Rethinking Translation Memory Augmented Neural Machine Translation

Hongkun Hao[1*]   **Guoping Huang**[2]   **Lemao Liu**[2†]
Zhirui Zhang[2]   **Shuming Shi**[2]   **Rui Wang**[1†]
[1]Shanghai Jiao Tong University   {haohongkun, wangrui12}@sjtu.edu.cn
[2]Tencent AI Lab   {donkeyhuang, redmondliu, shumingshi}@tencent.com   zrustc11@gmail.com

Abstract

This paper rethinks translation memory augmented neural machine translation (TM-augmented NMT) from two perspectives, i.e., a probabilistic view of retrieval and the variance-bias decomposition principle. The finding demonstrates that TM-augmented NMT is good at the ability of fitting data (i.e., lower bias) but is more sensitive to the fluctuations in the training data (i.e., higher variance), which provides an explanation to a recently reported contradictory phenomenon on the same translation task: TM-augmented NMT substantially advances vanilla NMT under the high-resource scenario whereas it fails under the low-resource scenario. Then we propose a simple yet effective TM-augmented NMT model to promote the variance and address the contradictory phenomenon. Extensive experiments show that the proposed TM-augmented NMT achieves consistent gains over both conventional NMT and existing TM-augmented NMT under two variance-preferable (low-resource and plug-and-play) scenarios as well as the high-resource scenario.

1 Introduction

The effectiveness of Translation Memory (TM) in Machine Translation has long been recognized (Garcia, 2009; Koehn and Senellart, 2010; Utiyama et al., 2011; Wang et al., 2013; Liu et al., 2019), because a TM retrieved from a bilingual dataset (i.e., training data or an external dataset) may provide valuable knowledge for the source sentence to be translated. Many notable approaches recently have been proposed to enhance neural machine translation (NMT) by using a TM (Feng et al., 2017; Gu et al., 2018; Cao et al., 2020; Hoang et al., 2022b; Cai et al., 2021; Huang et al., 2021).

For example, on the standard JRC-Acquis task, TM-augmented NMT achieves substantial

---

Model	High-Resource	Low-Resource
w/o TM	60.83	54.54
w/ TM	63.76 ↑	53.92 ↓

Table 1: Testing BLEU comparison on JRC-Acquis German⇒English task. w/o TM and w/ TM denote the vanilla Transformer and TM-augmented Transformer, respectively; High-Resource and Low-Resource denote full and quarter train data are used for NMT training and TM retrieval.

gains over the vanilla NMT (without TM) under the conventional high-resource training scenario. Unfortunately, Cai et al. (2021) surprisingly find that TM-augmented NMT fails to advance the vanilla NMT model on the same task under a low-resource scenario, if a quarter of the full data is used for training and TM retrieval, as reproduced in Table 1. Due to the lack of theoretical insights, it is unclear the reason why such a contradictory phenomenon happens. This motivates us to rethink the working mechanism of TM-augmented NMT as well as its statistical principle.

In this paper, we first cast TM-augmented NMT as an approximation of a latent variable model where the retrieved TM is the latent variable through a probabilistic view of retrieval. From this statistical viewpoint, we identify that the success of such an approximation depends on the variance of TM-augmented NMT with respect to the latent variable. Then, we empirically estimate the variance of TM-augmented NMT from the principle of variance-bias decomposition in the learning theory. Our findings demonstrate that TM-augmented NMT is worse than the vanilla NMT in terms of variance which indicates the sensitivity to fluctuations in the training set, although TM-augmented NMT is better in terms of the bias which indicates the ability of fitting data. The finding about the variance takes the responsibility of the contradictory phenomenon in Table 1, because

limited training data may amplify its negative effect on variance (Vapnik, 1999; Niyogi and Girosi, 1996; Bishop and Nasrabadi, 2006).

To better trade off the variance and the bias, we further propose a simple yet effective method for TM-augmented NMT. The proposed method is general to be applied on top of any TM-augmented NMT models. To validate the effectiveness of the proposed approach, we conduct extensive experiments on several translation tasks under different scenarios including low-resource scenario, plug-and-play scenario, and high-resource scenario.

Contributions of this paper are three-fold:
• It rethinks and analyzes the variance of TM-augmented NMT models from the probabilistic view of retrieval and the bias-variance decomposition perspective.
• It proposes a simple yet effective lightweight network to ensemble TM-augmented NMT models, which better trades off variance and bias.
• Its experiments show the effectiveness of the aforementioned approach, which outperforms both vanilla Transformer and baseline TM-augmented NMT models under the low-resource scenario, plug-and-play scenario, and conventional high-resource scenario.

2 Preliminary

2.1 NMT

Suppose $\mathbf{x} = \{x_1, \cdots, x_n\}$ is a source sentence and $\mathbf{y} = \{y_1, \cdots, y_m\}$ is the corresponding target sentence. NMT builds a probabilistic model with neural networks parameterized by $\boldsymbol{\theta}$, which is used to translate $\mathbf{x}$ in the source language to $\mathbf{y}$ in the target language. Formally, NMT aims to generate output $\mathbf{y}$ given $\mathbf{x}$ according to the conditional probability defined by Eq. (1):

$$P(\mathbf{y}|\mathbf{x}; \boldsymbol{\theta}) = \prod_{t=1}^{m} P(y_t|\mathbf{x}, \mathbf{y}_{<t}; \boldsymbol{\theta})$$
$$= \prod_{t=1}^{m} \text{Softmax}(f(H_t))[y_t], \quad (1)$$

where H_t denotes the NMT decoding state.

2.2 TM-Augmented NMT

In general, TM-augmented NMT works in the following two-step paradigm. It first retrieves top-K TM bilingual sentences $\mathbf{Z} = \{\mathbf{z_k}\}_{k=1}^{K}$, where $\mathbf{z}_k = (\mathbf{x}_k^{\text{tm}}, \mathbf{y}_k^{\text{tm}})$ is the k-th TM; Then it generates

the translation $\mathbf{y}$ by using the information from the source sentence $\mathbf{x}$ and its retrieved TMs $\mathbf{Z}$.

Retrieval Model Following previous works (Gu et al., 2018; Zhang et al., 2018; Xia et al., 2019; He et al., 2021), for $\mathbf{x}$ we employ Apache Lucene (Białecki et al., 2012) to retrieve top-100 similar bilingual sentences from datastore. Then we adopt the similarity function in Eq. (2) to re-rank the retrieved bilingual sentences and maintain top-K (e.g. $K = 5$) bilingual sentences as the TMs for $\mathbf{x}$:

$$\text{sim}(\mathbf{x}, \mathbf{z}_k) = 1 - \frac{\text{dist}(\mathbf{x}, \mathbf{x}_k^{\text{tm}})}{\max(|\mathbf{x}|, |\mathbf{x}_k^{\text{tm}}|)}, \quad (2)$$

where dist denotes the edit-distance.

Generation Model Given a source sentence $\mathbf{x}$ and a small set of relevant TMs $\mathbf{Z} = \{\mathbf{z_k}\}_{k=1}^{K}$, the generation model defines the conditional probability $P(\mathbf{y}|\mathbf{x}, \mathbf{Z}; \boldsymbol{\theta})$:

$$P(\mathbf{y}|\mathbf{x}, \mathbf{Z}; \boldsymbol{\theta}) = \prod_{t=1}^{m} P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z}; \boldsymbol{\theta})$$
$$= \prod_{t=1}^{m} \text{Softmax}(f(H_{t,\mathbf{z}}))[y_t], \quad (3)$$

where $H_{t,\mathbf{z}}$ denotes the decoding state of TM-augmented NMT. There are different TM-augmented NMT models, and accordingly there are different instantiations of $H_{t,\mathbf{z}}$. We refer readers to Gu et al. (2018); Bulte and Tezcan (2019); Cai et al. (2021); He et al. (2021) for its detailed definitions.

3 Rethinking TM-Augmented NMT

3.1 Probabilistic View of Retrieval

Given the source sentence $\mathbf{x}$, the top-K retrieval aforementioned actually can be considered as a probabilistic retrieval model (i.e., $P(\mathbf{Z}|\mathbf{x})$), from which a translation memory $\mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^{K}$ is sampled. Mathematically, such a retrieval model $P(\mathbf{Z}|\mathbf{x})$ is defined as follows:

$$P(\mathbf{Z}|\mathbf{x}) = \prod_{k} P(\mathbf{z}_k|\mathbf{x}) \propto \exp(\text{sim}(\mathbf{x}, \mathbf{z}_k)/T), \quad (4)$$

where sim is defined as in Eq. (2), and $T > 0$ is a temperature. Note that if T is a sufficiently small number, sampling $\mathbf{z}_k$ from the above probabilistic retrieval model is similar to the deterministic $\arg\max$ retrieval widely used in prior studies (Gu et al., 2018; Zhang et al., 2018; Xia et al., 2019).

