Fast and Accurate Neural Machine Translation with Translation Memory

Qiuxiang He[1] **Guoping Huang**[2] **Qu Cui**[3] **Li Li**[1*] **Lemao Liu**[2]
[1] Southwest University [2] Tencent AI Lab [3] Nanjing University
hqxiang@email.swu.edu.cn cuiq@smail.nju.edu.cn
{donkeyhuang,redmondliu}@tencent.com lily@swu.edu.cn

Abstract

It is generally believed that a translation memory (TM) should be beneficial for machine translation tasks. Unfortunately, existing wisdom demonstrates the superiority of TM-based neural machine translation (NMT) only on the TM-specialized translation tasks rather than general tasks, with a non-negligible computational overhead. In this paper, we propose a fast and accurate approach to TM-based NMT within the Transformer framework: the model architecture is simple and employs a single bilingual sentence as its TM, leading to efficient training and inference; and its parameters are effectively optimized through a novel training criterion. Extensive experiments on six TM-specialized tasks show that the proposed approach substantially surpasses several strong baselines that use multiple TMs, in terms of BLEU and running time. In particular, the proposed approach also advances the strong baselines on two general tasks (WMT news Zh→En and En→De).

1 Introduction

A translation memory (TM) is originally collected from the translation history of professional translators, and provides the most similar source-target sentence pairs for the source sentence to be translated (Garcia, 2009; Koehn and Senellart, 2010b; Utiyama et al., 2011; Robinson, 2012; Huang et al., 2021). A TM generally provides valuable translation information particularly for those input sentences preferably matching the source sentences in the TM, and many efforts have been devoted to integrating a TM into statistical machine translation (Simard and Isabelle, 2009; Koehn and Senellart, 2010a; Ma et al., 2011; Wang et al., 2013; Liu et al., 2019).

Recently there are increasing interests in improving neural machine translation (NMT) with a TM (Li et al., 2016; Farajian et al., 2017; Gu et al., 2018; Xia et al., 2019; Bulte and Tezcan, 2019; Xu et al., 2020). Many notable approaches have been proposed to augment an NMT model by using a TM. For example, Zhang et al. (2018) and He et al. (2019) extract scored n-grams from a TM and then reward each partial translation once it matches an extracted n-gram during beam search. Gu et al. (2018) and Xia et al. (2019) use an auxiliary network to encode a TM and then integrate it into the NMT architecture. Bulte and Tezcan (2019) and Xu et al. (2020) employ data augmentation to train an NMT model whose training instances are bilingual sentences augmented by their TMs. Despite their improvements on the TM-specialized translation tasks (aka JRC-Acquis corpora) where a TM is very similar to test sentences, they consume considerable computational overheads in either training or testing, and particularly it is unclear whether they can deliver gains over standard NMT on general tasks where a TM is not very similar to test sentences. Indeed, both Zhang et al. (2018) and Xu et al. (2020) reported their failures on WMT news translation tasks.

In this paper, we present a fast and accurate approach for TM-based NMT which can be applied to general translation tasks besides TM-specialized tasks. We first design a light-weight TM-based NMT model for efficiency: its TM includes a single bilingual sentence and we explore variant ways to encode the TM. Also, the designed model outperforms strong TM-based baselines. Second, we deeply analyze its translation performance and observe an issue of *robustness*: it decreases significantly for those input sentences which are not very similar to their TMs, although it obtains substantial improvements for other inputs. To address this issue, we propose a novel training criterion for optimizing the parameters of our model inspired by multiple-task learning (van Dyk and Meng, 2001;

*Corresponding author.

Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing, pages 3170–3180
August 1–6, 2021.

Ben-David and Borbely, 2008; Qiu et al., 2013). The loss function includes two terms: the first term is induced by the bilingual corpus with a TM whereas the second term is induced by the bilingual corpus without any TM. In this way, the TM-based NMT model gains better performance and is robust to translate any input sentences no matter they are similar to their TM or not. Additionally, this makes it possible that a single unified model can handle both translation situations (with or without a TM), which is practical for online services.

To validate the effectiveness of the proposed approach, we conduct extensive experiments on eight translation tasks including both TM-specialized tasks and general tasks (WMT). Our experiments justify that the proposed approach is better than several strong TM-based baselines in speed, and it further delivers substantial gains (up to 4.7 BLUE points) over those baselines on TM-specialized tasks, leading to up to 8.5 BLEU points over standard Transformer-based NMT. In particular, it also outperforms strong baselines on two general translation tasks, i.e., with a gain of 0.7 BLEU points on WMT14 En→De task and 1.0 BLEU point on WMT17 Zh→En task.

This paper makes the following contributions:

- It points out a critical issue about robustness when training TM-based NMT models and provides an elegant method to address this issue.
- It proposes a simple TM-based NMT model that outperforms strong TM-based baselines in terms of both translation quality and speed.
- It verifies that a well-designed TM-based translation model is able to advance strong MT baselines on general translation tasks where a TM is not very similar to input source sentences.

2 Preliminary on NMT

Suppose $\mathbf{x} = \{x_1, ..., x_n\}$ is a source sentence and $\mathbf{y} = \{y_1, ..., y_m\}$ is the corresponding target sentence. From the probabilistic perspective, NMT models the conditional probability of the target sentence $\mathbf{y}$ given the source sentence $\mathbf{x}$. Formally, for a given $\mathbf{x}$, NMT aims to generate the output $\mathbf{y}$ according to the conditional probability $P(\mathbf{y}|\mathbf{x})$ defined by neural networks:

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

where $\mathbf{y}_{<i} = \{y_1, \ldots, y_{i-1}\}$ denotes a prefix of $\mathbf{y}$, and each factor $P(y_i|\mathbf{x}, \mathbf{y}_{<i})$ is defined as follows:

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

where $h_i^{D,L}$ indicates the i_{th} hidden unit at L_{th} layer in the ***D***ecoding phrase under the encoder-decoder framework (Bahdanau et al., 2016), and ϕ is a linear network that projects hidden units onto vectors with dimension of the target vocabulary.

