Prompting Neural Machine Translation with Translation Memories

Abudurexiti Reheman[1], Tao Zhou[1], Yingfeng Luo[1], Di Yang[2], Tong Xiao[1,2], Jingbo Zhu[1,2*]

[1]School of Computer Science and Engineering, Northeastern University, Shenyang, China
[2]NiuTrans Research, Shenyang, China
rexiti_neu@outlook.com, zhoutao_neu@outlook.com, luoyingfengmail@163.com,
yangdi@niutrans.com, {xiaotong, zhujingbo}@mail.neu.edu.cn

Abstract

Improving machine translation (MT) systems with translation memories (TMs) is of great interest to practitioners in the MT community. However, previous approaches require either a significant update of the model architecture and/or additional training efforts to make the models well-behaved when TMs are taken as additional input. In this paper, we present a simple but effective method to introduce TMs into neural machine translation (NMT) systems. Specifically, we treat TMs as prompts to the NMT model at test time, but leave the training process unchanged. The result is a slight update of an existing NMT system, which can be implemented in a few hours by anyone who is familiar with NMT. Experimental results on several datasets demonstrate that our system significantly outperforms strong baselines.

Introduction

Integrating TM is one of the commonly used techniques to improve real-world MT systems. In TM-assisted MT systems, it is often assumed that there is a database in which high-quality bilingual sentence pairs are stored. When translating an input sentence, the most (or top-K) similar sentence pair, which is retrieved from TM, is used to optimize the translation. From the perspective of practical application, this approach is particularly useful for MT, especially when sentences are highly repetitive, such as in translating technical manuals, legal provisions, etc. Previous works show that translation quality can be significantly improved when a well-matched TM sentence pair is provided both in Statistical Machine Translation (SMT) (Ma et al. 2011; Wang, Zong, and Su 2013; Li, Way, and Liu 2014) and Neural Machine Translation (NMT) (Gu et al. 2018; Khandelwal et al. 2020).

However, there are two major problems with this type of work in real-world applications. First, it is difficult to find such a TM dataset in most cases, especially when users can not share their TM data with the public for some reason. Second, previous approaches often require model changes, including training the model with TM (Bulté and Tezcan 2019; Hossain, Ghazvininejad, and Zettlemoyer 2020; Xu, Crego, and Senellart 2020), changing the NMT model architecture for TM integration (Gu et al. 2018; Bapna and Firat 2019; Xia et al. 2019; He et al. 2021), and introducing additional modules (Zhang et al. 2018; He et al. 2019; Khandelwal et al. 2020). In this case, it is difficult to incorporate TM into the NMT system even if TM data is provided, since TM incorporation can not be accomplished on a generic decoder, and a deeply customized decoder is needed.

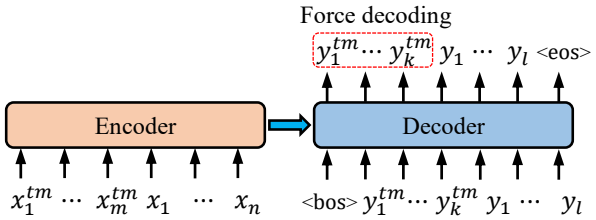

Figure 1: Structure of the proposed method. The source and target sentences of TM are concatenated with the input sentence and hypothesis in a specific concatenation template, respectively. The tokens in the target TM together with the concatenation template are generated in a forced manner. The lengths of the source and target TM together with the concatenation templates are m and k, and the lengths of the input sentence and hypothesis are n and l, respectively.

Here, we address this problem by using few-shot learning (Wang et al. 2021b), which enables the system to quickly adapt to a small number of samples. The recent prevalence of prompt-based approaches (Brown et al. 2020), which transfer the original task into a generation task by designing an appropriate template without modifying the language model, gives us some inspiration that the retrieved TM can prompt the translation of the input sentence without modifying the NMT model.

Based on this idea, we propose a simple approach to quickly adapt the NMT model in the few-shot TM scenario. Specifically, we treat TMs as prompts to the NMT model during the decoding process, with very small changes to the decoder. Our method can cover the advantages of conventional TM augmented methods and bring some new ideas, such as incorporating users' local historical translation data into NMT.

*Corresponding author.

In order to prompt the translation with the retrieved TM, we design several templates to concatenate the source TM with the input sentence and feed the concatenated sentence into the model encoder. On the decoder side, we generate the target TM and the concatenation template in a forced way first, then let the model generate the other parts automatically. Regarding TM granularity, our method works well on sentence-level and fragment-level TM by designing appropriate templates. Experimental results on several datasets show that our method can further improve the translation quality on strong NMT models, and with comparable performance with the state-of-the-art.

Background

NMT Decoding

Suppose $x = \{x_1, ..., x_n\}$ is the source sentence, and NMT translates it into the corresponding target sentence $y = \{y_1, ..., y_m\}$ by using a trained NMT model. In practice, it turns the decoding into a searching problem, and a beam searcher is adopted to get the target sentence with the highest generation probability. Generally, an NMT model generates in an auto-regressive way. Therefore, the generation of each token relies on the source sentence and the generated prefix of the target sentence. The generation of the whole target sentence can be formulated as a conditional probability $P(y|x)$ described below:

$$P(y|x) = \prod_{i=0}^{m} P(y_i|x, y_{<i}) \quad (1)$$

where $y_{<i} = \{y_1, y_2, ..., y_{i-1}\}$ denotes the generated prefix tokens of target sentence at time-step i.