By using the probabilistic retrieval model $P(\mathbf{Z}|\mathbf{x})$, the translation model $P(\mathbf{y}|\mathbf{x})$ is related to the variable $\mathbf{Z}$ theoretically through the following latent variable model:

$$P(\mathbf{y}|\mathbf{x}) = \sum_{\mathbf{Z}} P(\mathbf{Z}|\mathbf{x})P(\mathbf{y}|\mathbf{x},\mathbf{Z}) = \mathbb{E}_{\mathbf{Z}}P(\mathbf{y}|\mathbf{x},\mathbf{Z}).$$
(5)

In practice, it is impossible to perform the summation over all possible $\mathbf{Z}$. Instead it can be estimated by the Monte Carlo sampling:

$$P(\mathbf{y}|\mathbf{x}) \approx P(\mathbf{y}|\mathbf{x},\mathbf{Z}), \text{ with } \mathbf{Z} \sim P(\mathbf{Z}|\mathbf{x}). \quad (6)$$

As a result, according to Eq. (6), we can see the following statement: the TM-augmented NMT model $P(\mathbf{y}|\mathbf{x},\mathbf{Z})$ can be considered as an approximation of $P(\mathbf{y}|\mathbf{x})$ via Monte Carlo sampling over a latent variable model in Eq. (5). In particular, whether TM-augmented NMT $P(\mathbf{y}|\mathbf{x},\mathbf{Z})$ is a good estimator depends on the **expected approximate error** defined by $\mathbb{E}_{\mathbf{Z}}\big(P(\mathbf{y}|\mathbf{x},\mathbf{Z}) - P(\mathbf{y}|\mathbf{x})\big)^2$. In other words, $P(\mathbf{y}|\mathbf{x},\mathbf{Z})$ is a good estimator of $P(\mathbf{y}|\mathbf{x})$ if the expected approximate error is small; otherwise $P(\mathbf{y}|\mathbf{x},\mathbf{Z})$ is not a good estimator (Voinov and Nikulin, 2012).

Because of the Equation (5), the expected estimation error is actually derived by the variance of $P(\mathbf{y}|\mathbf{x},\mathbf{Z})$ with respect to $\mathbf{Z}$ as follows:

$$\mathbb{E}_{\mathbf{Z}}\big(P(\mathbf{y}|\mathbf{x},\mathbf{Z}) - P(\mathbf{y}|\mathbf{x})\big)^2 =$$
$$\mathbb{E}_{\mathbf{Z}}\big(P(\mathbf{y}|\mathbf{x},\mathbf{Z}) - \mathbb{E}_{\mathbf{Z}}P(\mathbf{y}|\mathbf{x},\mathbf{Z})\big)^2 := \mathbb{V}_{\mathbf{Z}}P(\mathbf{y}|\mathbf{x},\mathbf{Z}).$$
(7)

From the above equation, it is easy to observe that $\mathbb{V}_{\mathbf{Z}}P(\mathbf{y}|\mathbf{x},\mathbf{Z})$ actually controls the approximate effect of $P(\mathbf{y}|\mathbf{x},\mathbf{Z})$. Therefore, $\mathbb{V}_{\mathbf{Z}}P(\mathbf{y}|\mathbf{x},\mathbf{Z})$ would have a negative effect on TM-augmented NMT due to the fluctuations with respect to the variable $\mathbf{Z}$ if $\mathbb{V}_{\mathbf{Z}}P(\mathbf{y}|\mathbf{x},\mathbf{Z})$ is relatively large.

It is worth noting that the above analysis on the variance is only related to the variable $\mathbf{Z}$, but has nothing to do with the variables $\mathbf{x}$ and $\mathbf{y}$ and it is agnostic to neural network architecture. Moreover, it is intractable to estimate the approximate error, i.e., the variance $\mathbb{V}_{\mathbf{Z}}P(\mathbf{y}|\mathbf{x},\mathbf{Z})$ with respect to $\mathbf{Z}$ because $P(\mathbf{y}|\mathbf{x})$ requires the summation over all possible $\mathbf{Z}$. In the next subsection, we will employ the variance-bias decomposition principle to quantify the variance with respect to all variables given specific models.

3.2 Variance-Bias Decomposition

Estimation of Bias and Variance Bias-variance trade-off is a fundamental principle for understanding the generalization of predictive learning models and larger variance may induce the lower generalization ability of models (Geman et al., 1992; Hastie et al., 2009; Yang et al., 2020).

The bias-variance decomposition is typically defined in terms of the Mean Squared Error at the example level for classification tasks and Yang et al. (2020) reorganize its definition in terms of the Cross-Entropy loss. In the machine translation task the optimization loss is the Cross-Entropy loss at the token level, we hence simply extend the variance and bias decomposition in terms of cross-entropy (Yang et al., 2020) to the token level.[1] Specifically, assume $P_0(y|\mathbf{x},\mathbf{y}_{<t})$ is the empirical distribution (i.e., $P_0(y|\mathbf{x},\mathbf{y}_{<t})$ is one if $y = y_t$ is the ground-truth word y_t and 0 otherwise), and $P(y|\mathbf{x},\mathbf{y}_{<t})$ is the model output distribution. Then the expected cross entropy with respect to a random variable $\tau = \{\langle \mathbf{x}, \mathbf{y}\rangle\}$ (i.e., the training data) can be defined:

$$\mathbb{E}_{\tau}\big(H(P_0, P)\big) =$$
$$-\mathbb{E}_{\tau}\big(\sum_{t}\sum_{y} P_0(y|\mathbf{x},\mathbf{y}_{<t})\log P(y|\mathbf{x},\mathbf{y}_{<t})\big).$$
(8)

Further, it can be decomposed as:

$$\mathbb{E}_{\tau}\big(H(P_0, P)\big) = \underbrace{D_{\text{KL}}(P_0||\overline{P})}_{\textbf{Bias}^2}$$
$$+ \underbrace{\mathbb{E}_{\tau}\big(D_{\text{KL}}(P||\overline{P})\big)}_{\textbf{Variance}}, \quad (9)$$

where $\overline{P}$ is the expected probability after normalization:

$$\overline{P}(y|\mathbf{x},\mathbf{y}_{<t}) \propto \exp\Big(\mathbb{E}_{\tau}\big(\log P(y|\mathbf{x},\mathbf{y}_{<t})\big)\Big).$$
(10)

Generally speaking, the bias indicates the ability of the model P to fit the data whereas the variance measures the sensitivity of the model P to fluctuations in the training data.

We follow classic methods (Hastie et al., 2009; Yang et al., 2020) to estimate the variance in Eq. (9),

---

[1]Since it is not clear how to extend the variance-bias decomposition on top of the BLEU score used as an evaluation metric, we instead consider it on top of the token-level cross-entropy loss.

Model	Single Encoder		Dual Encoder	
	Var	Bias^2	Var	Bias^2
w/o TM	**0.2088**	1.9519	**0.1573**	1.9992
w/ TM	0.2263	**1.7500**	0.2168	**1.8460**

Table 2: Estimated variance and bias on JRC-Acquis German⇒English task under single encoder and dual encoder backbone respectively.

which is shown in Algorithm 1 of Appendix B. The key idea is to estimate the expectation over the random variable τ and it can be achieved by randomly splitting the given training dataset into several parts and training several models on each part for average. The above bias and variance estimation method is defined for $P(y|\mathbf{x}, \mathbf{y}_{<t})$ but it is similar to estimate the bias and variance for the TM-augmented NMT $P(y|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z})$ by retrieving a TM $\mathbf{Z}$ for each $\mathbf{x}$ via top-K retrieval as default rather than sampling as in §3.1.

Experiments We conduct experiments on JRC-Acquis German⇒English task to estimate variance and bias of vanilla Transformer and TM-augmented NMT models. Similar to the preliminary experiment in §1, we retrieve top-5 TMs for TM-augmented NMT. In order to eliminate the effect of model architecture, we use both TM-augmented NMT backbones, which respectively involve a single encoder (Bulte and Tezcan, 2019) and dual encoder (Cai et al., 2021), see details in Appendix C. Following Yang et al. (2020), Bias^2 is estimated by subtracting the variance from the loss.

Table 2 shows the variance-bias decomposition results of different models. Firstly, within each backbone, we can find that the variance of TM-augmented NMT model is larger than that of vanilla Transformer, which verifies our hypothesis that variance of TM-augmented NMT model is worse than that of vanilla Transformer, which results in poor performance under the low-resource scenario indicated in §1. Secondly, the bias of TM-augmented NMT model within each backbone is smaller than that of vanilla Transformer, which explains the better performance under the high-resource scenario indicated in §1. In addition, since the variance is highly dependent on the scale of training data and the limited training data may even amplify its negative effect on variance (Niyogi and Girosi, 1996; Vapnik, 1999; Bishop and Nasrabadi,

2006), the higher variance of TM-augmented NMT in Table 2 gives an explanation for the contradictory phenomenon in Table 1.