Recently, self-attention networks have attracted many interests due to their flexibility in parallel computation and modeling $h_i^{D,L}$. The state-of-the-art NMT model is Transformer (Vaswani et al., 2017), which uses stacked self-attention and fully connected layers for its encoder and decoder. Self-attention relies on an attention mechanism to compute a representation of a sequence. In Transformer, there are three kinds of attention mechanisms, including encoder multi-head attention, decoder masked multi-head attention and encoder-decoder multi-head attention. Attention with H heads can be calculated by the equations:

$$\begin{aligned} \text{MH-Att}(q, \mathbf{u}) &= \Big[\text{Att}(q, \phi_j(\mathbf{u}), \psi_j(\mathbf{u}))\Big]_{j=1}^{H}, \\ \text{Att}(q, \mathbf{u}, \mathbf{v}) &= \text{softmax}\left(\frac{q\mathbf{u}^\top}{\sqrt{d}}\right)\mathbf{v} \end{aligned} \quad (3)$$

where q is a query vector and $\mathbf{u}$ is a two-dimensional matrix, $[u_j]_{j=1}^{H}$ denotes concatenation of all vectors u_j, ϕ_j and ψ_j stand for two linear projections from one matrix to another matrix, respectively. The $\frac{1}{\sqrt{d}}$ is the scaling factor, and d is the dimension of q. And we refer enthusiastic readers to Vaswani et al. (2017) for detailed definitions.

3 Model Architecture

In this section, in order to preferably bridge TM and NMT, we propose the architecture of TM-based NMT within the Transformer. To make our proposed model fast in running time and powerful in quality, at first, we present a configuration of TM to make the proposed model efficient. Then we explore three different methods to encode the TM into a sequence of vectors in a coarse-to-fine manner. Finally, we propose the architecture that decodes a target word given an input source sentence and its TM representation.

3.1 TM Configuration

Following previous works (Gu et al., 2018; Zhang et al., 2018; Xia et al., 2019), for each source

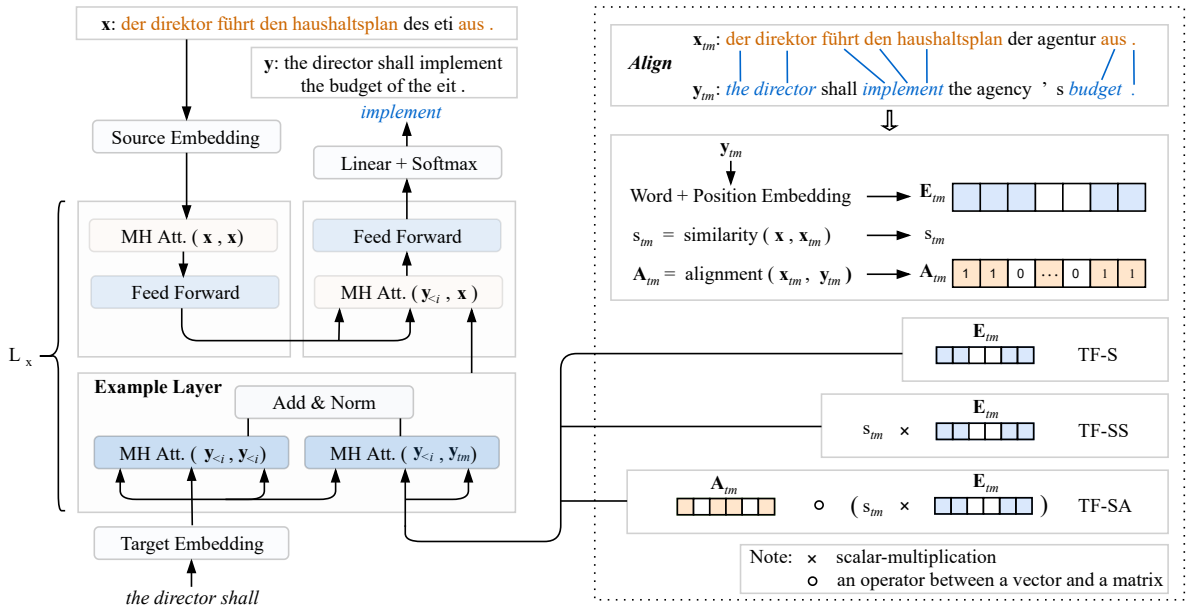

Figure 1: The architecture of the proposed three methods. 1. The part in the dashed box is an example of our methods. The source and target languages are German and English respectively. **x** is the source sentence and **y** is the corresponding target sentence. 2. The part outside of the dashed box shows the whole model architecture. The core component is the Example Layer which consists of multi-head attention and cross attention mechanisms. For simplicity, we omit an add and layer normalization in other sub-layers. L is the number of operation layers.

sentence **x** we employ Apache Lucene (Bialecki et al., 2020) to retrieve top-100 similar bilingual sentences from the training data. Then we adopt the following similarity to re-rank the retrieved bilingual sentences and maintain top-K ($K < 100$) bilingual sentences as the TM for **x**:

$$\text{sim}(\mathbf{x}, \mathbf{x}_{tm}) = 1 - \frac{dist(\mathbf{x}, \mathbf{x}_{tm})}{\max(|\mathbf{x}|, |\mathbf{x}_{tm}|)} \quad (4)$$

where $dist$ denotes the edit-distance, and $\mathbf{x}_{tm}$ is a retrieved source sentence from the training data and its reference is $\mathbf{y}_{tm}$.

Previous studies show that the best translation quality is achieved when the size K of the TM is larger than 1. For example, the optimized K is set to be 5 in Gu et al. (2018) and Xia et al. (2019), and it is even set to be 100 in Zhang et al. (2018). Unfortunately, such a large K significantly decreases the translation speed because the computational complexity is linear in the size of K. To make our inference as efficient as possible, we set $K = 1$ and employ the most similar bilingual sentence denoted by $\langle \mathbf{x}_{tm}, \mathbf{y}_{tm} \rangle$ as the TM for **x**.[1]

3.2 Encoding TM

In this subsection, we will describe how to encode the TM $\langle \mathbf{x}_{tm}, \mathbf{y}_{tm} \rangle$ into a sequence of vectors **m**. Three variant methods for encoding a TM are illustrated in the right part of Figure 1.

