentences of TM. Although translation pieces are very useful to accomplish word selection, it is hard to capture sufficient contextual information beyond 4-grams in a TM, leading to the limited translation performance: in particular, given the TM source sentence, it is hard for the translation pieces to guide the NMT model to generate the reliable translation even if its reference is in the TM.

Then, inspired by our intuition stated in Section 1, we study the position of word y in the collected translation pieces, and find that:

- If there is a *low* similarity between the TM source sentence and the input sentence, the positions of word y in translation pieces are less helpful to guide the decoding process.
- In the *middle* similarity situation, the positions of word y in translation pieces are helpful to guide the decoding process.
- In the *high* similarity situation, the positions of word y in translation pieces are very helpful to guide the decoding process.

In general, word positions may be helpful to supply more contextual information or long distance knowledge, and it depends on the similarity between the source and the TM source sentences. As shown in Fig. 3, if the TM source is highly similar to the source, the word position i' in the TM target should be not far away from the word position i in the decoding process. For example, at decoding step 4, the positions of output word “object” are 3 and 7 in TM as shown in red.

Therefore, if we consider the global position of a word in a TM, it is possible to improve NMT with translation pieces. Hence, we try some methods to capture the position distribution such as the linear distribution, the normal distribution, and the multinomial distribution. Finally, we select the normal distribution. As shown in Fig. 2(**a**)(**c**), v refers to a word in the target vocabulary, and i' refers to the expected position of word v_3 according to TM. And we add word position rewards into the NMT output layer according to normal distributions. Therefore, the position reward at time i' is larger than that at time i.

In this paper, we will design two types of position rewards, namely sentence level rewards and piece level rewards, for the given target word v from the retrieved TM according to normal distributions as follows.

3.1 Sentence Level Position

To capture contextual information or long distance knowledge, in this paper, we use the normal distribution to represent the relationship between positions. And we adopt the top-1 TM instance $\mathbf{x}^m, \mathbf{y}^m$ to learn the parameters of distributions for word positions at the sentence level. Finally, the mathematical expectation of the normal distribution is i' and the standard deviation is $2 \cdot sim(\mathbf{x}, \mathbf{x}^m)$. Specifically, for the target word y_i and the translation target position i during decoding, the corresponding position score s_{ps} at the sentence level is calculated as following:

$$s_{ps}(\mathbf{x}, y_i, i) = \frac{e^{-\frac{1}{2}\cdot\left(\frac{i-i'}{2\cdot\mathrm{sim}(\mathbf{x},\mathbf{x}^m)}\right)^2}}{2\sqrt{2\pi}\cdot\mathrm{sim}(\mathbf{x},\mathbf{x}^m)} \tag{10}$$

where i' refers to the position of the word y_i in $\mathbf{y}^m$.

Then, an additional sentence level position reward is calculated as following:

$$R_{ps}(y_i|i, \mathbf{y}_{<i}, \mathbf{x}) = \delta\Big(y_i \in \mathbf{x}^m, s_{ps}(\mathbf{x}, y_i, i)\Big) \tag{11}$$

In this way, the NMT results capture sentence level patterns as we expected, overcoming the limitation of translation pieces and the presence of mismatched source words.

3.2 Piece Level Position

The piece level positions are beneficial to help the underlying NMT system to further capture local patterns. Similar to integrating the sentence level position above, the score of piece level position n $(0 \leq n \leq 3)$ of the word y_i in the collected translation piece u is simply based on the standard normal distribution with the mathematical expectation is 0 and the standard deviation is 1:

$$s_{pp}(\mathbf{x}, y_i, n) = \frac{e^{-\frac{(n+1)^2}{2}}}{\sqrt{2\pi}} \tag{12}$$

where n refers to the relative position of the word y_i in the piece u. For example, as shown in Fig. 3, the translation pieces are collected using the method stated in Section 2.2; such as "associated", "is associated", "that is associated" and "object that is associated" are collected. And at time step 7 when decoding the word "associated" in the NMT ouput layer, the values of n in those four pieces are 0, 1, 2 and 3, separately.