In summary, although the TM-augmented NMT model has lower bias, resulting in fitting on the training data better especially when the training data size is large, it has non-negligible higher variance, which leads to its poor performance with less training data. Therefore, the inherent flaw drives us to find ways to reduce the variance of TM-augmented NMT models, as shown in the next section. Note that our proposed methods shown below can all theoretically reduce the variance with respect to the variable $\mathbf{Z}$ (i.e. TM) in §3.1, and our experiments below empirically show that reducing the variance with respect to $\mathbf{Z}$ can actually reduce the model variance introduced here.

4 Proposed Approach for Lower Variance

In this section, we first propose two techniques to reduce the variance of the TM-augmented NMT model. Then we propose a new TM-augmented NMT on the basis of the two techniques to address the contradictory phenomenon as presented in §1. Finally we empirically quantify the variance and bias for the proposed TM-augmented NMT. Note that the proposed TM-augmented NMT is general enough to be applied on top of any specific TM-augmented NMT models.

4.1 Two Techniques to Reduce Variance

Technique 1: Conditioning on One TM Sentence In conventional TM-augmented NMT, the translation model $P(\mathbf{y}|\mathbf{x}, \mathbf{Z})$ is conditioned on $\mathbf{Z}$ consisting of five[2] retrieved TMs $\{\mathbf{z}_1, \mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5\}$ with $\mathbf{z}_i = (\mathbf{x}_i^{\text{tm}}, \mathbf{y}_i^{\text{tm}})$. Similar to the analysis in Eq.(5-6), we can obtain the following equations:

$$
\begin{aligned}
&P(\mathbf{y}|\mathbf{x}, \mathbf{z}_1) \\
&= \sum_{\mathbf{z}_2, \cdots, \mathbf{z}_5} P(\mathbf{y}|\mathbf{x}, \mathbf{z}_1, \mathbf{Z}_{>1}) P(\mathbf{Z}_{>1}|\mathbf{x}, \mathbf{z}_1) \\
&= \sum_{\mathbf{z}_2, \cdots, \mathbf{z}_5} P(\mathbf{y}|\mathbf{x}, \mathbf{z}_1, \mathbf{Z}_{>1}) P(\mathbf{Z}_{>1}|\mathbf{x}) \\
&\approx P(\mathbf{y}|\mathbf{x}, \mathbf{Z}), \text{ with } \mathbf{Z}_{>1} \sim P(\mathbf{Z}_{>1}|\mathbf{x}),
\end{aligned}
\tag{11}
$$

where $\mathbf{Z}_{>1} = \{\mathbf{z}_2, \mathbf{z}_3, \mathbf{z}_4, \mathbf{z}_5\}$, and the second equation holds due to the conditional independence assumption in Eq. (6). In this sense, we can see that $P(\mathbf{y}|\mathbf{x}, \mathbf{Z})$ conditioned on five retrieved

---

[2]Since many studies (Gu et al., 2018; Xia et al., 2019; Cai et al., 2021) find that using five TMs is suitable.

TMs is actually an approximation of $P(\mathbf{y}|\mathbf{x}, \mathbf{z}_1)$ conditioned on a single retrieved TM $\mathbf{z}_1$. As a result, as analysed in §3.1, whether $P(\mathbf{y}|\mathbf{x}, \mathbf{Z})$ is a good estimator depends on the variance with respect to $\mathbf{Z}_{>1}$, i.e., $\mathbb{V}_{\mathbf{Z}_{>1}} P(\mathbf{y}|\mathbf{x}, \mathbf{z}_1, \mathbf{Z}_{>1})$.

Based on the above analyses, to alleviate the variance, we directly estimate $P(\mathbf{y}|\mathbf{x})$ by using a single sampled sentence pair. Formally, suppose $\mathbf{z} = \langle \mathbf{x}^{\text{tm}}, \mathbf{y}^{\text{tm}} \rangle$ is a **single** bilingual sentence sampled from $P(\mathbf{z}|\mathbf{x})$, we adopt the following TM-augmented NMT conditioned on a single retrieved sentence $\mathbf{z}$:

$$
\begin{aligned}
P(y_t|\mathbf{x}, \mathbf{y}_{<t}) &= \mathbb{E}_z P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}) \\
&\approx P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}), \\
P(\mathbf{y}|\mathbf{x}) &= \prod_t P(y_t|\mathbf{x}, \mathbf{y}_{<t}) \\
&\approx \prod_t P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}),
\end{aligned} \tag{12}
$$

where $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z})$ can be any model architecture of TM-augmented NMT models $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z})$ by replacing top-K TMs $\mathbf{Z}$ with a single top-1 TM $\mathbf{z}$. In addition, the training of the above model $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z})$ is the same as the training of the conventional model $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z})$.

Technique 2: Enlarging the Sample Size In Eq. (12), $\mathbb{E}_z P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z})$ is approximated by one sample $\mathbf{z}$, which still induces some potential estimation errors due to the variance $\mathbb{V}_{\mathbf{z}} P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z})$. In fact, the estimation errors can be further reduced by the estimation using multiple samples as follows.

Proposition 1. *If $\mathbf{z}_1, \cdots, \mathbf{z}_K$ are independent and identically distributed random variables sampled from the $P(\mathbf{z}|\mathbf{x})$, then the following inequality holds (The proof is presented in Appendix A.):*

$$
\begin{aligned}
\mathbb{V}_{\mathbf{z}} P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}) &\geq \\
\mathbb{V}_{\mathbf{z}_1, \cdots, \mathbf{z}_K} &\left(\frac{1}{K} \sum_k P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k) \right).
\end{aligned} \tag{13}
$$

According to the above Proposition, we propose the method to approximate $P(y_t|\mathbf{x}, \mathbf{y}_{<t})$ through the average ensemble of all $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$:

$$
\begin{aligned}
P(y_t|\mathbf{x}, \mathbf{y}_{<t}) &\approx \frac{1}{K} \sum_{k=1}^{K} P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k), \\
P(\mathbf{y}|\mathbf{x}) &\approx \prod_{t=1}^{m} \sum_{k=1}^{K} P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k).
\end{aligned} \tag{14}
$$

Note that we do not particularly retrain the averaged ensemble model by directly taking the parameters from the model $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z})$ trained in Technique 1 above in this paper.

4.2 TM-Augmented NMT via Weighted Ensemble

In our experiments as shown in §4.3 later, we found that the both techniques deliver better variance but sacrifice its ability of fitting data (i.e., bias) compared to the standard TM-augmented NMT conditioning on all retrieved $\mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^{K}$. In order to further improve the bias for a better ability to fit data, we propose the weighted ensemble to establish a stronger relationship between the source sentence and each TM $\mathbf{z}_k$ by endowing a more powerful representation ability via the weighting coefficient $w(\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$ in Eq. (15). Formally, the weighted ensemble model is defined as:

$$
P(\mathbf{y}|\mathbf{x}) \approx \prod_{t=1}^{m} \sum_{k=1}^{K} w(\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k) P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k),
$$
$$(15)$$

where $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$ can be any TM-augmented model architecture as claimed in §4.1, and $w(\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$ is a weighting network via the following equation:

$$
w(\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k) = \text{Softmax}\big(f(H_t, H_{t,k})\big)[k], \tag{16}
$$

where f consists of two linear layers with residual connection and layer normalization (Vaswani et al., 2017), H_t is the decoding state of a vanilla translation model $P(y_t|\mathbf{x}, \mathbf{y}_{<t})$, $H_{t,k}$ is the decoding state for any TM-augmented model $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$ similar to $H_{t,\mathbf{z}}$ in Eq. (3). For example, if we implement the weighted ensemble model on top of the model architecture of Cai et al. (2021), then the source sentence and the k-th TM are encoded by two separate encoders respectively, resulting in a hidden state H_t for current time step and a contextualized TM representation $H_{t,k}$ for the k-th TM in current time step.

Similar to the average ensemble method, we do not train the whole network from scratch. Instead, we start from the trained model parameters of $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$ in §4.1 and then we just fine-tune the whole parameters including those from both $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$ and $w(\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$ for only about 2,000 updates on 90% part of valid data while the other 10% part of valid data are used to select the checkpoint for testing.

Model	w/o TM	TM-base	TM-single	TM-average	TM-weight
Var	0.1573	0.2168	0.1944	0.1918	**0.1814**
Bias^2	1.9992	**1.8460**	1.9369	1.9395	1.9137

Table 3: Estimated variance and bias.