Method 1: sentence (TF-S) Given $\langle \mathbf{x}_{tm}, \mathbf{y}_{tm} \rangle$ for **x**, the first method utilizes word embedding and position embedding of $\mathbf{y}_{tm}$ to represent **m** as follows:

$$\begin{aligned} \mathbf{m} = \mathbf{E}_{tm} = [&E_w(y^1_{tm}) + E_p(y^1_{tm}), \\ &\cdots, E_w(y^{J'}_{tm}) + E_p(y^{J'}_{tm})] \end{aligned} \quad (5)$$

where E_w and E_p are word embedding and position embedding respectively, J' is the length of $\mathbf{y}_{tm}$ and the symbol + denotes a simple addition operator.

Method 2: sentence with score (TF-SS) The first method is agnostic to the similarity score. Intuitively, if a TM $\langle \mathbf{x}_{tm}, \mathbf{y}_{tm} \rangle$ is with high similarity, $\mathbf{y}_{tm}$ may be more helpful to predict a good translation. So, the second method takes the similarity score into account and it defines **m** as follows:

$$\mathbf{m} = s_{tm} \times \mathbf{E}_{tm} \quad (6)$$

where $s_{tm} = \text{sim}(\mathbf{x}, \mathbf{x}_{tm})$ is the similarity score and the symbol $\times$ denotes the scalar-multiplication.

Method 3: sentence with alignment (TF-SA) As shown in Figure 1, $\mathbf{x}_{tm}$ consists of the matched parts (in orange color) and the unmatched parts (in dark color) to **x**. Since each word in the TM is not of the same importance to the source sentence **x**, we should pay more attention to the words that

[1] We also did some experiments on $K = 2$ and $K = 4$ in our proposed model, but we did not observe significant gains.

are in the matched parts. So, we further obtain the word alignment between $\mathbf{x}_{tm}$ and $\mathbf{y}_{tm}$ through fast-align toolkit (Dyer et al., 2013).[2] Suppose $\mathbf{A}_{tm}$ is the word alignment between $\mathbf{x}_{tm}$ and $\mathbf{y}_{tm}$: $A^j_{tm} = 1$ denotes y_j is aligned to some x_i otherwise $A^j_{tm} = 0$, where x_i is also in $\mathbf{x}$. Therefore, the third method defines $\mathbf{m}$ as follows:

$$\mathbf{m} = \mathbf{A}_{tm} \circ \left(s_{tm} \times \mathbf{E}_{tm}\right) \quad (7)$$

where the symbol $\circ$ denotes an operator between a vector and a matrix such that

$$m_j = \begin{cases} s_{tm} \times E^j_{tm} & \text{if } A^j_{tm} = 0 \\ E^j_{tm} & \text{if } A^j_{tm} = 1 \end{cases} \quad (8)$$

3.3 TM Augmented NMT

Suppose the encoded TM $\langle \mathbf{x}_{tm}, \mathbf{y}_{tm} \rangle$ is denoted by $\mathbf{m}$, a sequence of vectors. We aim to build a model $P(y_i \mid \mathbf{x}, \mathbf{y}_{<i}, \mathbf{m})$ for the source sentence $\mathbf{x}$, given the $\mathbf{m}$ and prefix translation $\mathbf{y}_{<i}$ at time step i, leading to the entire translation model:

$$P(\mathbf{y} \mid \mathbf{x}, \mathbf{x}_{tm}, \mathbf{y}_{tm}; \theta) = \prod_i P(y_i \mid \mathbf{x}, \mathbf{y}_{<i}, \mathbf{m}) \quad (9)$$

where θ denotes the parameter of our proposed model.[3]

Example Layer The model architecture of $P(y_i \mid \mathbf{x}, \mathbf{y}_{<i}, \mathbf{m})$ is illustrated at the left part of Figure 1, where its architecture is generally similar to standard Transformer and the core component is the Example Layer. Specifically, the Example Layer includes two multi-head attention operators: the left multi-head attention (i.e. MH-Att $(\mathbf{y}_{<i}, \mathbf{y}_{<i})$) is the same as Transformer, and it is defined on the prefix translation $\mathbf{y}_{<i}$; the right multi-head attention (i.e. MH-Att $(\mathbf{y}_{<i}, \mathbf{y}_{tm})$) attempts to capture information from the TM, and its query is from $\mathbf{y}_{<i}$ while key and value are from the representation of TM $\mathbf{m}$. After the two parallel attention operators, two resulting sequences are passed to Add & Norm operator and a new sequence is obtained as the query for the next multi-head attention (i.e. MH-Att $(\mathbf{y}_{<i}, \mathbf{x})$). The following sub-layer is the same as Transformer and $P(y_i \mid \mathbf{x}, \mathbf{y}_{<i}, \mathbf{m})$ can be obtained similar to the definition of standard NMT $P(y_i \mid \mathbf{x}, \mathbf{y}_{<i})$ as presented in Section 2. We skip those formal equations to rewrite $P(y_i \mid \mathbf{x}, \mathbf{y}_{<i}, \mathbf{m})$ due to space limitation.

[2] Although some advanced word alignment toolkits (Dou and Neubig, 2021; Chen et al., 2021; Jalili Sabet et al., 2020) may lead to better performance, we still employ fast-align to be in line with previous work for fair comparison (Zhang et al., 2018; Xia et al., 2019).

[3] In the rest of this paper, we may drop θ in the model for easier notations.

In summary The entire model architecture is illustrated in Figure 1: the dashed box in the right part shows the memory encoder, and the left part shows how the memory representation is used in the NMT model similar to the Transformer. In our model architecture, the encoder block contains two sub-layers and the decoder block contains three sub-layers. The core sub-layer in the decoder block is our proposed Example Layer, which consists of multi-head attention and cross attention. By introducing the memory encoder and Example Layer, the parameters in our model are increased only by 8.96% compared to the standard NMT baseline.

