Neural Machine Translation with Monolingual Translation Memory[*]

Deng Cai[♡], **Yan Wang**[♠], **Huayang Li**[♠], **Wai Lam**[♡], and **Lemao Liu**[♠]

[♡]The Chinese University of Hong Kong
thisisjcykcd@gmail.com
wlam@se.cuhk.edu.hk
[♠]Tencent AI Lab
{brandenwang,alanili,redmondliu}@tencent.com

Abstract

Prior work has proved that Translation memory (TM) can boost the performance of Neural Machine Translation (NMT). In contrast to existing work that uses bilingual corpus as TM and employs source-side similarity search for memory retrieval, we propose a new framework that uses monolingual memory and performs learnable memory retrieval in a cross-lingual manner. Our framework has unique advantages. First, the cross-lingual memory retriever allows abundant monolingual data to be TM. Second, the memory retriever and NMT model can be jointly optimized for the ultimate translation goal. Experiments show that the proposed method obtains substantial improvements. Remarkably, it even outperforms strong TM-augmented NMT baselines using bilingual TM. Owning to the ability to leverage monolingual data, our model also demonstrates effectiveness in low-resource and domain adaptation scenarios.

1 Introduction

Augmenting parametric neural network models with non-parametric memory (Khandelwal et al., 2019; Guu et al., 2020; Lewis et al., 2020a,b) has recently emerged as a promising direction to relieve the demand for ever-larger model size (Devlin et al., 2019; Radford et al., 2019; Brown et al., 2020). For the task of Machine Translation (MT), inspired by the Computer-Aided Translation (CAT) tools by professional human translators for increasing productivity for decades (Yamada, 2011), the usefulness of Translation Memory (TM) has long been recognized (Huang et al., 2021). In general, TM is a database that stores pairs of source text and its corresponding translations. Like for human translation, early work (Koehn and Senellart, 2010; He et al., 2010; Utiyama et al., 2011; Wang et al., 2013, inter alia) presents translations for similar source input to statistical translation models as additional cues.

Recent work has confirmed that TM can help Neural Machine Translation (NMT) models as well. In a similar spirit to prior work, TM-augmented NMT models do not discard the training corpus after training but keep exploiting it in the test time. These models perform translation in two stages: In the retrieval stage, a retriever searches for nearest neighbors (i.e., source-target pairs) from the training corpus based on source-side similarity such as lexical overlaps (Gu et al., 2018; Zhang et al., 2018; Xia et al., 2019), embedding-based matches (Cao and Xiong, 2018), or a hybrid (Bulte and Tezcan, 2019; Xu et al., 2020); In the generation stage, the retrieved translations are injected into a standard NMT model by attending over them with sophisticated memory networks (Gu et al., 2018; Cao and Xiong, 2018; Xia et al., 2019; He et al., 2021) or directly concatenating them to the source input (Bulte and Tezcan, 2019; Xu et al., 2020), or biasing the word distribution during decoding (Zhang et al., 2018). Most recently, Khandelwal et al. (2020) propose a token-level nearest neighbor search using complete translation context, i.e., both the source-side input and target-side prefix.

Despite their differences, we identify two major limitations in previous research. First, the translation memory has to be a *bilingual* corpus consisting of aligned source-target pairs. This requirement limits the memory bank to bilingual pairs and precludes the use of abundant monolingual data, which can be especially helpful for low-resource scenarios. Second, the memory retriever is *non-learnable*, not end-to-end optimized, and lacks for the ability to adapt to specific downstream NMT models. Concretely, current retrieval mechanisms (e.g., BM25)

[*]The work described in this paper is partially supported by a grant from the Research Grant Council of the Hong Kong Special Administrative Region, China (Project Code: 14200719).

are *generic* similarity search, adopting a simple heuristic. That is, the more a source sentence overlaps with the input sentence, the more likely its target-side translation pieces will appear in the correct translation. Although this observation is true, the most similar one does not necessarily serve the best for NMT models. Ideally, the retrieval metric would be learned from the data in a task-dependent way: we wish to consider a memory only if it can indeed boost the quality of final translation.

In this work, we propose to augment NMT models with *monolingual* TM and a *learnable cross-lingual* memory retriever. Specifically, we align source-side sentences and the corresponding target-side translations in a latent vector space using a simple dual-encoder framework (Bromley et al., 1993), such that the distance in the latent space yields a score function for retrieval. As a result, our memory retriever directly connects the dots between the source-side input and target-side translations, enabling *monolingual* data in the target language to be used alone as TM. Before running each translation, the memory retriever selects the highest-scored memories from a large collection of monolingual sentences (TM), which may include but are not limited to the target side of training corpus, and then the downstream NMT model attends over those memories to help inform its translation. We design the memory retriever with differentiable neural networks. To unify the memory retriever and its downstream NMT model into a learnable whole, the retrieval scores are used to bias the attention scores to the most useful retrieved memories. In this way, our memory retrieval can be end-to-end optimized for the translation objective: a retrieval that improves the golden translation's likelihood is helpful and should be rewarded, while an uninformative retrieval should be penalized.