Remark The above average ensemble model involves K retrieved sentence pairs similar to the standard translation memory augmented models (Gu et al., 2018; Zhang et al., 2018; Cai et al., 2021), but the notable difference is that our model $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$ conditions on a single $\mathbf{z}_k$ whereas those standard TM-augmented NMT models (see Eq. (3)) directly condition on K retrieved sentence pairs $\mathbf{Z} = \{\mathbf{z}_k\}_{k=1}^K$.

4.3 Empirical Analysis on Bias and Variance

To verify the effectiveness of the proposed methods empirically, we conduct the variance-bias decomposition experiments similar to §3.2. Specifically, we estimate the variance and bias of both three proposed models and two baselines (please refer to §5.1 for detailed settings).

Table 3 shows the overall results of estimated variance and bias. By comparing results of different models, we can get the following two observations: Firstly, through the results of variance, we can find that the three proposed methods all achieve lower variance compared with the default TM-augmented NMT model. It is notable that the weighted ensemble method achieves the best variance within all the TM-augmented NMT models. Secondly, through the results of bias, we can find that all TM-augmented NMT models achieve lower bias compared to vanilla Transformer (Vaswani et al., 2017) without TM. Although our methods have a slightly higher bias, the proposed weighted ensemble method can achieve comparable bias with regard to the default TM-augmented NMT model.

5 Experiments

In this section, we validate the effectiveness of the proposed methods in three scenarios: (1) the low-resource scenario where training pairs are scarce, (2) the plug-and-play scenario where additional bilingual pairs are added as the data store and the model is not re-trained any more as the data store is enlarged, and (3) the conventional high-resource

scenario where the entire training data are used for training and retrieval. We use BLEU score (Papineni et al., 2002) as the automatic metric for the translation quality evaluation.

5.1 Settings

Data We use the JRC-Acquis corpus (Steinberger et al., 2006) and select four translation directions including Spanish⇒English (Es⇒En), En⇒Es, German⇒English (De⇒En), and En⇒De, for evaluation. Besides, we also use the re-split version of the Multi-Domain data set in Aharoni and Goldberg (2020) originally collected by Koehn and Knowles (2017) for our experiments, which includes five domains: Medical, Law, IT, Koran and Subtitle. Detailed data statistics and descriptions are shown in Appendix E.

Models To study the effect of the proposed methods in §4, we implement a series of model variants by using the fairseq toolkit (Ott et al., 2019). #1 vanilla NMT without TM (Vaswani et al., 2017). We remove the model components related to TM, and only employ the encoder-decoder architecture for NMT. #2 Default TM-augmented NMT with top-5 TMs. We use top-5 TMs to train and test the model. Note that this is also a baseline model in Cai et al. (2021). #3 TM-augmented NMT with single TM. To study the effect of technique 1 in §4.1 which conditions only one TM, we use top-1 TM to test the model. In order to avoid overfitting, during each epoch we use training pairs with top-5 TMs and empty TM to train the model six times, which is similar to He et al. (2021). #4 TM-augmented NMT with the average ensemble. To study the effect of technique 2 in §4.1 which enlarges the sample size, we use the trained model in #3 directly and average ensemble top-5 TMs. #5 TM-augmented NMT with the weighted ensemble. We fine-tune the trained model in #3 with weighted ensemble top-5 TMs. Detailed model descriptions and experimental settings are in Appendix C and D respectively.

5.2 Low-Resource Scenario

One of the major advantages of our proposed weighted ensemble is that it has a lower variance, which means that it is less sensitive to fluctuations in the training data. This motivates us to conduct experiments in low-resource scenarios, where we use only a part of the training data to train models. Specifically, we create low-resource scenario by

Model	Medical	Law	IT	Koran	Subtitle	Average
#1 w/o TM (Vaswani et al., 2017)	47.62	50.85	34.40	14.45	20.22	33.51
#2 TM-base (Cai et al., 2021)	43.53	49.36	32.76	14.43	20.03	32.02
#3 TM-single (ours §4.1 Technique 1)	47.04	50.84	35.33	14.80	21.22	33.85
#4 TM-average (ours §4.1 Technique 2)	47.13	50.82	35.50	14.87	21.30	33.92
#5 TM-weight (ours §4.2)	**47.97**	**52.28**	**35.84**	**16.59**	**22.58**	**35.05**

Table 4: Experimental results (test set BLEU scores) on Multi-Domain dataset under low-resource scenario.

Model	Es⇒En	En⇒Es	De⇒En	En⇒De
w/o TM	58.44	56.11	54.54	49.97
TM-base	57.31	55.06	53.92	48.67
TM-single	57.56	55.24	54.03	48.82
TM-average	57.08	54.91	53.77	48.41
TM-weight	**59.14**	**56.53**	**55.36**	**50.51**

Table 5: Experimental results (test set BLEU scores) on four translation tasks of JRC-Acquis corpus under low-resource scenario.

Model	Es⇒En	En⇒Es	De⇒En	En⇒De
w/o TM	63.26	61.63	60.83	54.95
TM-base	66.42	62.81	63.76	57.79
TM-single	65.69	62.59	63.34	57.40
TM-average	65.29	62.72	63.01	57.42
TM-weight	**66.89**	**63.61**	**64.29**	**58.67**

Table 6: Test results on four translation tasks of JRC-Acquis corpus under high-resource scenario.

randomly partitioning each training set in JRC-Acquis corpus and Multi-Domain dataset into four subsets of equal size. Then we only use the training pairs in the first subset to train each model.

Results The test results of the above models on Multi-Domain dataset and JRC-Acquis corpus are presented in Table 4 and Table 5 respectively. We can get the following observations: (1) Our proposed weighted ensemble method delivers the best performance on test sets across all translation tasks, outperforming the vanilla Transformer by a large margin. (2) The performance of default TM-augmented NMT with top-5 TMs is degraded compared with vanilla Transformer, while single TM method and average ensemble method make up for the degradation issue to some extent.

5.3 Plug-and-Play Scenario

"Plug-and-play" is one of the most remarkable properties of TM (Cai et al., 2021), that is to say, the corpus used for TM retrieval is different during training and testing. This is useful especially in online products because we can adapt a trained model to a new corpus without retraining by adding or using a new TM. Specifically, we directly use models trained in §5.2, and add the second, the third, and the last subset to the TM data store gradually. At each datastore size, we retrieve for the test set again and test the performance of models with the newly retrieved TMs.

Results Figure 1 and Figure 2 show the main results on the test sets of JRC-Acquis corpus and

Multi-Domain dataset respectively. The general patterns are consistent across all experiments: The larger the TM becomes, the better translation performance the model achieves. When using all training data (4/4), the translation quality is boosted significantly. At the same time, our proposed weighted ensemble method achieves the best performance all the time.

5.4 High-Resource Scenario

Following prior works (He et al., 2021; Cai et al., 2021) in TM-augmented NMT, we also conduct experiments in the high-resource scenario where all the training pairs of JRC-Acquis corpus are used to train models in order to prove the practicability and universality of our proposed methods.

Results The results are presented in Table 6. We can see that our proposed weighted ensemble method still has the best performance even under the high-resource scenario, where default TM-augmented NMT using top-5 TMs is also relatively good because in this condition the bias plays an important role in fitting the whole training data to achieve good performance.

6 Related Work

This work mainly contributes to the research line of Translation Memory (TM) augmented Neural Machine Translation (NMT). Early works mainly concentrated on model architecture designing (Feng et al., 2017; Gu et al., 2018; Cao and Xiong, 2018; Cao et al., 2020; Zhang et al., 2018; Xia et al., 2019; Bulte and Tezcan, 2019; Xu
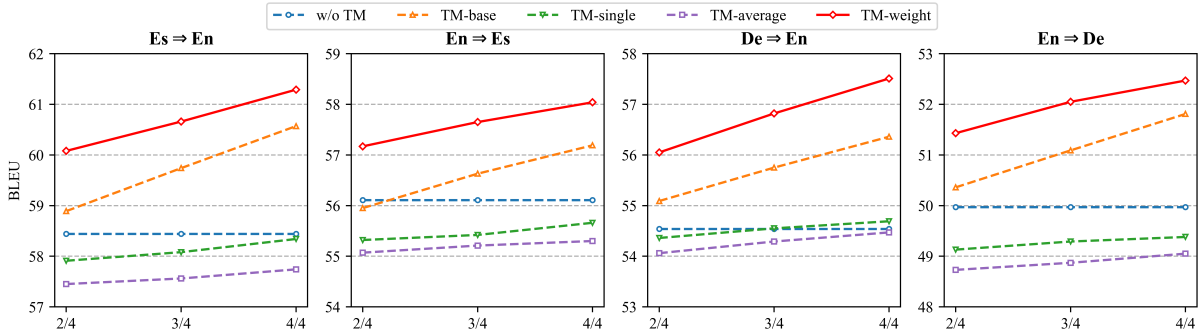
Figure 1: Test results of JRC-Acquis corpus under plug and play scenario with 2/4, 3/4 and 4/4 TMs respectively.
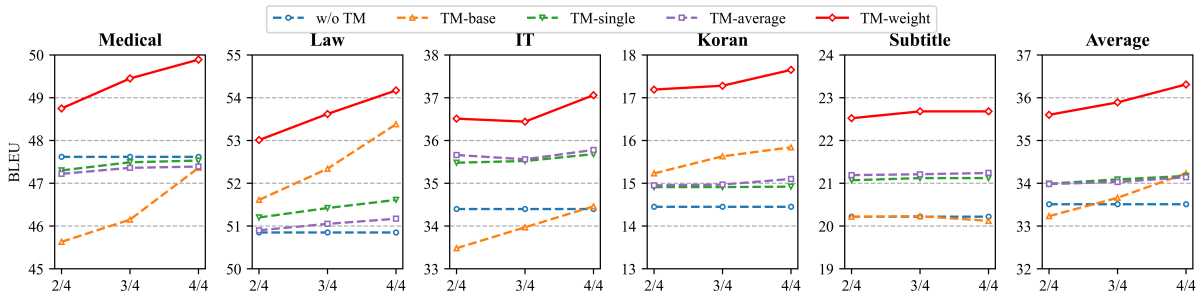