TM

TM is a database of language pairs that stores segments (such as fragments, sentences, paragraphs, or other sentence-like units) that have previously been translated by human translators for later use. It can provide identical or similar segments to help translate the input sentence. In the early stage, TM was widely used in Computer Aided Translation (CAT) (Dillon and Fraser 2006). When translating an input sentence, the target sentence of TM is returned as the answer when an identical source sentence is found. In most cases, the translation result is obtained by fixing the most similar or top-K similar sentences retrieved from TM, since it is difficult to find a completely identical sentence. In the MT environment, TM is playing the same role as in CAT. Many approaches have been proposed to incorporate similar TM into MT systems to get a more accurate translation.

Methodology

In this section, we introduce our proposed method in detail from the perspectives of incorporating TM into NMT decoding, designing templates for concatenation, and retrieving similar TM.

Incorporating TMs into NMT Decoding

We incorporate TMs into the decoding process. For an input sentence x and a retrieved TM sentence pair $\langle x^{tm}, y^{tm} \rangle$, we concatenate x^{tm} and x in a specific concatenation template and feed it into the model encoder. On the decoder side, we first force the model to generate the exact tokens in y^{tm} together with the concatenation template. Then the rest of the generation, which is returned as the answer, is done automatically without interfering. In practice, we set the generation probability of the tokens in the target TM and the concatenation template as 1 during the force decoding phase.

Concatenation Templates

As our method incorporates TM on a pre-trained NMT model with no modifications, we must adhere to the following principles when designing the concatenation template: tokens in the concatenation template must be recognized by the NMT model and each part of the concatenated sentence maintain relatively complete semantics. Following this idea, we designed several templates for TMs in different granularity. In concatenation, the TM comes first on both the source and target sides, and a specific template is used on both sides.

Sentence-level TM. We concatenate sentence-level TMs in five different templates. As the example shown in Table 1, our designed templates for sentence-level TMs are as follows:

(a) Concatenate directly. We concatenate them directly and add a period at the end of TMs if it is not ended up with punctuation marks.

(b) Concatenate with comma. Before concatenation, we replace the punctuation mark at the end of TMs with a comma or add a comma there if it is not ended up with any punctuation mark.

(c) Concatenate with semicolon. The concatenation is the same as the comma concatenation process, using a semicolon instead of the comma.

(d) Concatenate with conjunctions. In this template, we adopt conjunctions that express juxtaposed semantics, such as "and" in English and "und" in German. Specifically, we add a period at the end of the source and target TM if they are not ended up with any punctuation mark, then add a conjunction word in the corresponding language and a comma after that. Then we concatenate them with the input sentence and the hypothesis on the source and target side, respectively.

(e) Enclose in parentheses. We enclose both source and target TMs in parentheses, then perform the concatenation.

Fragment-level TM. Unlike the sentence-level TM, we do some preprocessing on fragment-level TMs. First, we obtain the common fragments between the input sentence and source TM. Then, the tokens which are the translation of the words in the above common fragments are acquired from the target TM, using word alignment tools. After that, we construct the fragment-level TM using the tokens from the source and target TM above. An example of common fragments between input sentence and source TM and the word

Input Sentence		She gave us a full account of the traffic accident .
Source TM		She gave the police a full account of the incident .
Target TM		Sie gab der Polizei einen voll@@ ständigen Bericht über den Vorfall .
Sentence Level TMs		
Directly	En/input	She gave the police a full account of the incident . She gave us a full account of the traffic accident .
	De/input	<bos> Sie gab der Polizei einen voll@@ ständigen Bericht über den Vorfall . {Hypothesis}
Comma	En/input	She gave the police a full account of the incident , She gave us a full account of the traffic accident .
	De/input	<bos> Sie gab der Polizei einen voll@@ ständigen Bericht über den Vorfall , {Hypothesis}
Semicolon	En/input	She gave the police a full account of the incident ; She gave us a full account of the traffic accident .
	De/input	<bos> Sie gab der Polizei einen voll@@ ständigen Bericht über den Vorfall ; {Hypothesis}
Conjunction	En/input	She gave the police a full account of the incident . And , She gave us a full account of the traffic accident .
	De/input	<bos> Sie gab der Polizei einen voll@@ ständigen Bericht über den Vorfall . Und , {Hypothesis}
Parenthesis	En/input	(She gave the police a full account of the incident .) She gave us a full account of the traffic accident .
	De/input	<bos> (Sie gab der Polizei einen voll@@ ständigen Bericht über den Vorfall .) {Hypothesis}
Fragment Level TMs		
Parenthesis	En/input	(She gave) (a full account of the) She gave us a full account of the traffic accident .
	De/input	<bos> (Sie gab) (einen voll@@ ständigen Bericht über den) {Hypothesis}

Table 1: An example of model input in our proposed method. Here, Directly, Comma, Semicolon, Conjunction, and Parenthesis denote our designed templates for concatenation, and En/input and De/input denote the input of the encoder and the decoder, respectively. The <bos> token denotes the begin-of-sentence tag, and {Hypothesis} denotes the automatically generated part of the target sentence.