As a result, an additional piece level position reward can be added according to:

$$R_{pp}(y_i|i, \mathbf{y}_{<i}, \mathbf{x}) = \lambda \sum_{n=0}^{3} \delta\big(y_{i-n+1}^{i} \in G_{\mathbf{x}}, s_{pp}(\mathbf{x}, y_i, n)\big) \tag{13}$$

In summary, at each time step i, we update the probabilities over the output vocabulary and increase the probabilities of those that match the expected positions according to:

$$P'(y_i|\mathbf{y}_{<i}, \mathbf{x}) = P(y_i|\mathbf{y}_{<i}, \mathbf{x}) \times e^{R_p(y_i|\mathbf{y}_{<i}, \mathbf{x})} \times e^{R_{ps}(y_i|i, \mathbf{y}_{<i}, \mathbf{x})} \times e^{R_{pp}(y_i|i, \mathbf{y}_{<i}, \mathbf{x})} \tag{14}$$

4 Experiments

In this section, we demonstrate, by experiments, the advantages of the proposed model: it yields better translation on the basis of [14] with the help of word positions from translation memory; and it still be able to keep the low latency in terms of running time mainly because of the lightweight position formulation using normal distributions.

4.1 Settings

To fully explore the effectiveness of our proposed model, we conduct translation experiments on 7 language pairs, namely, zh-en, fr-en, en-fr, es-en, en-es, de-en, and en-de. And we use case-insensitive BLEU score on single references as the automatic metric [7] for translation quality evaluation. We collect about 2 million news sentences from several online news websites for zh-en experiments, and manage to obtain pre-processed JRC-Acquis corpus from [3] for other language pairs. The highly related text in the corpus is suitable for us to make evaluations. For each language pair, we randomly select 2000 samples to form a development and a test set respectively. The rest of the pairs are used as the training set. In addition, we employ Byte Pair Encoding [8] on the previous datasets. We maintain a source/target vocabulary of 35k tokens for each language pair.

As the proposed method is directly build upon the Transformer architecture [11], which is referred to as **TFM** in this paper. Following [14], we implement translation pieces based system on top of Transformer for fair comparison, and it is denoted by **TFM-P**. The implemented systems for the proposed word position integration methods are denoted by **TFM-PS** and **TFM-PSP** for the sentence level positions and the sentence + piece level positions, respectively.

For each sentence, we retrieve 100 translation pairs from the training set by using Apache Lucene, and score them with fuzzy matching score, finally select

Input	关于 增进 了解 与 公共行政 、 参与性 治理 、 能力 建设 、 促进 专业 精神 和 职业道德 以及 知识 管理 促进 发展 有关 的 问题 的 对话 得到 加强
Reference	enriched dialogue on improved understanding of the issues related to public administration , participatory governance , capacity-building and promotion of professionalism and ethics , and knowledge management for development
TM Source	关于 增进 了解 与 公共行政 、 参与性 治理 、 能力 建设 、 促进 专业 精神 和 职业道德 以及 知识 管理 促进 发展 等 有关 的 问题 的 对话 内容 更加 丰富
TM Target	enhanced dialogue on improved understanding of the issues related to public administration , participatory governance , capacity-building and promotion of professionalism and ethics , and knowledge management for development
TFM	strengthened dialogue on enhancing understanding of issues related to public administration , participatory governance , capacity-building , professionalism and ethics and knowledge management for development (Under-translation: “促进”--> “promotion”)
TFM-P	enhanced dialogue on understanding of issues related to public administration , participatory governance , capacity-building , the promotion of professionalism and ethics , and the promotion of the development of knowledge management (Under-translation: “增进” --> “improved”)
TFM-PS	enhanced dialogue on improved understanding of the issues related to public administration , participatory governance , capacity-building and promotion of professionalism and ethics , and the promotion of development of knowledge management
TFM-PSP	enhanced dialogue on improved understanding of the issues related to public administration , participatory governance , capacity-building and promotion of professionalism and ethics , and knowledge management for development

Fig. 4. An example of translation results generated by other methods and our model. **TM Source** denotes the sentence that is most similar to the input. **TM Target** denotes the target sentence of the TM source. The blue parts in the **TFM-*** are the translation pieces extracted from the TM target according to word alignments. Under-translation in the input and its corresponding in the reference are shown in red.

top $N = 5$ translation sentence pairs as the TMs for the sentence $\mathbf{x}$ to be translated.

Furthermore, since there is a hyper-parameter λ in the system TFM-PSP (the same principle for TFM-P and TFM-PS) which is sensitive to the specific translation task, we tune it carefully on the development set for all translation tasks.