Figure 2: Test results of Multi-Domain dataset under plug and play scenario with 2/4, 3/4 and 4/4 TMs respectively.

et al., 2020; He et al., 2021; Wang et al., 2022d; Hoang et al., 2022a; Cai et al., 2022). Recently, Cai et al. (2021) used a dense retrieval method with a dual encoder to compute the similarity between the source sentence and TMs, which can use monolingual translation memory instead of bilingual ones. Hoang et al. (2022b) proposed to shuffle the retrieved suggestions to improve the robustness of TM-augmented NMT models. Cheng et al. (2022) proposed to contrastively retrieve translation memories that are holistically similar to the source sentence while individually contrastive to each other providing maximal information gains.

The distinctions between our work and prior works are obvious: (1) We rethink TM-augmented NMT from a probabilistic view of retrieval and the variance-bias decomposition principle; (2) We consider the performance of TM-augmented NMT under low-resource scenario, plug-and-play scenario, and conventional high-resource scenario, instead of only high-resource scenario. (3) Our methods are agnostic to specific model and retrieval method and thus can be applied on top of any advanced architecture.

Another research line highly related to our work is kNN-MT (Khandelwal et al., 2021; Zheng et al., 2021a,b; Meng et al., 2022; Wang et al., 2021, 2022a; Zhu et al., 2022; Wang et al., 2022c; Du

et al., 2022, 2023; Wang et al., 2022b). However, kNN-MT retrieves similar key-value pairs on the token level, resulting in slow retrieval speed (Meng et al., 2022; Dai et al., 2023) and large storage space (Wang et al., 2022a; Zhu et al., 2022), whereas TM-augmented NMT retrieves similar pairs on the sentence level, which has more efficient retrieval speed and takes up less storage space.

7 Conclusion

Existing work surprisingly finds that TM-augmented NMT fails to advance the NMT model under low-resource scenario, but it is unclear the reason why such a contradictory phenomenon happens. This paper rethinks TM-augmented NMT from latent variable probabilistic model view and variance-bias decomposition view respectively, giving the reason for the failure under low-resource scenario. Estimation for variance and bias indicates that TM-augmented NMT is better at fitting data (bias) yet worse at sensitivity to fluctuations in the training data (variance). To better trade off the bias and variance, this paper proposes a simple yet effective weighted ensemble method for TM-augmented NMT. Experiments under three scenarios demonstrate that the proposed method outperforms both vanilla Transformer and baseline

2596

TM-augmented models consistently. Future work could aim to find factors that influence the bias of TM-augmented NMT models, such as the quality of retrieved TMs.

Limitations

Compared with the standard NMT, TM-augmented NMT models incur extra retrieval time during both training and inference. Besides, since both TMs and the source sentence need to be encoded, the speed of TM-augmented NMT is slower than that of standard NMT. These characteristics indeed slightly induce some overhead. Specifically, experiments show that the latency time cost for all TM-augmented NMT models is higher than for vanilla Transformers. And the decoding time of our proposed weighted ensemble method is twice as much as that of the standard NMT. This issue is one limitation of TM-augmented NMT, which can be further studied and addressed.

Ethics Statement

This paper will not pose any ethical problems. First, machine translation is a standard task in natural language processing. Second, the datasets used in this paper have been already used in previous papers.

Acknowledgements

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7747–7763, Online. Association for Computational Linguistics.

Andrzej Białecki, Robert Muir, Grant Ingersoll, and Lucid Imagination. 2012. Apache lucene 4. In *SIGIR 2012 workshop on open source information retrieval*, page 17.

Christopher M Bishop and Nasser M Nasrabadi. 2006. *Pattern recognition and machine learning*, volume 4. Springer.

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. Association for Computational Linguistics.

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 (Volume 1: Long Papers)*, pages 7307–7318, Online. Association for Computational Linguistics.

Deng Cai, Yan Wang, Lemao Liu, and Shuming Shi. 2022. Recent advances in retrieval-augmented text generation. In *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '22, page 3417–3419, New York, NY, USA. Association for Computing Machinery.

Qian Cao, Shaohui Kuang, and Deyi Xiong. 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*, pages 1982–1989. IOS Press.

Qian Cao and Deyi Xiong. 2018. Encoding gated translation memory into neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3042–3047, Brussels, Belgium. Association for Computational Linguistics.

Xin Cheng, Shen Gao, Lemao Liu, Dongyan Zhao, and Rui Yan. 2022. Neural machine translation with contrastive translation memories. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 3591–3601, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yuhan Dai, Zhirui Zhang, Qiuzhi Liu, Qu Cui, Weihua Li, Yichao Du, and Tong Xu. 2023. Simple and scalable nearest neighbor machine translation. *arXiv preprint arXiv:2302.12188*.

Yichao Du, Weizhi Wang, Zhang Zhirui, Boxing Chen, Tong Xu, Jun Xie, and Enhong Chen. 2022. Non-parametric domain adaptation for end-to-end speech translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 306–320, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Yichao Du, Zhirui Zhang, Bingzhe Wu, Lemao Liu, Tong Xu, and Enhong Chen. 2023. Federated nearest neighbor machine translation. *arXiv preprint arXiv:2302.12211*.

Yang Feng, Shiyue Zhang, Andi Zhang, Dong Wang, and Andrew Abel. 2017. Memory-augmented neural machine translation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1390–1399, Copenhagen, Denmark. Association for Computational Linguistics.

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

Stuart Geman, Elie Bienenstock, and René Doursat. 1992. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 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*, pages 5133–5140. AAAI Press.

Trevor Hastie, Robert Tibshirani, Jerome H Friedman, and Jerome H Friedman. 2009. *The elements of statistical learning: data mining, inference, and prediction*, volume 2. Springer.

Qiuxiang He, Guoping Huang, Qu Cui, Li Li, and Lemao Liu. 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 (Volume 1: Long Papers)*, pages 3170–3180, Online. Association for Computational Linguistics.

Cuong Hoang, Devendra Sachan, Prashant Mathur, Brian Thompson, and Marcello Federico. 2022a. Improving retrieval augmented neural machine translation by controlling source and fuzzy-match interactions. *arXiv preprint arXiv:2210.05047*.

Cuong Hoang, Devendra Sachan, Prashant Mathur, Brian Thompson, and Marcello Federico. 2022b. Improving robustness of retrieval augmented translation via shuffling of suggestions. *arXiv preprint arXiv:2210.05059*.

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*.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021*. OpenReview.net.

Diederik P. Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Philipp Koehn, Hieu Hoang, Alexandra Birch, Chris Callison-Burch, Marcello Federico, Nicola Bertoldi, Brooke Cowan, Wade Shen, Christine Moran, Richard Zens, Chris Dyer, Ondřej Bojar, Alexandra Constantin, and Evan Herbst. 2007. Moses: Open source toolkit for statistical machine translation. In *Proceedings of the 45th Annual Meeting of the Association for Computational Linguistics Companion Volume Proceedings of the Demo and Poster Sessions*, pages 177–180, Prague, Czech Republic. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39, Vancouver. Association for Computational Linguistics.

Philipp Koehn and Jean Senellart. 2010. Convergence of translation memory and statistical machine translation. In *Proceedings of the Second Joint EM+/CNGL Workshop: Bringing MT to the User: Research on Integrating MT in the Translation Industry*, pages 21–32, Denver, Colorado, USA. Association for Machine Translation in the Americas.

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. *Computer Speech & Language*, 54:176–206.

Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang, and Jiwei Li. 2022. Fast nearest neighbor machine translation. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 555–565, Dublin, Ireland. Association for Computational Linguistics.

Partha Niyogi and Federico Girosi. 1996. On the relationship between generalization error, hypothesis complexity, and sample complexity for radial basis functions. *Neural Computation*, 8(4):819–842.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier,

and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 48–53, Minneapolis, Minnesota. Association for Computational Linguistics.

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*, pages 311–318, Philadelphia, Pennsylvania, USA. Association for Computational Linguistics.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufiş, and Dániel Varga. 2006. The JRC-Acquis: A multilingual aligned parallel corpus with 20+ languages. In *Proceedings of the Fifth International Conference on Language Resources and Evaluation (LREC'06)*, Genoa, Italy. European Language Resources Association (ELRA).

Masao Utiyama, Graham Neubig, Takashi Onishi, and Eiichiro Sumita. 2011. Searching translation memories for paraphrases. In *Proceedings of Machine Translation Summit XIII: Papers*, Xiamen, China.

Vladimir N Vapnik. 1999. An overview of statistical learning theory. *IEEE transactions on neural networks*, 10(5):988–999.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 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*, pages 5998–6008.

Vasiliĭ Grigorevich Voinov and Mikhail Stepanovich Nikulin. 2012. *Unbiased estimators and their applications: volume 1: univariate case*, volume 263. Springer Science & Business Media.

Dexin Wang, Kai Fan, Boxing Chen, and Deyi Xiong. 2022a. Efficient cluster-based k-nearest-neighbor machine translation. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2175–2187, Dublin, Ireland. Association for Computational Linguistics.

Dongqi Wang, Haoran Wei, Zhirui Zhang, Shujian Huang, Jun Xie, and Jiajun Chen. 2022b. Non-parametric online learning from human feedback for neural machine translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 36(10):11431–11439.

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 (Volume 1: Long Papers)*, pages 11–21, Sofia, Bulgaria. Association for Computational Linguistics.

Qiang Wang, Rongxiang Weng, and Ming Chen. 2022c. Learning decoupled retrieval representation for nearest neighbour neural machine translation. In *Proceedings of the 29th International Conference on Computational Linguistics*, pages 5142–5147, Gyeongju, Republic of Korea. International Committee on Computational Linguistics.

Shuhe Wang, Jiwei Li, Yuxian Meng, Rongbin Ouyang, Guoyin Wang, Xiaoya Li, Tianwei Zhang, and Shi Zong. 2021. Faster nearest neighbor machine translation. *arXiv preprint arXiv:2112.08152*.

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022d. Training data is more valuable than you think: A simple and effective method by retrieving from training data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3170–3179, Dublin, Ireland. Association for Computational Linguistics.

Mengzhou Xia, Guoping Huang, Lemao Liu, and Shuming Shi. 2019. Graph based translation memory for neural machine translation. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):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, Online. Association for Computational Linguistics.

Zitong Yang, Yaodong Yu, Chong You, Jacob Steinhardt, and Yi Ma. 2020. Rethinking bias-variance trade-off for generalization of neural networks. In *Proceedings of the 37th International Conference on Machine Learning, ICML 2020, 13-18 July 2020, Virtual Event*, volume 119 of *Proceedings of Machine Learning Research*, pages 10767–10777. PMLR.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 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, Volume 1 (Long Papers), pages 1325–1335, New Orleans, Louisiana. Association for Computational Linguistics.

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021a. 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 (Volume 2: Short Papers)*, pages 368–374, Online. Association for Computational Linguistics.

Xin Zheng, Zhirui Zhang, Shujian Huang, Boxing Chen, Jun Xie, Weihua Luo, and Jiajun Chen. 2021b. Non-parametric unsupervised domain adaptation for neural machine translation. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 4234–4241, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenhao Zhu, Shujian Huang, Yunzhe Lv, Xin Zheng, and Jiajun Chen. 2022. What knowledge is needed? towards explainable memory for knn-mt domain adaptation. *arXiv preprint arXiv:2211.04052*.

A Proof of Proposition 1 in §4

Throughout this section, we prove the Proposition 1 in §4. For simplicity, we use $f(\mathbf{z})$ and $f(\mathbf{z}_k)$ to denote $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z})$ and $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{z}_k)$ in Proposition 1, respectively. Note that each $\mathbf{z}_k$ are i.i.d sampled from $P(\mathbf{z}|\mathbf{x})$.

Proof. Firstly, for the relationship between $\mathbb{E}_{\mathbf{z}} f(\mathbf{z})$ and $\mathbb{E}_{\mathbf{z}_1, \cdots, \mathbf{z}_K} \left(\frac{1}{K} \sum_k f(\mathbf{z}_k) \right)$ we can get the following equations:

$$
\begin{aligned}
& \mathbb{E}_{\mathbf{z}_1, \cdots, \mathbf{z}_K} \left(\frac{1}{K} \sum_k f(\mathbf{z}_k) \right) \\
= {} & \frac{1}{K} \mathbb{E}_{\mathbf{z}_1, \cdots, \mathbf{z}_K} \left(\sum_k f(\mathbf{z}_k) \right) \\
= {} & \frac{1}{K} \sum_k \left(\mathbb{E}_{\mathbf{z}_k} f(\mathbf{z}_k) \right) \\
= {} & \frac{1}{K} \sum_k \left(\mathbb{E}_{\mathbf{z}} f(\mathbf{z}) \right) \\
= {} & \frac{1}{K} * K * \mathbb{E}_{\mathbf{z}} f(\mathbf{z}) \\
= {} & \mathbb{E}_{\mathbf{z}} f(\mathbf{z}).
\end{aligned}
$$

Secondly, we start from the relationship between $\mathbb{V}_{\mathbf{z}} f(\mathbf{z})$ and $\mathbb{V}_{\mathbf{z}_1, \mathbf{z}_2} \left(f(\mathbf{z}_1) + f(\mathbf{z}_2) \right)$:

$$
\begin{aligned}
& \mathbb{V}_{\mathbf{z}_1, \mathbf{z}_2} \left(f(\mathbf{z}_1) + f(\mathbf{z}_2) \right) \\
= {} & \mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2} \Big(f(\mathbf{z}_1) + f(\mathbf{z}_2) \\
& \quad - \mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2} \left(f(\mathbf{z}_1) + f(\mathbf{z}_2) \right) \Big)^2 \\
= {} & \mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2} \left(f(\mathbf{z}_1) + f(\mathbf{z}_2) \right)^2 \\
& \quad - \left(\mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2} \left(f(\mathbf{z}_1) + f(\mathbf{z}_2) \right) \right)^2 \\
= {} & \mathbb{E}_{\mathbf{z}_1} \left(f(\mathbf{z}_1) \right)^2 + \mathbb{E}_{\mathbf{z}_2} \left(f(\mathbf{z}_2) \right)^2 \\
& \quad + 2 \mathbb{E}_{\mathbf{z}_1, \mathbf{z}_2} \left(f(\mathbf{z}_1) f(\mathbf{z}_2) \right) \\
& \quad - \left(\mathbb{E}_{\mathbf{z}_1} f(\mathbf{z}_1) \right)^2 - \left(\mathbb{E}_{\mathbf{z}_2} f(\mathbf{z}_2) \right)^2 \\
& \quad - 2 \left(\mathbb{E}_{\mathbf{z}_1} f(\mathbf{z}_1) \right) \left(\mathbb{E}_{\mathbf{z}_2} f(\mathbf{z}_2) \right) \\
= {} & \mathbb{E}_{\mathbf{z}_1} \left(f(\mathbf{z}_1) \right)^2 + \mathbb{E}_{\mathbf{z}_2} \left(f(\mathbf{z}_2) \right)^2 \\
& \quad + 2 \left(\mathbb{E}_{\mathbf{z}_1} f(\mathbf{z}_1) \right) \left(\mathbb{E}_{\mathbf{z}_2} f(\mathbf{z}_2) \right) \\
& \quad - \left(\mathbb{E}_{\mathbf{z}_1} f(\mathbf{z}_1) \right)^2 - \left(\mathbb{E}_{\mathbf{z}_2} f(\mathbf{z}_2) \right)^2 \\
& \quad - 2 \left(\mathbb{E}_{\mathbf{z}_1} f(\mathbf{z}_1) \right) \left(\mathbb{E}_{\mathbf{z}_2} f(\mathbf{z}_2) \right) \\
= {} & \mathbb{E}_{\mathbf{z}_1} \left(f(\mathbf{z}_1) \right)^2 - \left(\mathbb{E}_{\mathbf{z}_1} f(\mathbf{z}_1) \right)^2 \\
& \quad + \mathbb{E}_{\mathbf{z}_2} \left(f(\mathbf{z}_2) \right)^2 - \left(\mathbb{E}_{\mathbf{z}_2} f(\mathbf{z}_2) \right)^2 \\
= {} & \mathbb{V}_{\mathbf{z}_1} f(\mathbf{z}_1) + \mathbb{V}_{\mathbf{z}_2} f(\mathbf{z}_2) \\
= {} & 2 \mathbb{V}_{\mathbf{z}} f(\mathbf{z}).
\end{aligned}
$$