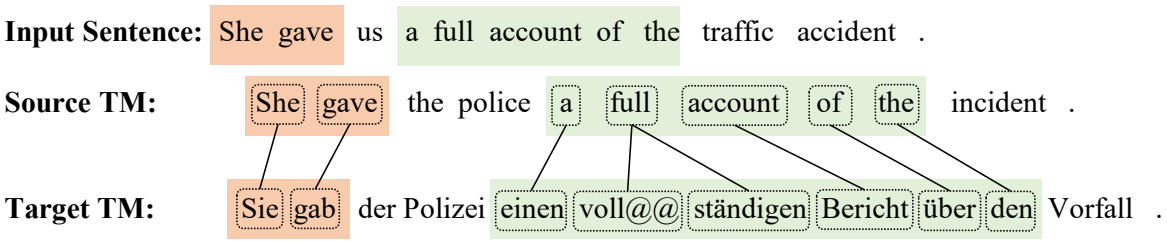

Figure 2: An example of obtaining fragments for fragment-level TMs. For a given bilingual TM, common fragments between the input sentence and the source TM are acquired first, then the words in the target TM that align with the words in the common fragments are extracted. Common fragments and their corresponding fragments in Target TM are tagged by the same color box, and the lines denote word alignments.

alignment between the source and target TM is given in Figure 2, and its corresponding fragment-level TM is given in Table 1.

Specifically, for a given input sentence x and a retrieved bilingual TM $\langle x^{tm}, y^{tm} \rangle$, we acquire the encoder and decoder input for the NMT model in the following steps:

(a) Perform the longest common subsequence matching algorithm to x and x^{tm}, and obtain the longest common subsequence $P_s = \{w_1, w_2, ..., w_m\}$.

(b) Use word alignment tools to x^{tm} and y^{tm}, and get the aligned subsequence $P_t = \{w'_1, w'_2, ..., w'_n\}$, which is corresponding to P_s, from y^{tm}.

(c) Group the words, which appear continuously in x^{tm}, from P_s in their original order to form source TM fragments $P'_s = \{f_1, f_2, ..., f_i\}$.

(d) Group the words, which appear continuously in y^{tm}, from P_t in the order of the correspondence with P'_s to form target TM fragments $P'_t = \{f'_1, f'_2, ..., f'_j\}$.

(e) Concatenate each fragment in P'_s and P'_t with a specific template to form fragment-level TM, and directly concatenate them with input sentence and hypothesis as is done in sentence-level TM.

As for the concatenation template, we enclose each fragment in parenthesis to maintain its semantic integrity. In practice, we remove the fragments that consist of a single stop word and remove the punctuation at two sides of a fragment.

Retrieving Similar TMs

To retrieve the most similar bilingual TM for the input sentence, we use a word-level fuzzy matching strategy and remove punctuations and numbers from the sentence. Instead of retrieving from the whole TM database, we first employ the search engine library Apache Lucene (Bialecki, Muir, and Ingersoll 2012) to retrieve the top 500 similar bilingual sentences from TM. Then we rerank them by adopting *Fuzzy*

Match Score (FMS) to obtain the most similar TM sentence pair. *FMS* is a length normalized Levenshtein Distance (Li and Liu 2007), known as Edit Distance:

$$FMS(x, x^{tm}) = 1 - \frac{LD(x, x^{tm})}{\max(|x|, |x^{tm}|)} \quad (2)$$

where $LD(\cdot, \cdot)$ denotes the word level Levenshtein Distance, and $|\cdot|$ denotes word level length of a sentence.

Experiments

In order to verify the validity of our proposed method, we conducted several experiments on TM specialized translation task and domain adaptation task, respectively. We also put our approach into practice on a commercial NMT system to assess its usability in the practical setting. In the end, we investigated the impact of the NMT model, TM similarity, and input sentence length on translation quality.

Datasets and Models

For TM specialized translation tasks, we evaluated our method on two datasets: 1) DGT-TM, the entire body of European legislation in 22 European languages, on German-English in both directions (En-De and De-En) and 2) United Nations Parallel Corpus (UNPC), consisting of United Nations General Assembly Resolutions with translations in the six official languages, on English-Chinese (En-Zh), Russian-Chinese (Ru-Zh) and French-Chinese (Fr-Zh). These two datasets are relatively easy to retrieve TM sentences with a high degree of similarity. For the test set and TM database, we cleaned the above corpora first, then randomly selected 3,000 sentence pairs for the test dataset, whereas the remaining corpora were utilized as the TM database. For tokenization, we used NiuTrans (Xiao et al. 2012) word segmentation tool for Chinese and Moses toolkit (Koehn et al. 2007) for other languages.

In addition, we performed an experiment using a homemade English-Chinese dataset (denoted as H-m in Table 2) of 3401 sentences. Each test sentence has one bilingual TM sentence whose source side is similar to the test sentence. The statistics of these TM databases and the TM similarity ratios of retrieved TMs in FMS metric are shown in Table 2.

In the domain adaptation task, following Khandelwal et al. (2020), we used the multi-domain datasets from Aharoni and Goldberg (2020), which contains German-English bilingual datasets in five different domains: Medical, Law, IT, Koran, and Subtitles, respectively. We treated the training data in each domain as our TM database.

After cleaning the above corpora and splitting them into a test set and TM database, we retrieved the most similar TM for each test sentence from the TM database in an offline way and applied BPE (Sennrich, Haddow, and Birch 2016) to the test sets and TM with the BPE-codes provided by the pre-trained NMT models. To obtain the alignment information for the tokens in the TM source and target sentence, we trained a word aligner – Mask-Align – as proposed in Chen, Sun, and Liu (2021), and constructed corresponding fragment level TMs. The TM data scale and the TM similarity ratios of retrieved TMs in FMS metric are given in Table 3.