4.2 Results and Analysis

Some of translation examples are given in Fig. 4. As shown in Fig. 4, TFM and TFM-P have under-translations while TFM-PS and TFM-PSP don't. Under-translation refers to that some source words are not translated. Our proposed methods can make full use of the fragment information in TM target and obtain translation results which are highly similar to those in TM target, with the help of word positions from translation memory.

Translation Accuracy Table 1 shows the main experimental results. From the overall perspective, we can see that our methods outperform the baseline TFM-P system 0.1 − 2.2 BLEU points varying as tasks. The zh-en translation task

Table 1. Translation accuracy in terms of BLEU on 7 translation tasks. **Best** results are highlighted.

		zh-en	fr-en	en-fr	es-en	en-es	de-en	en-de
Dev	TFM	41.59	65.29	64.46	64.96	62.09	60.50	54.06
	TFM-P	48.87	70.74	68.94	67.10	67.35	65.48	60.86
	TFM-PS	50.57	71.12	69.46	68.90	67.76	65.96	61.66
	TFM-PSP	**50.70**	**71.18**	**69.49**	**69.02**	**67.87**	**65.99**	**61.71**
Test	TFM	40.14	65.43	64.07	63.92	61.48	60.37	53.38
	TFM-P	46.65	70.95	69.12	67.32	66.95	65.13	60.06
	TFM-PS	48.82	71.00	69.45	68.28	67.17	65.49	60.77
	TFM-PSP	**48.84**	**71.01**	**69.50**	**68.51**	**67.22**	**65.54**	**60.81**

Table 2. Similarity Analysis - Translation quality (BLEU score) on zh-en task for the divided subsets according to similarity. **Best** results are highlighted.

	Dev				Test			
Similarity	[0.0,0.4)	[0.4,0.7)	[0.7,1.0]	[0.0,1.0]	[0.0,0.4)	[0.4,0.7)	[0.7,1.0]	[0.0,1.0]
Ratio(%)	70.64	8.06	21.30	100.00	72.98	7.37	19.65	100.00
TFM	37.39	49.01	49.05	41.59	36.83	49.11	46.83	40.14
TFM-P	37.60	57.77	71.67	48.87	37.53	56.05	66.93	46.65
TFM-PS	**37.62**	59.19	77.55	50.57	**37.57**	**57.08**	75.60	48.82
TFM-PSP	37.61	**59.45**	**78.13**	**50.70**	37.54	57.03	**75.90**	**48.84**

obtains the maximized promotion with the word position integration, while the fr-en translation task cannot make an immediate benefits as the bold numbers shown in Table 1. The main reason is that the baseline is extraordinarily strong (fr-en: 70.95 vs zh-en: 46.65), and this result is still consistent with the discovery reported in [14].

Influence on Similarity In order to dig deeper on the influence of various similarities, we reported the translation quality on zh-en task for the divided subsets according to similarity, in terms of BLEU and TER [10] as shown in Table 2 and 3, respectively.

The low similarity subset which is in the range of [0.0, 0.4), does little to help the result. And the middle similarity subset [0.4, 0.7) obtains improvements by 1 BLEU point. The high similarity subset that is in the range of [0.7, 1.0], obtains significant improvements, up to 9 BLEU points and down to 9.16 TER (The lower the TER value, the better.) points for the test set, respectively, with the help of word position rewards as we expected according to [13].

Table 4 shows statistics of each dev and test set on seven translation tasks where sentences are grouped by their similarity scores. In addition, the sentence level word positions are the main contributors to the quality improvement. In this way, we can conclude that the word positions extracted from TM are efficient to improve the final translation results in most cases, especially for those source sentences that are very similar to TM.

Table 3. Similarity Analysis - Translation quality (TER score) on zh-en task for the divided subsets according to similarity. **Best** results are highlighted.

	Dev				Test			
Similarity	[0.0,0.4)	[0.4,0.7)	[0.7,1.0]	[0.0,1.0]	[0.0,0.4)	[0.4,0.7)	[0.7,1.0]	[0.0,1.0]
Ratio(%)	70.64	8.06	21.30	100.00	72.98	7.37	19.65	100.00
TFM	50.85	40.74	40.08	47.20	50.68	40.86	42.59	48.07
TFM-P	**50.81**	36.20	25.41	43.00	50.59	35.32	30.77	45.00
TFM-PS	50.83	35.10	20.21	41.60	**50.44**	**35.23**	21.75	42.75
TFM-PSP	50.84	**35.01**	**19.65**	**41.50**	50.45	35.27	**21.61**	**42.74**

Table 4. Composition of dev and test sets based on similarity score on 7 translation tasks.