4 Training

Suppose the training corpus is $\mathcal{D} = \{\langle \mathbf{x}^i, \mathbf{y}^i, \mathbf{x}^i_{tm}, \mathbf{y}^i_{tm} \rangle \mid i \in [1, N]\}$, where $\langle \mathbf{x}^i, \mathbf{y}^i \rangle$ is a bilingual sentence, and $\langle \mathbf{x}^i_{tm}, \mathbf{y}^i_{tm} \rangle$ is the related TM which consists of a single bilingual sentence. Our goal is to learn the parameter θ of the TM-based NMT model $P(\mathbf{y} \mid \mathbf{x}, \mathbf{x}_{tm}, \mathbf{y}_{tm}; \theta)$ defined in Eq.(9) using $\mathcal{D}$.

The common wisdom is to optimize the parameter under the maximum likelihood estimation (MLE), i.e. standard training. Formally, it minimizes the following criterion:

$$-\sum_i^N \log P(\mathbf{y}^i \mid \mathbf{x}^i, \mathbf{x}^i_{tm}, \mathbf{y}^i_{tm}; \theta).$$

Robustness issue Unfortunately, the model trained with MLE suffers from an issue about robustness even if its overall performance is much better than standard Transformer and outperforms TM-based baselines on the Es→En task. According to our experiments (see Table 4 later), our proposed model performs worse than the Transformer for those sentences which do not have a similar TM. As a result, it would be dangerous to use the model for online services because users may provide an input sentence whose TM is not similar to itself.

The possible reason for the above issue is explained as follows. On the average case, the reference $\mathbf{y}$ is strongly correlated to its TM target $\mathbf{y}_{tm}$ in the training corpus $\mathcal{D}$. For example, the average similarity score is about 0.58 for Es→En translation task, according to our statistics. Because of the powerful fitting ability of neural networks, the model parameters will be guided to heavily depend on the given TM target $\mathbf{y}_{tm}$ during training. In this way, if an input source sentence $\mathbf{x}$ has a high similarity with its given TM, the model will output

high-quality results, as we also observed in Table 5. On the contrary, once an input sentence is provided with a low similar TM $\langle \mathbf{x}_{tm}, \mathbf{y}_{tm} \rangle$ (for instance, the similarity between 0 and 0.3, as shown in Table 4), the translation quality of its output rapidly decreases.

Training criterion In order to avoid the TM over-fitting, we propose a simple yet elegant method, inspired by data augmentation (van Dyk and Meng, 2001; Li et al., 2019; Zhong et al., 2020) and multiple-task learning (Ben-David and Borbely, 2008; Qiu et al., 2013; Liu et al., 2016). Specifically, we first construct another corpus $\mathcal{D}_0 = \{\langle \mathbf{x}^i, \mathbf{y}^i, \text{null}, \text{null} \rangle \mid i \in [1, N]\}$ from $\mathcal{D} = \{\langle \mathbf{x}^i, \mathbf{y}^i, \mathbf{x}^i_{tm}, \mathbf{y}^i_{tm} \rangle \mid i \in [1, N]\}$. In the constructed corpus, $\langle \text{null}, \text{null} \rangle$ plays a role of a TM, but both source and target sides of the TM are empty sentences.[4] Then we train the model $P(\mathbf{y} \mid \mathbf{x}, \mathbf{x}_{tm}, \mathbf{y}_{tm}; \theta)$ using both $\mathcal{D}$ and $\mathcal{D}_0$, i.e. joint training, which is similar to multiple-task learning. Formally, we minimize the following joint loss function:

$$\ell(\mathcal{D}, \mathcal{D}_0; \theta) = -\sum_{i}^{N} \Big(\log P(\mathbf{y}^i \mid \mathbf{x}^i, \mathbf{x}^i_{tm}, \mathbf{y}^i_{tm}; \theta) + \lambda \times \log P(\mathbf{y}^i \mid \mathbf{x}^i, \text{null}, \text{null}; \theta) \Big) \quad (10)$$

where $0 < \lambda$ is a coefficient to trade off both loss terms. Intuitively, the first term induced by $\mathcal{D}$ guides the model to use the information from a TM for prediction, and thereby it will generate accurate translations for those input source sentences whose TM is with high similarity. On the other hand, the second term induced by $\mathcal{D}_0$ teaches the model to output good translations without information from a TM. Additionally, this makes it possible that a single unified model can handle both translation scenarios (with or without a TM), which is practical for online services.

Note that the proposed method is slightly different from standard data augmentation (Sennrich et al., 2016a; Fadaee et al., 2017; Fadaee and Monz, 2018; Wang et al., 2018) and multiple-task learning (Dong et al., 2015; Kiperwasser and Ballesteros, 2018; Wang et al., 2020) in NMT research. These data augmentation techniques automatically generate pseudo data based on the original training data and then train a model using both original and generated data. However, the dataset $\mathcal{D}_0$ is directly taken from the original $\mathcal{D}$ in our scenario. Also, multiple-task learning in their works typically involves different models that share some partial parameters rather than all parameters. In contrast, both terms in our joint loss correspond to the same task, i.e. translation prediction given a source sentence and its TM; and both models are exactly the same.

[4] In the experiments, we implement null as the sentence including a single word, i.e. "⟨eos⟩".

Algorithm 1: Joint Training Algorithm

Input: Mini-batch size b, maximal iteration M, a learning rate schema η and two corpus: $\mathcal{D} = \{\langle \mathbf{x}^i, \mathbf{y}^i, \mathbf{x}^i_{tm}, \mathbf{y}^i_{tm} \rangle \mid i \in [1, N]\}$ and $\mathcal{D}_0 = \{\langle \mathbf{x}^i, \mathbf{y}^i, \text{null}, \text{null} \rangle \mid i \in [1, N]\}$

Output: The parameter θ.