Corpus	Lang	TM scale	TM FMS ratio [0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, 0.8)	[0.8, 1.0)
DGT -TM	En-De	3.1M	2%	24%	16%	16%	42%
	De-En	3.1M	4%	26%	17%	17%	36%
UNPC	En-Zh	11.7M	2%	44%	22%	11%	22%
	Fr-Zh	11.5M	2%	45%	18%	11%	23%
	Ru-Zh	11.2M	8%	46%	16%	9%	20%
H-m	En-Zh	-	18%	30%	35%	23%	7%

Table 2: Sentence numbers in the TM databases and the similarity ratios of the retrieved TM.

Domain	TM scale	TM FMS ratio [0, 0.2)	[0.2, 0.4)	[0.4, 0.6)	[0.6, 0.8)	[0.8, 1.0)
Medical	248K	7%	23%	20%	17%	33%
Law	467K	8%	31%	18%	14%	28%
IT	223K	14%	18%	28%	26%	14%
Koran	18K	2%	26%	33%	28%	11%
Subtitles	500K	3%	27%	43%	23%	4%

Table 3: Sentence numbers in the TM database in each domain and the similarity ratios of the retrieved TM.

As for the pre-trained NMT model, we applied Facebook's WMT19 De-En, En-De model (Ng et al. 2019) and NiuTrans' WMT20 En-Zh model (Zhang et al. 2020) as our base model. All of these models are very competitive that they trained on more than 20 million training data and 10 million extra back-translated data.

Main Experiment

Our main experiment involves the TM specialized translation task, the domain adaptation task, and the implementation on commercial NMT system.

TM Specialized translation. In this experiment, we decoded the DGT-TM En-De and De-En test sets using facebook's WMT19 En-De and De-En models (Ng et al. 2019), respectively. Besides, we decoded the UNPC En-Zh test set and the homemade En-Zh test set with NiuTrans' WMT20 En-Zh model (Zhang et al. 2020). For the Mask-Align model training, we used WMT20 En-Zh training data for the En-Zh aligner and DGT-TM's TM database for En-De and De-En aligner. From the experimental results in Table 4, we have the following observations.

First, in sentence-level TM, the BLEU score on DGT-TM De-En, En-De, and homemade En-Zh test sets increased significantly, with maximum BLEU score increases of 8.63, 5.74, and 7.74 points, respectively. Meanwhile, the translation improved slightly on the UNPC En-Zh test set in directly, semicolon, and parenthesis concatenations. Second, in fragment-level TM, the BLEU score increased by about 1 to 2 points or even decreased, compared to the baseline (without TM). The main reason for this is that the NMT model is trained on sentence-level training data rather than

Corpus		DGT-TM		UNPC	H-m
Lang		De-En	En-De	En-Zh	En-Zh
W/o TM		45.40	39.03	41.42	46.43
Sentence TM	Directly	53.74	44.32	41.70	52.97
	Comma	52.44	43.03	41.33	51.85
	Semico	53.42	44.54	**42.31**	52.89
	Conjunc	53.65	44.00	41.15	**54.17**
	Parenth	**54.03**	**44.77**	41.90	53.87
Fragment TM		47.21	41.65	39.85	47.67

Table 4: Experimental results on the DGT-TM En-De, De-En, UNPC En-Zh, and the homemade En-Zh test sets.

sentence pieces. In addition, it is also affected by the performance of the word aligner, which may provide error alignment information.

Domain Adaptation. Following the kNN-MT (Khandelwal et al. 2020) and its optimized counterparts, we conducted the domain adaptation experiment and compared our method with kNN-MT. We applied Facebook's WMT19 De-En model (Ng et al. 2019) for decoding. Experimental results are given in Table 5.

From the table, we can find that our method improves the translation in all domains except Subtitles, with maximum BLEU score improvements of 2.54, 3.05, 4.64 in IT, Law, and Medical domains, respectively, whereas in Koran the result is only 0.42 BLEU score higher. The fragment-level TM method has the same tendency as the above experiment, which is slightly higher only in IT and Medical domain than the baseline. Besides, the improvement of our method is less than kNN-MT in every domain. The original design of our approach leads to this result. Our method retrieves the most similar single TM and leverages the knowledge it contains to improve the translation. How to leverage the knowledge provided by TM is fully dependent on the NMT model itself. While kNN-MT introduces an extra module to incorporate the information explicitly from multiple similar context vectors.

The main advantage of our method over kNN-MT is that our method performs the retrieval based on string similarity, and there is no need to store the context vectors, which saves a lot of storage space. At the same time, kNN-MT searches the context vectors in each beam in every timestep, which is much slower than the vanilla NMT. However, our method searches the TM only once and generates two sentences (target TM and the hypothesis) in the way of a vanilla NMT. This will make our method much faster than kNN-MT.

Implementation on Commercial NMT System. We implemented our proposed method on a commercial NMT system – NiuTrans Enterprise – to evaluate our method's applicability in the real-world environment. We experimented on UNPC En-Zh, Fr-Zh, and Ru-Zh, without modifying the NMT model, even not aware of what kind of NMT model is used. The word aligners for fragment-level TM on Fr-Zh and Ru-Zh are trained on Fr-Zh and Ru-Zh TM databases, respectively. Experimental results in Table 6 show that the

Domains		IT	Koran	Law	Medical	Subtitles
kNN-MT		45.82	19.45	61.78	54.35	31.73
W/o TM		38.09	17.11	45.92	41.14	**29.45**
Sentence TM	Directly	40.19	17.20	48.78	45.29	28.18
	Comma	39.20	16.46	47.42	43.47	25.09
	Semico	39.74	17.09	48.91	44.93	26.44
	Conjunc	40.13	17.03	**48.97**	45.13	27.68
	Parenth	**40.63**	**17.53**	48.31	**45.78**	29.03
Fragment TM		39.38	16.49	45.58	43.31	28.24

Table 5: Experimental results on multi-domain datasets.