Then, we can get the relationship between $\mathbb{V}_{\mathbf{z}} f(\mathbf{z})$ and $\mathbb{V}_{\mathbf{z}_1, \mathbf{z}_2} \left(\frac{1}{2} \left(f(\mathbf{z}_1) + f(\mathbf{z}_2) \right) \right)$ as follows:

$$
\begin{aligned}
& \mathbb{V}_{\mathbf{z}_1, \mathbf{z}_2} \left(\frac{1}{2} \left(f(\mathbf{z}_1) + f(\mathbf{z}_2) \right) \right) \\
= {} & \frac{1}{2^2} \mathbb{V}_{\mathbf{z}_1, \mathbf{z}_2} \left(f(\mathbf{z}_1) + f(\mathbf{z}_2) \right) \\
= {} & \frac{1}{4} * 2 * \mathbb{V}_{\mathbf{z}} f(\mathbf{z}) \\
= {} & \frac{1}{2} \mathbb{V}_{\mathbf{z}} f(\mathbf{z}) \\
< {} & \mathbb{V}_{\mathbf{z}} f(\mathbf{z}).
\end{aligned}
$$

Similarly, for the relationship between $\mathbb{V}_{\mathbf{z}} f(\mathbf{z})$ and $\mathbb{V}_{\mathbf{z}_1, \cdots, \mathbf{z}_K} \left(\frac{1}{K} \sum_k f(\mathbf{z}_k) \right)$ we can get the

following equations:

$$\mathbb{V}_{\mathbf{z}_1,\cdots,\mathbf{z}_K}\left(\frac{1}{K}\sum_k f(\mathbf{z}_k)\right)$$

$$= \frac{1}{K^2}\mathbb{V}_{\mathbf{z}_1,\cdots,\mathbf{z}_K}\left(\sum_k f(\mathbf{z}_k)\right)$$

$$= \frac{1}{K^2}\left(\sum_k\left(\mathbb{E}_{\mathbf{z}_k}\left(f(\mathbf{z}_k)\right)^2 - \left(\mathbb{E}_{\mathbf{z}_k}f(\mathbf{z}_k)\right)^2\right)\right)$$

$$= \frac{1}{K^2}\sum_k\left(\mathbb{V}_{\mathbf{z}_k}f(\mathbf{z}_k)\right)$$

$$= \frac{1}{K^2}\sum_k\left(\mathbb{V}_{\mathbf{z}}f(\mathbf{z})\right)$$

$$= \frac{1}{K^2}*K*\mathbb{V}_{\mathbf{z}}f(\mathbf{z})$$

$$= \frac{1}{K}\mathbb{V}_{\mathbf{z}}f(\mathbf{z})$$

$$\leq \mathbb{V}_{\mathbf{z}}f(\mathbf{z}),$$

where the equality holds if and only if $K = 1$. $\square$

B   Variance-Bias Estimation Method

In §3.2, we mention that we follow classic methods (Hastie et al., 2009; Yang et al., 2020) to estimate the variance and bias in Eq. (9). Here we detail the method to estimate the variance, which is shown in Algorithm 1. Then Bias^2 is estimated by subtracting the variance from the loss (Yang et al., 2020). Specifically, in this paper we set N to be 1 and k to be 4 in Algorithm 1. Besides, we compute the variance for each test point $(\mathbf{x}, \mathbf{y}_{<t})$ in test set, and average them to get the final variance for each model.

C   Details about Architecture

In §3.2, we use single encoder backbone (Bulte and Tezcan, 2019) and dual encoder backbone (Cai et al., 2021) respectively. Here we provide a detailed description of these two architectures.

Single Encoder Architecture   For this architecture, we follow Bulte and Tezcan (2019). Specifically, the model architecture is the same as vanilla Transformer, which consists of an encoder and a decoder. The encoder encodes the concatenation of the source sentence and TM, relying on the encoder's self-attention to compare the source sentence to the TM and determine which TM phrases are relevant for the translation(Hoang et al., 2022a). Therefore, the only difference is that we need to change the input format to the concatenation of the source sentence and TM.

Algorithm 1 Estimating Variance of NMT Models

Input: Test point $(\mathbf{x}, \mathbf{y}_{<t})$, Training data $\boldsymbol{\tau}$.
1: **for** $i = 1$ to k **do**
2:      Split the $\boldsymbol{\tau}$ into $\boldsymbol{\tau}_1^{(i)}, \cdots, \boldsymbol{\tau}_N^{(i)}$.
3:      **for** $j = 1$ to N **do**
4:          Train the model using $\boldsymbol{\tau}_j^{(i)}$;
5:          Evaluate the model at $(\mathbf{x}, \mathbf{y}_{<t})$; call the result $P_j^{(i)}(y|\mathbf{x}, \mathbf{y}_{<t})$;
6:          Normalize the top 100 probability in $P_j^{(i)}(y|\mathbf{x}, \mathbf{y}_{<t})$ and setting others to be 0 (to reduce computation complexity);
7:      **end for**
8: **end for**
9: Compute $\hat{P}(y|\mathbf{x}, \mathbf{y}_{<t}) = \exp\left(\frac{1}{N\cdot k}\sum_{i,j}\log P_j^{(i)}(y|\mathbf{x}, \mathbf{y}_{<t})\right)$ ($\hat{P}(y|\mathbf{x}, \mathbf{y}_{<t})$ estimates $\overline{P}(y|\mathbf{x}, \mathbf{y}_{<t})$).
10: Normalize $\hat{P}(y|\mathbf{x}, \mathbf{y}_{<t})$ to get a probability distribution.
11: Compute the variance $var = \frac{1}{N\cdot k}\sum_{ij} D_{KL}\left(P_j^{(i)}(y|\mathbf{x}, \mathbf{y}_{<t})||\hat{P}(y|\mathbf{x}, \mathbf{y}_{<t})\right)$

Output: var

Dual Encoder Architecture   This architecture is similar to that used by Cai et al. (2021), which achieves relatively good performance and thus can serve as a strong baseline (Cai et al., 2021).

On the encoder side, the source sentence and TM are encoded by two separate encoders respectively, resulting in a set of contextualized token representations $\{z_{kj}\}_{j=1}^{L_k}$, where L_k is the length of the k-th TM.

On the decoder side, after using $\mathbf{y}_{<\mathbf{t}}$ and $\mathbf{x}$ to get the hidden state H_t at current time step t, it can produce the original NMT probability $P_{nmt}(y_t|\mathbf{x}, \mathbf{y}_{<t})$ and the TM probability $P_{tm}(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z})$ concurrently, and finally get the probability $P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z})$.

Specifically, the original NMT probability $P_{nmt}(y_t|\mathbf{x}, \mathbf{y}_{<t})$ is defined by as follows:

$$P_{nmt}(y_t|\mathbf{x}, \mathbf{y}_{<t}) = \text{Softmax}\left(f(H_t)\right)[y_t], \quad (17)$$

where f is a linear layer that maps hidden state H_t to a probability distribution. And for the TM probability $P_{tm}(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z})$, the decoder firstly computes a cross attention score $\alpha = \{\alpha_{kj}\}$ over all tokens of the k-th TM sentence where α_{kj} is the attention score of H_t to the j-th token in the k-th

Model	# Param.	GPU Hours	Hyperparam.	
			learning rate	dropout
Transformer	83M	16h	7e-4	0.1
TM-augmented Transformer	98M	19h	7e-4	0.1

Table 7: The number of parameters, training budget (in GPU hours), and hyperparameters of each model.

Usage	Package	License
Preprocessing	mosesdecoder (Koehn et al., 2007)[1]	LGPL-2.1
	subword-nmt (Sennrich et al., 2016)[2]	MIT
Model training	fairseq (Ott et al., 2019)[3]	MIT
Evaluation	BLEU (Papineni et al., 2002)[1]	LGPL-2.1

[1] https://github.com/moses-smt/mosesdecoder
[2] https://github.com/rsennrich/subword-nmt
[3] https://github.com/facebookresearch/fairseq

Table 8: Packages we used for preprocessing, model training and evaluation.