<table>
<tr><th>(Dev|Test)
Ratio(%)</th><th>zh-en</th><th>fr-en</th><th>en-fr</th><th>es-en</th><th>en-es</th><th>de-en</th><th>en-de</th></tr>
<tr><td>[0,0.1)</td><td>4.03|5.23</td><td>1.35|0.85</td><td>0.25|0.35</td><td>0.20|0.15</td><td>1.50|1.20</td><td>0.45|0.45</td><td>2.00|1.80</td></tr>
<tr><td>[0.1,0.2)</td><td>43.74|42.81</td><td>9.85|11.3</td><td>4.85|6.55</td><td>5.45|4.95</td><td>10.00|11.20</td><td>9.65|9.25</td><td>12.45|13.25</td></tr>
<tr><td>[0.2,0.3)</td><td>16.23|18.55</td><td>11.10|10.05</td><td>12.15|10.55</td><td>15.00|15.30</td><td>13.55|13.75</td><td>13.45|14.65</td><td>11.40|11.55</td></tr>
<tr><td>[0.3,0.4)</td><td>6.64|6.38</td><td>10.00|10.40</td><td>10.90|10.50</td><td>13.25|11.90</td><td>10.15|8.45</td><td>10.85|10.80</td><td>10.35|9.20</td></tr>
<tr><td>[0.4,0.5)</td><td>3.00|2.97</td><td>7.90|7.15</td><td>7.40|8.30</td><td>8.20|8.60</td><td>7.80|6.25</td><td>8.50|7.95</td><td>7.00|6.05</td></tr>
<tr><td>[0.5,0.6)</td><td>2.89|2.37</td><td>8.65|8.10</td><td>11.55|10.05</td><td>8.60|10.45</td><td>6.50|9.40</td><td>8.55|8.65</td><td>8.30|8.85</td></tr>
<tr><td>[0.6,0.7)</td><td>2.18|2.03</td><td>10.15|10.65</td><td>10.50|10.30</td><td>8.45|8.65</td><td>8.65|8.05</td><td>8.60|8.15</td><td>7.80|7.70</td></tr>
<tr><td>[0.7,0.8)</td><td>2.89|2.70</td><td>13.00|12.90</td><td>12.75|14.10</td><td>9.00|9.30</td><td>8.80|9.35</td><td>9.40|9.75</td><td>8.55|9.85</td></tr>
<tr><td>[0.8,0.9)</td><td>5.77|5.50</td><td>15.05|15.55</td><td>16.30|16.20</td><td>16.30|15.65</td><td>16.25|16.15</td><td>17.65|15.70</td><td>17.20|17.00</td></tr>
<tr><td>[0.9,1)</td><td>12.58|11.45</td><td>12.95|13.05</td><td>13.25|13.10</td><td>15.65|15.05</td><td>16.80|16.20</td><td>12.90|14.65</td><td>14.95|14.75</td></tr>
<tr><td>[0,1)</td><td>100|100</td><td>100|100</td><td>100|100</td><td>100|100</td><td>100|100</td><td>100|100</td><td>100|100</td></tr>
</table>

Running Time We eliminate the retrieval time and directly compare running time for neural models as shown in Table 5. From this table, we observe that our proposed approach still be able to keep the low latency, compared to the baseline TFM-P employing translation pieces, and our system TFM-PSP achieves better translation performance with sentence and piece level positions.

Hyper-parameter Robustness At last, we try to verify the robustness of the hyper-parameter λ among various translation tasks, and show the search process in Table 6 on zh-en task. As shown in Table 6, there is enough parameter space for λ to keep smaller translation quality volatility. In general, we can search a better value for λ in the range of [1.0, 1.3] for other translation tasks.

In summary, the extensive experimental results show that the proposed approach achieves better translation on the basis of [14] with the help of word positions from TM, especially for those source sentences that are very similar to TM. In addition, this approach still be able to keep the low latency in terms of running time.

5 Related Work

In SMT paradigm, many research works are devoted to integrating a translation memory into the SMT [4, 6, 12]. Such as [4] extracted bilingual segments from a

Table 5. Running time in terms of seconds/sentence on zh-en task. The average lengths of sentences in Dev and Test are 31.34 and 31.17 words/sentence, respectively.

	TFM	TFM-P	TFM-PS	TFM-PSP
Dev	0.31	0.76	0.76	0.86
Test	0.31	0.76	0.71	0.85

Table 6. Translation quality (BLEU score) among various values of λ on zh-en task.