Lang		En-Zh	Fr-Zh	Ru-Zh
W/o TM		41.59	29.83	35.62
Sentence TM	Directly	44.18	33.10	37.94
	Comma	43.85	31.63	37.36
	Semico	44.48	**33.38**	37.89
	Conjunc	44.16	33.04	37.97
	Parenth	**44.53**	33.05	**38.68**
Fragment TM		38.74	27.78	32.65

Table 6: Experimental results on a commercial NMT system.

maximum improvement of sentence-level TM on En-Zh, Fr-Zh, and Ru-Zh are 2.94, 3.55, 3.06 BLEU points, respectively, and the fragment-level TM approach still get lower BLEU scores than the baseline. The NMT models used in this experiment have been trained on much more high-quality training data than other models used in the above experiments. The experimental results demonstrate that even strong commercial NMT systems can be further improved when similar TMs provided and that the sentence-level TM approach can be applied in real-world situations where similar TMs for input sentences are available.

NMT Model's Effect on Translation

In our proposed method, the generation of each token in the target sentence relies on source TM, input sentence, target TM and the generated part of target sentence, and the target TM is generated in a forced way. Therefore, the translation depends on the translation ability of the NMT model, "strong" models improve greater, and "weak" models improve less or even get worse results. In order to investigate to what extent the results depend on the NMT model's "strength", we conducted a series of experiments on the UNPC En-Zh test set (see Table 2) with different NMT models. We measure the translation ability of a model in terms of the training data scale and the model architecture. So, we trained several NMT models using WMT20 En-Zh training data, from the perspectives of training data scale and model architecture.

Training Data Scale. The training data of WMT20 En-Zh has 20 million bilingual sentences. We uniformly split them into four parts after shuffling, then trained four NMT models with different data scales, in which the first model is trained

Models	b5M	b10M	b15M	b20M	ba20M	bb20M
W/o TM	41.11	41.40	42.53	43.19	41.47	42.53
Directly	41.77	42.45	44.27	44.26	41.87	43.75
Comma	41.29	41.88	43.79	43.99	41.18	43.36
Semico	**42.27**	42.45	**44.64**	**45.13**	**42.12**	**44.28**
Conjunc	41.48	42.04	43.87	44.33	41.54	43.09
Parenth	41.68	**42.63**	44.49	44.53	41.93	43.85
Max Δ	1.16	1.23	2.11	1.94	0.65	1.75

Table 7: Experimental results on UNPC En-Zh test set with different NMT models, including four transformer big models trained on 5 million, 10 million, 15 million, and 20 million training data (denoted as b5M, b10M, b15M, b20M, respectively), and a transformer base and a bigger model trained on 20 million training data (denoted as ba20M and bb20M, respectively), max Δ denotes the maximum improvement comparing to decoding without TM.

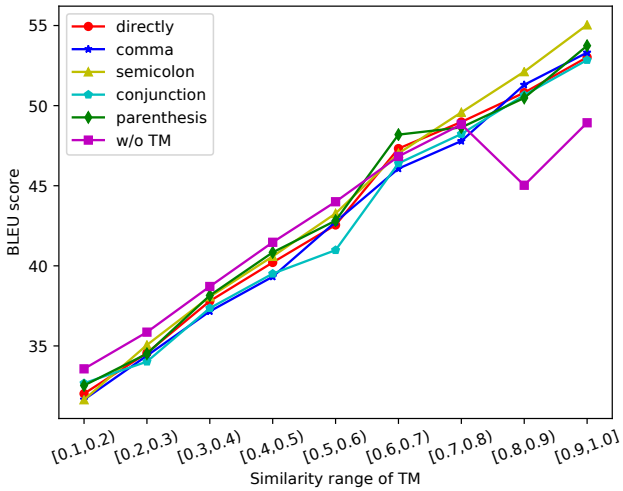

Figure 3: BLEU scores on different similarity ranges.

on the first 5 million datasets, the second model is trained on the first and second 5 million datasets, and so on. All of the models are transformer big models proposed in Vaswani et al. (2017). The experimental results are given in Table 7.

Model Architecture. In this experiment, we investigate the impact of the model architecture on our proposed method. We chose WMT20 En-Zh dataset with 20 million bilingual sentences in the above experiment and trained the transformer base, big and bigger models, respectively. Their attention heads, hidden sizes, and filter sizes are (8, 512, 2048), (16, 1024, 4096), and (24, 1536, 6144), respectively. From the experimental results in Table 7, we have the observations below.

For the same model architecture, with the increase of training data scale, the translation ability of the model is getting stronger, and the BELU improvement of our method is also higher compared with the baseline, as the maximum BLEU score improvement of the models b15M and b20M are higher than that of b5M and b10M. In addition, for the models trained on the same training data, the ba20M model is "weaker" than the b20M and bb20M models, and the maximum BLEU score improvement is also lower than the latter two models. We can find a similar phenomenon if we look back to Table 4 and Table 6. The dataset for UNPC En-Zh is the same in these two experiments, and the NMT model of the commercial NMT system is much "strong" than the NMT model used in table 4, and the maximum improvement of the former is an 2.94 BLEU points, whereas the latter's is 0.89 BLEU points. From these results, we conclude that the sentence-level TM approach of our method can further improve strong baselines, and the "stronger" the NMT model, the greater the improvement will be.