1 **for** $1 \le t \le M$ **do**
2 Sample a mini-batch $\mathcal{B}$ with size of $b/2$ from $\mathcal{D}$
3 Sample a mini-batch $\mathcal{B}_0$ with size of $b/2$ from $\mathcal{D}_0$
4 Calculate gradient $\Delta = \nabla_\theta \ell(\mathcal{B}, \mathcal{B}_0; \theta)$ as defined in Eq.(10)
5 Update parameter: $\theta = \theta - \eta_t \Delta$

The detailed joint training algorithm is presented in Algorithm 1. It follows the standard gradient descent method for optimization. Note that in line 2 and 3, it samples two mini-batches which do not share the same bilingual sentences to promote diversity, i.e., $\mathcal{D}$ and $\mathcal{D}_0$ are independently and randomly sampled. In our experiments, we employ Adam (Kingma and Ba, 2014) with default settings as the learning rate schema.

5 Experiments

In this section, we validate the effectiveness of the proposed approach: robustness for handling both translation situations (with or without a TM), running efficiency compared with the previous TM-based NMT models, translation quality on both TM-specialized tasks and general MT tasks. We use the case-insensitive BLEU score as the automatic metric (Papineni et al., 2002) for the translation quality evaluation.

5.1 Setup

TM-specialized tasks We evaluate our proposed models with the JRC-Acquis corpora, which include three language pairs and lead to six translation tasks in total: English↔German (En↔De),

	TM-specialized Tasks			General WMT Tasks	
	Fr↔En	Es↔En	De↔En	En→De	Zh→En
Train/Sent(#)	740467	673856	693011	4558262	20605452
Dev/Sent(#)	2649	2511	2440	3000	2002
Test/Sent(#)	2650	2585	2461	3003/3004	2001
En/Word(#)	29.44	32.68	34.00	28.94	25.46
Other/Word(#)	33.35	35.58	34.22	29.90	23.03

Table 1: Statistics of the datasets. The last two lines are average sentence lengths in English and other languages.

	TM-specialized Tasks	General WMT Tasks	
	All	En→De	Zh→En
Layers	6	6	6
Dropout	0.1	0.1	0.1
Embedding	512	512	512
Batch size	3500	2500	4096
Source vocab	20000	32000	75000
Target vocab	20000	32000	63000

Table 2: Training settings. Batch size refers to the token number for each batch. Embedding refers to the number of word embedding dimensions. For a fair comparison, the source vocabulary size is 40000 in baseline FM$^+$ on Es→En task.

English↔Spanish (En↔Es) and English↔French (En↔Fr). To compare with previous work, we adopt the same splitting of training/dev/test and pre-processing as Gu et al. (2018), Zhang et al. (2018), and Xia et al. (2019).

General tasks The proposed models are evaluated on the widely-used general WMT tasks: WMT14 English-to-German (En→De) and WMT17 Chinese-to-English (Zh→En) tasks. For the En→De task, we use newstest2013 as the development set, as well as employ newstest2014 and newstest2017 as the test sets. For the Zh→En task, we employ newsdev2017 and newstest2017 as the development and test set respectively.

Table 1 summarizes the data statistics for both TM-specialized and general tasks. In addition, we employ Byte Pair Encoding (BPE) (Sennrich et al., 2016b) on all the tasks mentioned before.

BLEU	TF	TF-S	TF-SS	TF-SA
Dev	63.35	65.00	67.04	**67.23**
Test	62.79	65.52	67.04	**67.26**

Table 3: Performance of our models under the standard training criterion. BLEU is reported on Es→En task. **Best** results are highlighted.

Similarity	Sents (#)	Percents (%)	**Baseline** TF	**Std Train** TF-SA	**Joint Train** TF-SA
[0, 0.1)	2	0.08	36.91	64.05	74.48
[0.1, 0.2)	138	5.34	38.53	37.70	39.52
[0.2, 0.3)	462	17.87	47.88	47.07	49.09
[0.3, 0.4)	305	11.80	54.02	54.75	56.19
[0.4, 0.5)	272	10.52	62.29	64.01	66.18
[0.5, 0.6)	206	7.97	65.94	71.32	72.48
[0.6, 0.7)	203	7.85	71.88	79.63	80.08
[0.7, 0.8)	188	7.27	77.20	85.96	86.45
[0.8, 0.9)	377	14.59	79.93	90.71	91.31
[0.9, 1)	432	16.71	81.95	94.60	94.68
[0, 0.3)	602	23.29	45.36	44.45	**46.41**
[0.3, 1)	1983	76.71	70.97	78.22	**79.06**
[0, 1)	2585	100	62.79	67.26	**68.49**

Table 4: Translation accuracy in terms of BLEU on the Es→En task (test set only) for the divided subsets according to the similarity of TM.

Baseline systems We compare our proposed model with the strong baselines as follows:

- **TF** (Vaswani et al., 2017): it is the standard Transformer.
- **TF-P** (Zhang et al., 2018): it is re-implemented on top of Transformer by ourselves.
- **TF-G** (Xia et al., 2019) and **TF-SEQ** (Gu et al., 2018): **TF-SEQ** is a mimic implementation over Transformer by Xia et al. (2019). We report the results from Xia et al. (2019) since they were also implemented over Transformer as comparison.
- **FM**$^+$ (Xu et al., 2020): since Xu et al. (2020) adopt a different split on JRC corpus, the results are not comparable to ours. For a fair comparison, we re-implement a strong model FM$^+$ as a baseline which makes use of the same metric to retrieve a TM as ours and is better than the method in Bulte and Tezcan (2019).

Our models In the case of the three methods proposed in this paper, **TF-S**, **TF-SS** and **TF-SA** refer to the method encoding TM by the sentence, sentence with score, and sentence with alignment, respectively. We optimize their parameters through both standard training and joint training. For joint training, the hyperparameter λ is set to be 1 for all translation tasks.

System configuration For a fair comparison, we employ the same settings to train all baselines and our models, and the learning rate for all models is Adam with the default hyper-parameters. The

details of the settings are shown in Table 2.