λ	1.0	1.1	1.2	1.3	1.4
Dev	50.36	**50.70**	50.58	49.99	49.92
Test	48.82	48.84	**48.89**	48.70	48.15

TM which matched the source sentence to be translated, and adopted SMT to decode for those unmatched parts of the source sentence.

Recently, TM based NMT has been witnessed the increasing interests. As NMT does not explicitly rely on the translation rules as SMT, many works resort to different approaches. For example, Li et al. and Farajian et al. [5, 2] proposed a fine tuning approach to train a sentence-wise local neural model on top of a retrieved TM, which was further used for testing a particular sentence. The standard NMT model was augmented by additionally encoding a TM for each testing sentence in [3] and [13], and the proposed global models were trained to optimize for testing all source sentences. However, the above two approaches require intensive computation and considerable memory.

Considering the complexity in computation and memory, a simple and effective method that retrieved translation pieces to guide NMT for narrow domains was proposed in [14]. Their method was effective and simple, however, it can only captured local information in a hard manner while ignoring the global information in TM. Hence, in order to keep the low complexity and capture both global and local context information, in this work, we study the distribution of word positions in the collected translation pieces from TM, and employ the word position information as additional rewards to guide the decoding of NMT.

6 Conclusion

To capture sufficient contextual information in translation pieces extracted from translation memory, we have proposed a novel method that integrates sentence and piece level positions of translation memory into neural machine translation. The extensive experimental results on 7 translation tasks have demonstrated that the proposed method further achieve better translation results on the basis of integrating translation pieces, especially for those source sentences that are very similar to those retrieved from translation memory. What's more, this approach still be able to keep the low latency and memory consumption, and the system architecture in brief.

Acknowledgments

This work is supported by NSFC (grant No. 61877051).

References

1. Bahdanau, D., Cho, K., Bengio, Y.: Neural machine translation by jointly learning to align and translate (2016), arXiv preprint arXiv:1409.0473
2. Farajian, M.A., Turchi, M., Negri, M., Federico, M.: Multi-domain neural machine translation through unsupervised adaptation. In: Proceedings of the Second Conference on Machine Translation. pp. 127–137 (2017)
3. Gu, J., Wang, Y., Cho, K., Li, V.O.: Search engine guided non-parametric neural machine translation. In: Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018). pp. 5133–5140 (2018)
4. Koehn, P., Senellart, J.: Convergence of translation memory and statistical machine translation. In: Proceedings of AMTA Workshop on MT Research and the Translation Industry. pp. 21–31 (2010)
5. Li, X., Zhang, J., Zong, C.: One sentence one model for neural machine translation (2016), arXiv preprint arXiv:1609.06490
6. Ma, Y., He, Y., Way, A., van Genabith, J.: Consistent translation using discriminative learning: A translation memory-inspired approach. In: Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics (ACL 2011). pp. 1239–1248 (2011)
7. Papineni, K., Roukos, S., Ward, T., Zhu, W.: Bleu: a method for automatic evaluation of machine translation. In: Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics ACL 2002. pp. 311–318. ACL (2002)
8. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016). pp. 1715–1725 (2016)
9. Simard, M., Isabelle, P.: Phrase-based machine translation in a computer-assisted translation environment. In: Proceedings of the Twelfth Machine Translation Summit (MT Summit XII). pp. 120–127 (2009)
10. Snover, M., Dorr, B., Schwartz, R., Micciulla, L., Makhoul, J.: A study of translation edit rate with targeted human annotation. In: Proceedings of the 7th Conference of the Association for Machine Translation in the Americas. pp. 223–231 (2006)
11. Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A.N., Kaiser, L.u., Polosukhin, I.: Attention is all you need. In: Advances in Neural Information Processing Systems 30. pp. 5998–6008 (2017)
12. Wang, K., Zong, C., Su, K.Y.: Integrating translation memory into phrase-based machine translation during decoding. In: Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics (ACL 2013). pp. 11–21 (2013)
13. Xia, M., Huang, G., Liu, L., Shi, S.: Graph based translation memory for neural machine translation. In: Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019). pp. 7297–7304 (2019)
14. Zhang, J., Utiyama, M., Sumita, E., Neubig, G., Nakamura, S.: Guiding neural machine translation with retrieved translation pieces. In: Proceedings of the 16th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-HLT 2018). pp. 1325–1335 (2018)