TM Similarity

As our method obtain useful information from a single TM sentence pair, the translation result is influenced by the similarity of the retrieved TM. TMs with high similarity provide more useful information for the translation, while less similar ones may introduce noise to decoding. In this experiment, we explore the similarity threshold that a TM can help or not. We decoded the UNPC En-Zh test set using NiuTrans' WMT20 En-Zh model (Zhang et al. 2020). Specifically, the test set is divided into various portions based on the similarity score, and each portion of the test set is decoded individually both with the sentence-level TM approach and baseline (without TM).

From the experimental results in Figure 3, we can find that the BLEU scores of our method are lower than the baseline when the similarity score is lower than 0.6; when it is between 0.6 and 0.8, some of our concatenation methods perform better than the baseline with a marginal advantage; when it is higher than 0.8, all of the concatenation methods outperform the baseline significantly, with a maximum improvement of 7.09 and 6.11 BLEU points, respectively. Therefore, the FMS threshold for sentence-level TM of the NiuTrans WMT20 En-Zh model is 0.8.

The threshold is determined by the "strength" of the NMT model that "strong" models are more robust to obtaining useful information and avoiding noises introduced by less similar TMs. Thus, "strong" models have lower FMS thresholds. In practice, the threshold can be used to decide whether to apply TM for decoding.

Sentence Length

In this section, we investigate the impact of input sentence length on translation. To avoid the TM similarity influencing the experimental results, we used the test set itself as the retrieved TM, which means that the TMs are 100% similar to the input sentences. We split the test set into groups according to the length of the input sentence, and uniformly chose 170 sentences from each group as the test set (the minimum sentence number of the original groups is 171). The experimental results are given in Figure 4.

From the experimental results, we can find a sharp tendency that the performance of our proposed method decreases and eventually be comparable to the baseline as the

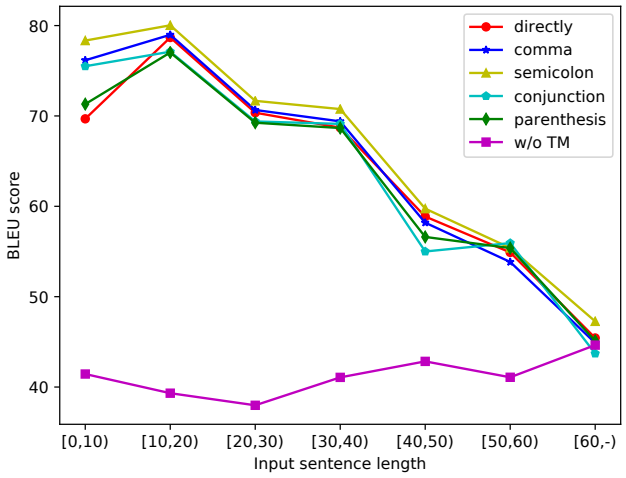

Figure 4: input sentence length impact on BLEU.

sentence length increases. The performance of the baseline, however, tends to be steady. This is also determined by the initial design of our method. For an input sentence, we employ an NMT model that has been trained on a sentence-level dataset. As we concatenate the source TM with the input sentence before feeding them into the NMT model, the length of the input for the model encoder will be doubled. In this way, the whole sentence length of a lengthy sentence and its corresponding TM will deviate from the training model's sentence length distribution. This is the main cause of our method's performance in the figure degrading on lengthy sentences. Therefore, our method can not handle long sentences well even though a highly similar sentence is retrieved. After calculation, we find that the average sentence length of the UNPC En-Zh test set and homemade En-Zh test set in Table 4 are 29.58 and 10.99. This is why the latter can improve the translation more significantly than the former even though there are fewer sentences with high similarity than there are in the former.

Related Work

Many studies have been conducted in recent years to enhance MT quality using TMs. With the emergence of NMT, the MT community is seeing an increasing interest in TM research. There are mainly two research lines for TM integration into NMT: constraining the decoding process with TM and using TM to train a more powerful NMT model.

The main idea of the first research line is to increase the generation probability of some target words based on the retrieved TM. Zhang et al. (2018) constrained the decoding process by increasing the generation possibility of the target words which are in the aligned slices extracted from retrieved TM. Following this work, He et al. (2019) added positional information for words in the TM slices. Unlike the above approaches, Li, Zhang, and Zong (2016) and Farajian et al. (2017) embedded the retrieved TM information into the NMT model by fine-tuning the NMT model with TMs before translating the input sentence. Instead of incorporating sentence level TM, the recent work – kNN-MT – retrieved TM from dense vectors (Khandelwal et al. 2020). First, they created a key-value datastore from the TM database, where the key is the translation context vector of each time step, and the value is the true target token. In the inference time, kNN-MT interpolates the generation probability of the NMT model and retrieved similar target distribution from that datastore at each time step. Following kNN-MT, several researches optimized kNN-MT from different perspectives. Meng et al. (2022) accelerated the inference process by narrowing the search range, instead of searching from the entire data store. By introducing a lightweight meta-k network, Zheng et al. (2021) dynamically determines how many neighbors should be introduced. Wang et al. (2021a) further accelerated kNN-MT inference by constraining search space when constructing the data store. Instead of retrieving a single token, Martins, Marinho, and Martins (2022) retrieved chunks of tokens from the data store to speed up kNN-MT.