5.2 Results and Analysis on Es→En Task

Standard training and robustness issue We first evaluate the proposed models under the standard training criterion. Table 3 shows the comparison among different TM encoding methods for our models. From this table, we can see that our models achieve substantial improvements over Transformer (TF) which does not use any TM, even if our models are simple and only utilize a single bilingual sentence in the TM. TF-SA performs better than TF-S and TF-SS thanks to the fine-grained alignment information encoded in the TM. Also, TF-SA outperforms all TM-based baselines by at least 1.0 BLEU point, compared with Table 6.

In addition, we exploit the influence of our models on the similarity of a TM. We thereby divide the test dataset into ten subsets according to the similarity score and report the results in Table 4. We find that the gains of our models over the TF baseline are mainly from those sentences whose TMs are with relatively high similarity. To our surprise, our models perform worse than TF on the subset with relatively low similarity except the subset with the lowest similarity.[5] This result demonstrates that our models with standard training are not robust to similarity scores, as deeply explained in the previous section.

Joint training Luckily the robustness issue can be fixed well by joint training, as depicted in the right part of Table 4. We can see that our model is better than the baseline TF on the subset of $[0, 0.3)$, and it substantially outperforms TF on the subset of $[0.3, 1)$. With the help of joint training, TF-SA delivers gains of 1.2 BLEU points over standard training, and gains of 5.7 BLEU points over the strong TF baseline on the entire test set.

Therefore, in the rest of the experiments, we employ joint training to set up all of our models because it is robust to the low similarity of TMs.

Without TM or with Ref as TM The situation without any TM and the situation with reference as a TM are more extreme cases of the robustness issue. As reported in Table 5, if a perfect TM is provided to our models, they can yield excellent translation results. Besides, the proposed methods are not inferior to the standard Transformer when no TM is provided. As a result, the proposed model makes it possible that a single unified model can handle both translation situations (with or without a TM), which is practical for online services.

[5]We further check these two exceptional sentences and find that they are very short in length. In particular, their word alignment results from the fast-align toolkit are very good, which may be beneficial to our proposed model. This might be the reason why our proposed model advances the baseline Transformer.

	BLEU	TF	TF-S	TF-SS	TF-SA
Es→En	Without TM	62.79	62.72	62.83	63.15
	Ref as TM	-	88.66	**93.19**	92.38
	With TM	-	67.99	68.40	**68.49**
Zh→En	Without TM	24.12	24.13	24.26	24.13
	Ref as TM	-	94.90	**99.43**	98.81
	With TM	-	24.22	**25.12**	25.03

Table 5: BLEU comparison on Es→En and Zh→En tasks. "Ref as TM", "With TM" and "Without TM" respectively denote our models are provided a reference as a TM, a retrieved TM, not provided a TM during inference.

BLEU	TF	TF-P	TF-SEQ	FM^+	TF-G	TF-SA
Dev	63.35	65.59	64.81	66.44	66.37	**68.68**
Test	62.79	65.22	65.16	65.90	66.21	**68.49**

Table 6: BLEU comparison with baselines on Es→En task.

Noisy TM To validate whether the model works well with noisy TMs, we also conduct a quick experiment by adding noises to TM for the test set by randomly replacing words in the target side of TM with incorrect words. After replacing one and two words, the proposed TF-SA achieves 68.17 BLEU points and 67.94 BLEU points, respectively. Both results are slightly worse than the noise-free TF-SA (68.49) but still better than the best TM baseline (66.21). Note that both results are obtained without retraining TF-SA model with noisy TM. This fact demonstrates our model is even robust to noisy TMs and thus it is useful for the online TM.

Comparison with baselines Table 6 illustrates the results between the proposed model TF-SA and the baselines. It is clearly shown that TF-SA surpasses all TM-based baselines with a substantial margin. In details, TF-SA outperforms TF-P and TF-SEQ by about 3.2 BLEU points, FM^+ by about 2.6 BLEU points, and the strong baseline TF-G by about 2.2 BLEU points.

Running time Since all TM-based models employ the same retrieval metric and their retrieval

Time(s)	TF	TF-P	TF-SEQ	TF-G	FM$^+$	TF-S	TF-SS	TF-SA
Train	3727	-	17841	7074	7720	4350	4361	4518
Test	0.30	0.71	1.91	0.55	0.33	0.39	0.40	0.41

Table 7: Running time comparison on Es→En task. Training time reports the time in seconds for training one epoch on average, and testing time reports the time in seconds for translating one sentence on average.

BLEU	TF	TF-P	TF-G	TF-S	TF-SS	TF-SA
Fr→En	66.25	69.69	70.87	72.00	**72.55**	72.35
En→Fr	66.49	69.08	69.59	70.38	71.03	**71.11**
Es→En	62.79	65.22	66.21	67.99	68.40	**68.49**
En→Es	60.11	61.94	62.76	66.52	66.61	**66.94**
De→En	58.50	61.49	61.72	**65.58**	64.86	65.56
En→De	53.15	57.01	56.88	**61.71**	60.87	61.35

Table 8: Translation accuracy in terms of BLEU on the TM-specialized tasks.

BLEU	WMT En→De			WMT Zh→En	
	news13	news14	news17	dev17	test17
TF	26.18	27.93	26.82	22.52	24.12
TF-P	26.26	27.79	26.70	22.65	24.17
TF-S	26.56	28.13	26.61	22.88	24.22
TF-SS	**27.02**	28.22	27.19	**23.85**	**25.12**
TF-SA	26.66	**28.66**	**27.48**	23.65	25.03

Table 9: Translation accuracy in terms of BLEU on the general WMT tasks.

time is exactly the same, we only report the running time of all TM-based NMT models excluding retrieval time in Table 7. As reported in this table, our proposed model further saves significant running time over TF-SEQ and TF-G for both training and testing, besides achieving better translation performance. In addition, although it requires slight overhead in training, its testing is more efficient than TF-P; and our training is faster than FM$^+$.