TM. The computation is shown as follows:

$$\alpha_{kj} = \frac{\exp(H_t^T W_{tm} z_{kj})}{\sum_{k=1}^{K} \sum_{j=1}^{L_k} \exp(H_t^T W_{tm} z_{kj})}, \quad (18)$$

where W_{tm} are parameters. Then we can get the contextualized TM representation $H_{t,\mathbf{Z}}$ for the TM $\mathbf{Z}$ in current time step, as shown in Eq. (19):

$$H_{t,\mathbf{Z}} = W_h \sum_{k=1}^{L} \sum_{j=1}^{L_k} \alpha_{kj} z_{kj}, \quad (19)$$

where W_h are parameters. Then, the TM probability $P_{tm}(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z})$ can be computed as follows:

$$P_{tm}(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z}) = \sum_{k=1}^{K} \sum_{j=1}^{L_k} \alpha_{kj} \mathbb{I}_{z_{kj}=y_t}, \quad (20)$$

where $\mathbb{I}$ is the indicator function.

Finally, the decoder computes the next-token probability as follows:

$$P(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z}) = (1 - \lambda_t)P_{nmt}(y_t|\mathbf{x}, \mathbf{y}_{<t}) + \lambda_t P_{tm}(y_t|\mathbf{x}, \mathbf{y}_{<t}, \mathbf{Z}) \quad (21)$$

where λ_t is a gating variable computed by another linear layer whose input is $H_{t,\mathbf{Z}}$.

D Detailed Settings for All Experiments

Here we provide a detailed description of our configuration settings in the paper for the

preliminary experiment (§1), the variance-bias decomposition experiment (§3.2 and §4) and the main experiment (§5) respectively.

Settings for Preliminary Experiment in §1 In this part, we implement vanilla Transformer base model (Vaswani et al., 2017) for the standard NMT. The learning rate schedule, dropout, and label smoothing are the same as Vaswani et al. (2017). For the TM-augmented NMT, we follow Cai et al. (2021) and implement the dual encoder architecture aforementioned. We conduct experiments on the JRC-Acquis German⇒English task. For high-resource scenario we use the full training data and train models with up to 100k steps. For low-resource scenario we randomly select a quarter of training data and train models with up to 30k steps.

Settings for Variance-Bias Decomposition Experiment in §3.2 and §4 For the experiment in §3.2, we use both single encoder and dual encoder architecture introduced above in order to eliminate the effect of model architecture. For dual encoder, since the amount of parameters in vanilla Transformer is originally smaller than that of TM-augmented NMT model, which makes the two models non-comparable (Yang et al., 2020), we use empty TMs to simulate vanilla Transformer and make the two models comparable. We set $N = 1$ and $k = 4$ in Algorithm 1 to estimate the variance, so we train models with up to 30k steps. The other settings are the same as the preliminary experiment

in §1. For single encoder, the configuration is the same as vanilla Transformer base model (Vaswani et al., 2017). And we also train models with up to 30k steps. For the experiment in §4, we use dual encoder architecture as in Table 3.

Settings for Main Experiment in §5 We build our model using Transformer blocks with the same configuration as Transformer Base (Vaswani et al., 2017) (8 attention heads, 512 dimensional hidden state, and 2048 dimensional feed-forward state). The number of Transformer blocks is 4 for the memory encoder, 6 for the source sentence encoder, and 6 for the decoder side. We retrieve the top 5 TM sentences. The FAISS index code is "IVF1024_HNSW32,SQ8" and the search depth is 64.

We follow the learning rate schedule, dropout, and label smoothing settings described in Vaswani et al. (2017). We use Adam optimizer (Kingma and Ba, 2015) and train models with up to 100K steps throughout all experiments. When fine-tuning models with the proposed weighted ensemble method, we only need to use 90% valid pairs to fine-tune with up to 2K steps and use the remaining 10% valid pairs to choose the checkpoint for testing.

Table 7 provides the number of parameters, training budget, and hyperparameters of each model. All experiments were performed on 8 V100 GPUs. We report the result of a single run for each experiment.

Packages Table 8 shows the packages we used for preprocessing, model training and evaluation.

E   Details about Data Statistics

In this paper, we use the JRC-Acquis corpus (Steinberger et al., 2006) and the re-split version of the Multi-Domain data set in Aharoni and Goldberg (2020) for our experiments.

JRC-Acquis For the JRC-Acquis corpus, it contains the total body of European Union (EU) law applicable to the EU member states. This corpus was also used by Gu et al. (2018); Zhang et al. (2018); Xia et al. (2019); Cai et al. (2021) and we managed to get the datasets originally preprocessed by Gu et al. (2018). Specifically, we select four translation directions, namely, Spanish⇒English (Es⇒En), En⇒Es, German⇒English (De⇒En), and En⇒De, for evaluation. Table 9 shows the detailed number of train/dev/test pairs for each language pair.

Dataset	#Train Pairs	#Dev Pairs	#Test Pairs
En ⇔ Es	679,088	2,533	2,596
En ⇔ Es	699,569	2,454	2,483

Table 9: Data statistics for the JRC-Acquis corpus.

Multi-Domain Data Set For the Multi-Domain data set, it includes German-English parallel data in five domains: Medical, Law, IT, Koran, and Subtitle. Table 10 shows the detailed number of train/dev/test pairs for each domain.

Dataset	#Train Pairs	#Dev Pairs	#Test Pairs
Medical	245,553	2,000	2,000
Law	459,721	2,000	2,000
IT	220,241	2,000	2,000
Koran	17,833	2,000	2,000
Subtitle	499,969	2,000	2,000

Table 10: Data statistics for the multi-domain dataset.

A For every submission:

☑ A1. Did you describe the limitations of your work?
Section Limitations.

☑ A2. Did you discuss any potential risks of your work?
Section Limitations and Section Ethics Statement.

☑ A3. Do the abstract and introduction summarize the paper's main claims?
Abstract and Section 1.

☒ A4. Have you used AI writing assistants when working on this paper?
Left blank.

B ☑ Did you use or create scientific artifacts?

Section 1, 3, 4, 5.

☑ B1. Did you cite the creators of artifacts you used?
Section 1, 3, 4, 5 and Appendix B, C, D, E.

☑ B2. Did you discuss the license or terms for use and / or distribution of any artifacts?
Section 1, 3, 4, 5 and Appendix B, C, D, E.

☑ B3. Did you discuss if your use of existing artifact(s) was consistent with their intended use, provided that it was specified? For the artifacts you create, do you specify intended use and whether that is compatible with the original access conditions (in particular, derivatives of data accessed for research purposes should not be used outside of research contexts)?
Section 1, 3, 4, 5 and Appendix B, C, D, E.

☑ B4. Did you discuss the steps taken to check whether the data that was collected / used contains any information that names or uniquely identifies individual people or offensive content, and the steps taken to protect / anonymize it?
Section Ethics Statement.

☑ B5. Did you provide documentation of the artifacts, e.g., coverage of domains, languages, and linguistic phenomena, demographic groups represented, etc.?
Section 1, 3, 4, 5 and Appendix B, C, D, E.

☑ B6. Did you report relevant statistics like the number of examples, details of train / test / dev splits, etc. for the data that you used / created? Even for commonly-used benchmark datasets, include the number of examples in train / validation / test splits, as these provide necessary context for a reader to understand experimental results. For example, small differences in accuracy on large test sets may be significant, while on small test sets they may not be.
Section 5 and Appendix E.

C ☑ Did you run computational experiments?

Section 1, 3, 4, 5.

☑ C1. Did you report the number of parameters in the models used, the total computational budget (e.g., GPU hours), and computing infrastructure used?
Appendix D.

☑ C2. Did you discuss the experimental setup, including hyperparameter search and best-found hyperparameter values?
Section 1, 3, 4, 5 and Appendix D.

☑ C3. Did you report descriptive statistics about your results (e.g., error bars around results, summary statistics from sets of experiments), and is it transparent whether you are reporting the max, mean, etc. or just a single run?
Section 1, 3, 4, 5.

☑ C4. If you used existing packages (e.g., for preprocessing, for normalization, or for evaluation), did you report the implementation, model, and parameter settings used (e.g., NLTK, Spacy, ROUGE, etc.)?
Section 1, 3, 4, 5 and Appendix D.

D ☒ Did you use human annotators (e.g., crowdworkers) or research with human participants?

Left blank.

☐ D1. Did you report the full text of instructions given to participants, including e.g., screenshots, disclaimers of any risks to participants or annotators, etc.?
No response.

☐ D2. Did you report information about how you recruited (e.g., crowdsourcing platform, students) and paid participants, and discuss if such payment is adequate given the participants' demographic (e.g., country of residence)?
No response.

☐ D3. Did you discuss whether and how consent was obtained from people whose data you're using/curating? For example, if you collected data via crowdsourcing, did your instructions to crowdworkers explain how the data would be used?
No response.

☐ D4. Was the data collection protocol approved (or determined exempt) by an ethics review board?
No response.

☐ D5. Did you report the basic demographic and geographic characteristics of the annotator population that is the source of the data?
No response.