The second research line aims to train the generation model to learn how to deal with the retrieved TMs. Bulté and Tezcan (2019) and Xu, Crego, and Senellart (2020) used a data augmentation way to concatenate the retrieved TM with input sentence during training. While some researches modified the NMT model architecture to better integrate TMs. Cao and Xiong (2018) and Gu et al. (2018) introduced a gating mechanism module to control the signal from the retrieved TM. Cao, Kuang, and Xiong (2020) designed an additional transformer encoder to encode the target sentence of retrieved TM, and integrate them through the attention mechanism. In Xia et al. (2019), the retrieved multiple TMs are compressed into a graph structure for speed up and space savings and then are integrated into the model via the attention mechanism. He et al. (2021) proposed a lightweight method to incorporate the target sentence of retrieved TM in an extra attention module. Unlike all of the above methods, Cai et al. (2021) proposed a method to incorporate monolingual TM into NMT, and the target sentence retriever and NMT model are trained jointly.

Conclusion and Future Work

In this paper, we propose a simple but effective method to incorporate TM into NMT decoding without modifying the pre-trained NMT model. Specifically, we treat the retrieved TMs as prompts for the translation of the input sentence by concatenating the source TM with the input sentence and generating the target token in a forced way. Experiments on the TM specialized translation task, domain adaptation task, and implementation on commercial MT system verify the effectiveness of our method. Our method is easy to implement and can be applied to customize a TM-incorporated machine translation system for TM data on the user side. Our method in this paper suffers from TM sentences with low similarity scores and long sentences. In the future, we will investigate more effective methods to alleviate the drawbacks of our methods in low similarity TM and long sentence translation situations.

Acknowledgments

This work was supported by National Key R&D Program of China (No. 2020AAA0107904). We are very thankful to anonymous reviewers for their valuable comments.

References

Aharoni, R.; and Goldberg, Y. 2020. Unsupervised Domain Clusters in Pretrained Language Models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 7747–7763. Association for Computational Linguistics.

Bapna, A.; and Firat, O. 2019. Non-Parametric Adaptation for Neural Machine Translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, June 2-7, 2019, Volume 1 (Long and Short Papers)*, 1921–1931. Association for Computational Linguistics.

Bialecki, A.; Muir, R.; and Ingersoll, G. 2012. Apache Lucene 4. In *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval, OSIR@SIGIR 2012, Portland, Oregon, USA, 16th August 2012*, 17–24. University of Otago, Dunedin, New Zealand.

Brown, T. B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; Agarwal, S.; Herbert-Voss, A.; Krueger, G.; Henighan, T.; Child, R.; Ramesh, A.; Ziegler, D. M.; Wu, J.; Winter, C.; Hesse, C.; Chen, M.; Sigler, E.; Litwin, M.; Gray, S.; Chess, B.; Clark, J.; Berner, C.; McCandlish, S.; Radford, A.; Sutskever, I.; and Amodei, D. 2020. Language Models are Few-Shot Learners. In *Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual*.

Bulté, B.; and Tezcan, A. 2019. Neural Fuzzy Repair: Integrating Fuzzy Matches into Neural Machine Translation. In *Proceedings of the 57th Conference of the Association for Computational Linguistics, ACL 2019, Florence, Italy, July 28- August 2, 2019, Volume 1: Long Papers*, 1800–1809. Association for Computational Linguistics.

Cai, D.; Wang, Y.; Li, H.; Lam, W.; and Liu, L. 2021. Neural Machine Translation with Monolingual Translation Memory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 7307–7318. Association for Computational Linguistics.

Cao, Q.; Kuang, S.; and Xiong, D. 2020. Learning to Reuse Translations: Guiding Neural Machine Translation with Examples. In *ECAI 2020 - 24th European Conference on Artificial Intelligence, 29 August-8 September 2020, Santiago de Compostela, Spain, August 29 - September 8, 2020 - Including 10th Conference on Prestigious Applications of Artificial Intelligence (PAIS 2020)*, volume 325 of *Frontiers in Artificial Intelligence and Applications*, 1982–1989. IOS Press.

Cao, Q.; and Xiong, D. 2018. Encoding Gated Translation Memory into Neural Machine Translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing, Brussels, Belgium, October 31 - November 4, 2018*, 3042–3047. Association for Computational Linguistics.

Chen, C.; Sun, M.; and Liu, Y. 2021. Mask-Align: Self-Supervised Neural Word Alignment. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 4781–4791. Association for Computational Linguistics.

Dillon, S.; and Fraser, J. 2006. Translators and TM: An investigation of translators' perceptions of translation memory adoption. *Mach. Transl.*, 20(2): 67–79.

Farajian, M. A.; Turchi, M.; Negri, M.; and Federico, M. 2017. Multi-Domain Neural Machine Translation through Unsupervised Adaptation. In *Proceedings of the Second Conference on Machine Translation, WMT 2017, Copenhagen, Denmark, September 7-8, 2017*, 127–137. Association for Computational Linguistics.

Gu, J.; Wang, Y.; Cho, K.; and Li, V. O. K. 2018. Search Engine Guided Neural Machine Translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, 5133–5140. AAAI Press.

He, Q.; Huang, G.; Cui, Q.; Li, L.; and Liu, L. 2021. Fast and Accurate Neural Machine Translation with Translation Memory. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, August 1-6, 2021*, 3170–3180. Association for Computational Linguistics.

He, Q.; Huang, G.; Liu, L.; and Li, L. 2019. Word Position Aware Translation Memory for Neural Machine Translation. In *Natural Language Processing and Chinese Computing - 8th CCF International Conference, NLPCC 2019, Dunhuang, China, October 9-14, 2019, Proceedings, Part I*, volume 11838 of *Lecture Notes in Computer Science*, 367–379. Springer.