5.3 Overall Translation Quality

5.3.1 On the TM-specialized Datasets

The experimental results of all the systems on the six translation tasks of TM-specialized datasets are reported in Table 8. Several observations can be made from the results. First, the baseline TF-P and TF-G achieve substantial gains over the strong baseline TF, outperforming by [1.1, 4.1] BLEU points. This result is in line with the finding in Zhang et al. (2018) and Xia et al. (2019). Second, on the basis of that, compared with the strongest baseline TF-G, our proposed TF-S, TF-SS and TF-SA can obtain further gains up to 4.9 BLEU points, at least 1.2 BLEU points.

5.3.2 On the General WMT Datasets

It is important to mention that all previous TM-based approaches failed in getting notable improvements on the general WMT datasets. Since Xia et al. (2019) did not conduct experiments on the WMT datasets and their implementation is not released, we compare our models with two baselines: TF and TF-P. Our experimental results on the general WMT datasets are reported in Table 9. As we can see, the method TF-P is only comparable to the baseline NMT, which is in line with the observation in Zhang et al. (2018). In contrast, our models perform well on these tasks. Our best model gains about 0.7 BLEU points on the En→De and 1.0 BLEU point on the Zh→En task, over both baselines on average. The experimental results demonstrate that a TM based translation model can advance strong MT baselines on general translation tasks where a TM is not very similar to input source sentences. What's more, as shown in Table 5, our models can get excellent translation results while a perfect TM is provided.

In a summary, based on the above extensive experimental results, our proposed models substantially surpass several baselines on TM-specialized tasks and general tasks, in terms of BLEU and running time.

6 Related Work

In the statistical machine translation (SMT) diagram, Koehn and Senellart (2010a) extract bilingual segments from a TM which matches the source sentence to be translated, and employ a heuristic score to decide whether the extracted segments should be used as decoding constraints or not, then hardly constrain SMT to decode for those unmatched parts of the source sentence. Ma et al. (2011) design a fine-grained classifier, rather than the heuristic score, to predict the score for making more reliable decisions. Simard and Isabelle (2009), Wang et al. (2013) and Wang et al. (2014) add the extracted bilingual segments to the translation table of SMT, and then bias the decoder in a

soft constraint manner when decoding the source sentence with the augmented translation table. Liu et al. (2012) use the retrieved bilingual sentences to update the parameters for the log-linear model based SMT.

In recent years, many efforts are made on neural machine translation (NMT) associated with a TM. Li et al. (2016) and Farajian et al. (2017) make full use of the retrieved TM sentence pairs to fine-tune the pre-trained NMT model on-the-fly. The most obvious drawback of fine-tuning is that the delay is too long for testing sentences. To avoid the online tuning process, Zhang et al. (2018) and He et al. (2019) dynamically integrate translation pieces, based on n-grams extracted from the matched segments in the TM target, into the beam search stage. The second type of approach is efficient but heavily depends on the global hyper-parameter λ, which is sensitive to the development set, leading to inferior performance.

Recently, there are notable approaches for the sake of further excavation on TM-based NMT. Bulte and Tezcan (2019) and Xu et al. (2020) propose data augmentation approaches by augmenting input sentences with a TM which do not modify the NMT model architecture. Gu et al. (2018) and Xia et al. (2019) employ an auxiliary network to encode TMs and integrate it into the NMT architecture. Our model architecture is simpler than Gu et al. (2018) and Xia et al. (2019) and we encode a single TM target sentence and utilize simple attention mechanisms on the TM. And the architecture is more efficient and leads to a faster translation speed compared with Gu et al. (2018) and Xia et al. (2019). In particular, we propose a novel training criterion to make the TM-based NMT model more robust in different translation situations (with or without a TM). In parallel with our work, Cai et al. (2021) extend the translation memory from the bilingual setting to the monolingual setting through a cross-lingual retrieval technique, and Khandelwal et al. (2021) report significant improvements in quality on general translation tasks as ours, but their inference speed is two orders of magnitude slower than Transformer because they perform contextual word retrieval whose search space is much larger than that of sentence retrieval.

7 Conclusion

This paper presents a simple TM-based NMT model that employs a single bilingual sentence as its TM and thus is fast in training and inference. Although the presented model with the standard training outperforms strong TM-based baselines, it suffers from a robustness issue: its performance highly depends on the similarity of a TM. To address this issue, we propose a novel training criterion inspired by multiple-task learning and data augmentation. Experiments on TM-specialized tasks demonstrate its superiority over strong baselines in terms of running time and BLEU. Also, it is shown that a TM-based NMT model can advance the strong Transformer on general translation tasks like WMT.

Acknowledgments

This work is supported by NSFC (grant No. 61877051). We thank Jiatao Gu and Mengzhou Xia for providing their preprocessed datasets. We also thank the anonymous reviewers for providing valuable suggestions and feedbacks.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2016. Neural machine translation by jointly learning to align and translate. ArXiv preprint arXiv:1409.0473.

Shai Ben-David and Reba Schuller Borbely. 2008. A notion of task relatedness yielding provable multiple-task learning guarantees. *Mach. Learn.*, 73(3):273–287.

Andrzej Bialecki, Robert Muir, and Grant Ingersoll. 2020. Apache lucene 4. In *Proceedings of the SIGIR 2012 Workshop on Open Source Information Retrieval, OSIR@SIGIR 2012*, pages 17–24.

Bram Bulte and Arda Tezcan. 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 1800–1809, Florence, Italy.

Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. 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*.

Chi Chen, Maosong Sun, and Yang Liu. 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*.

Daxiang Dong, Hua Wu, Wei He, Dianhai Yu, and Haifeng Wang. 2015. Multi-task learning for multiple language translation. In *Proceedings of the*

53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 1723–1732.

Zi-Yi Dou and Graham Neubig. 2021. Word alignment by fine-tuning embeddings on parallel corpora. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 2112–2128.

Chris Dyer, Victor Chahuneau, and Noah A. Smith. 2013. A simple, fast, and effective reparameterization of IBM model 2. In *Proceedings of the Conference of the North American Chapter of the Association of Computational Linguistics (NAACL 2013)*, pages 644–648.