Hossain, N.; Ghazvininejad, M.; and Zettlemoyer, L. 2020. Simple and Effective Retrieve-Edit-Rerank Text Generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 2532–2538. Association for Computational Linguistics.

Khandelwal, U.; Fan, A.; Jurafsky, D.; Zettlemoyer, L.; and Lewis, M. 2020. Nearest Neighbor Machine Translation. *CoRR*, abs/2010.00710.

Koehn, P.; Hoang, H.; Birch, A.; Callison-Burch, C.; Federico, M.; Bertoldi, N.; Cowan, B.; Shen, W.; Moran, C.; Zens, R.; Dyer, C.; Bojar, O.; Constantin, A.; and Herbst, E.

2007. Moses: Open Source Toolkit for Statistical Machine Translation. In *ACL 2007, Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics, June 23-30, 2007, Prague, Czech Republic*. The Association for Computational Linguistics.

Li, L.; Way, A.; and Liu, Q. 2014. A discriminative framework of integrating translation memory features into SMT. In *Proceedings of the 11th Conference of the Association for Machine Translation in the Americas: MT Researchers Track, AMTA 2014, Vancouver, Canada, October 22-26, 2014*, 249–260. Association for Machine Translation in the Americas.

Li, X.; Zhang, J.; and Zong, C. 2016. One Sentence One Model for Neural Machine Translation. *CoRR*, abs/1609.06490.

Li, Y.; and Liu, B. 2007. A Normalized Levenshtein Distance Metric. *IEEE Trans. Pattern Anal. Mach. Intell.*, 29(6): 1091–1095.

Ma, Y.; He, Y.; Way, A.; and van Genabith, J. 2011. Consistent Translation using Discriminative Learning - A Translation Memory-inspired Approach. In *The 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies, Proceedings of the Conference, 19-24 June, 2011, Portland, Oregon, USA*, 1239–1248. The Association for Computer Linguistics.

Martins, P. H.; Marinho, Z.; and Martins, A. F. T. 2022. Chunk-based Nearest Neighbor Machine Translation. *CoRR*, abs/2205.12230.

Meng, Y.; Li, X.; Zheng, X.; Wu, F.; Sun, X.; Zhang, T.; and Li, J. 2022. Fast Nearest Neighbor Machine Translation. In *Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, May 22-27, 2022*, 555–565. Association for Computational Linguistics.

Ng, N.; Yee, K.; Baevski, A.; Ott, M.; Auli, M.; and Edunov, S. 2019. Facebook FAIR's WMT19 News Translation Task Submission. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, 314–319. Association for Computational Linguistics.

Sennrich, R.; Haddow, B.; and Birch, A. 2016. Neural Machine Translation of Rare Words with Subword Units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A. N.; Kaiser, L.; and Polosukhin, I. 2017. Attention is All you Need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, 5998–6008.

Wang, K.; Zong, C.; and Su, K. 2013. 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, 4-9 August 2013, Sofia, Bulgaria, Volume 1: Long Papers*, 11–21. The Association for Computer Linguistics.

Wang, S.; Li, J.; Meng, Y.; Ouyang, R.; Wang, G.; Li, X.; Zhang, T.; and Zong, S. 2021a. Faster Nearest Neighbor Machine Translation. *CoRR*, abs/2112.08152.

Wang, Y.; Yao, Q.; Kwok, J. T.; and Ni, L. M. 2021b. Generalizing from a Few Examples: A Survey on Few-shot Learning. *ACM Comput. Surv.*, 53(3): 63:1–63:34.

Xia, M.; Huang, G.; Liu, L.; and Shi, S. 2019. Graph Based Translation Memory for Neural Machine Translation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, 7297–7304. AAAI Press.

Xiao, T.; Zhu, J.; Zhang, H.; and Li, Q. 2012. NiuTrans: An Open Source Toolkit for Phrase-based and Syntax-based Machine Translation. In *The 50th Annual Meeting of the Association for Computational Linguistics, Proceedings of the System Demonstrations, July 10, 2012, Jeju Island, Korea*, 19–24. The Association for Computer Linguistics.

Xu, J.; Crego, J. M.; and Senellart, J. 2020. Boosting Neural Machine Translation with Similar Translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, July 5-10, 2020*, 1580–1590. Association for Computational Linguistics.

Zhang, J.; Utiyama, M.; Sumita, E.; Neubig, G.; and Nakamura, S. 2018. Guiding Neural Machine Translation with Retrieved Translation Pieces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, June 1-6, 2018, Volume 1 (Long Papers)*, 1325–1335. Association for Computational Linguistics.

Zhang, Y.; Wang, Z.; Cao, R.; Wei, B.; Shan, W.; Zhou, S.; Reheman, A.; Zhou, T.; Zeng, X.; Wang, L.; Mu, Y.; Zhang, J.; Liu, X.; Zhou, X.; Li, Y.; Li, B.; Xiao, T.; and Zhu, J. 2020. The NiuTrans Machine Translation Systems for WMT20. In *Proceedings of the Fifth Conference on Machine Translation, WMT@EMNLP 2020, Online, November 19-20, 2020*, 338–345. Association for Computational Linguistics.

Zheng, X.; Zhang, Z.; Guo, J.; Huang, S.; Chen, B.; Luo, W.; and Chen, J. 2021. Adaptive Nearest Neighbor Machine Translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, August 1-6, 2021*, 368–374. Association for Computational Linguistics.