David A van Dyk and Xiao-Li Meng. 2001. The art of data augmentation. *Journal of Computational and Graphical Statistics*, 10(1):1–50.

Marzieh Fadaee, Arianna Bisazza, and Christof Monz. 2017. Data augmentation for low-resource neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 567–573.

Marzieh Fadaee and Christof Monz. 2018. Back-translation sampling by targeting difficult words in neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 436–446.

M. Amin Farajian, Marco Turchi, Matteo Negri, and Marcello Federico. 2017. Multi-domain neural machine translation through unsupervised adaptation. In *Proceedings of the Second Conference on Machine Translation*, pages 127–137.

Ignacio Garcia. 2009. Beyond translation memory: Computers and the professional translator. *The Journal of Specialised Translation*, 12(12):199–214.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O.K. Li. 2018. Search engine guided non-parametric neural machine translation. In *Proceedings of the 32nd AAAI Conference on Artificial Intelligence (AAAI 2018)*, pages 5133–5140.

Qiuxiang He, Guoping Huang, Lemao Liu, and Li Li. 2019. Word position aware translation memory for neural machine translation. In *Proceedings of the 8th CCF International Conference on Natural Language Processing and Chinese Computing*, pages 367–379.

Guoping Huang, Lemao Liu, Xing Wang, Longyue Wang, Huayang Li, Zhaopeng Tu, Chengyan Huang, and Shuming Shi. 2021. Transmart: A practical interactive machine translation system. ArXiv preprint arXiv:2105.13072.

Masoud Jalili Sabet, Philipp Dufter, François Yvon, and Hinrich Schütze. 2020. SimAlign: High quality word alignments without parallel training data using static and contextualized embeddings. In *Findings of the Association for Computational Linguistics: EMNLP 2020*, pages 1627–1643.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *Proceedings of the 2021 International Conference on Learning Representations*.

Diederik P. Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. ArXiv preprint arXiv:1412.6980.

Eliyahu Kiperwasser and Miguel Ballesteros. 2018. Scheduled multi-task learning: From syntax to translation. *Transactions of the Association for Computational Linguistics*, 6:225–240.

Philipp Koehn and Jeaf Senellart. 2010a. Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31.

Philipp Koehn and Jean Senellart. 2010b. Convergence of translation memory and statistical machine translation. In *Proceedings of AMTA Workshop on MT Research and the Translation Industry*, pages 21–31.

Guanlin Li, Lemao Liu, Guoping Huang, Conghui Zhu, and Tiejun Zhao. 2019. Understanding data augmentation in neural machine translation: Two perspectives towards generalization. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 5693–5699.

Xiaoqing Li, Jiajun Zhang, and Chengqing Zong. 2016. One sentence one model for neural machine translation. ArXiv preprint arXiv:1609.06490.

Lemao Liu, Hailong Cao, Taro Watanabe, Tiejun Zhao, Mo Yu, and Conghui Zhu. 2012. Locally training the log-linear model for SMT. In *Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*, pages 402–411.

Lemao Liu, Masao Utiyama, Andrew Finch, and Eiichiro Sumita. 2016. Neural machine translation with supervised attention. In *Proceedings of COLING 2016, the 26th International Conference on Computational Linguistics: Technical Papers*, pages 3093–3102.

Yang Liu, Kun Wang, Chengqing Zong, and Keh-Yih Su. 2019. A unified framework and models for integrating translation memory into phrase-based statistical machine translation. *Comput. Speech Lang.*, 54:176–206.

Yanjun Ma, Yifan He, Andy Way, and Josef van Genabith. 2011. Consistent translation using discriminative learning: A translation memory-inspired approach. In *Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics*, pages 1239–1248.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics ACL 2002*, pages 311–318.

Xipeng Qiu, Jiayi Zhao, and Xuanjing Huang. 2013. Joint chinese word segmentation and POS tagging on heterogeneous annotated corpora with multiple task learning. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing, EMNLP 2013*, pages 658–668.

Douglas Robinson. 2012. *Becoming a Translator: An Introduction to the Theory and Practice of Translation*. Routledge.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016a. Improving neural machine translation models with monolingual data. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 86–96.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016b. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (ACL 2016)*, pages 1715–1725.

Michel Simard and Pierre Isabelle. 2009. Phrase-based machine translation in a computer-assisted translation environment. In *Proceedings of the Twelfth Machine Translation Summit (MT Summit XII)*, pages 120–127.

Masao Utiyama, Graham Neubig, Takashi Onishi, and Eiichiro Sumita. 2011. Searching translation memories for paraphrases. In *Machine Translation Summit*, volume 13, pages 325–331.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, pages 5998–6008.

Kun Wang, Chengqing Zong, and Keh-Yih Su. 2013. Integrating translation memory into phrase-based machine translation during decoding. In *Proceedings of the 51st Annual Meeting of the Association for Computational Linguistics*, pages 11–21.

Kun Wang, Chengqing Zong, and Keh-Yih Su. 2014. Dynamically integrating cross-domain translation memory into phrase-based machine translation during decoding. In *Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers*, pages 398–408.

Xinyi Wang, Hieu Pham, Zihang Dai, and Graham Neubig. 2018. SwitchOut: an efficient data augmentation algorithm for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 856–861.

Yiren Wang, ChengXiang Zhai, and Hany Hassan. 2020. Multi-task learning for multilingual neural machine translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1022–1034.

Mengzhou Xia, Guoping Huang, Lemao Liu, and Shuming Shi. 2019. Graph based translation memory for neural machine translation. In *Proceedings of the 33rd AAAI Conference on Artificial Intelligence (AAAI 2019)*, pages 7297–7304.

Jitao Xu, Josep Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 1580–1590.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. 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)*, pages 1325–1335.

Zhun Zhong, Liang Zheng, Guoliang Kang, Shaozi Li, and Yi Yang. 2020. Random erasing data augmentation. In *The Thirty-Fourth AAAI Conference on Artificial Intelligence, AAAI*, pages 13001–13008.