One challenge for training our proposed framework is that, when starting from random initialization, the retrieved memories will likely be totally unrelated to the input. Since the memory retriever does not exert positive influence on NMT model's performance, it cannot receive a meaningful gradient and improve. This causes the NMT model to learn to ignore all retrieved memories. To avoid this cold-start problem, we propose to warm-start the retrieval model using two cross-alignment tasks.

Experiments show that (1) Our model leads to significant improvements over non-TM baseline NMT model, even outperforming strong TM-augmented baselines. This is remarkable given that previous TM-augmented models rely on bilingual TM while our model only exploits the target side. (2) Our model can substantially boost translation quality in low-resource scenarios by utilizing extra monolingual TM that is not present in training pairs. (3) Our model gains a strong cross-domain transferability by hot-swapping domain-specific monolingual memory.

2 Related Work

TM-augmented NMT This work contributes primarily to the research line of Translation Memory (TM) augmented Neural Machine Translation (NMT). Feng et al. (2017) augmented NMT with a bilingual dictionary to tackle infrequent word translation. Gu et al. (2018) proposed a model that retrieves examples similar to the test source sentence and encodes retrieved source-target pairs with key-value memory networks. Cao and Xiong (2018); Cao et al. (2019) used a gating mechanism to balance the impact of the translation memory. Zhang et al. (2018) proposed guiding models by retrieving n-grams and up-weighting the probabilities of retrieved n-grams. Bulte and Tezcan (2019) and Xu et al. (2020) used fuzzy-matching with translation memories and augment source sequences with retrieved source-target pairs. Xia et al. (2019) directly ignored the source side of a TM and packed the target side into a compact graph. Khandelwal et al. (2020) ran existing translation model on large bi-text corpora and recorded all hidden states for later nearest neighbor search at *each* decoding step, which is very compute-intensive. The distinctions between our work and prior work are obvious: (1) The TM in our framework is a collection of monolingual sentences rather than bilingual sentence pairs; (2) We use learnable task-specific retrieval rather than generic retrieval mechanisms.

Retrieval for Text Generation Discrete retrieval as an intermediate step has been shown beneficial to a variety of natural language processing tasks. One typical use is to retrieve supporting evidence for open-domain question answering (e.g., Chen et al., 2017; Lee et al., 2019; Karpukhin et al., 2020). Recently, retrieval-guided generation has gained increasing interest in a wide range of text generation tasks such as language modeling (Guu et al., 2018; Khandelwal et al., 2019; Guu et al., 2020), dialogue response generation (Weston et al., 2018; Wu et al., 2019; Cai et al.,

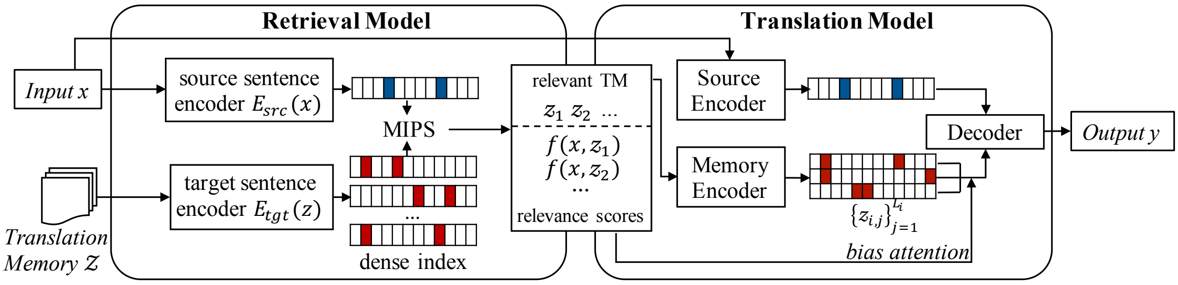

Figure 1: Overall framework. For an input sentence x in the source language, the retrieval model uses Maximum Inner Product Search (MIPS) to find the top-M TM sentences $\{z_i\}_{i=1}^{M}$ in the target language. The translation model takes $\{z_i\}_{i=1}^{M}$ and corresponding relevance scores $\{f(x, z_i)\}_{i=1}^{M}$ as input and generate the translation y.

2019a,b), code generation (Hashimoto et al., 2018) and other knowledge-intensive generation (Lewis et al., 2020b). It can be observed that there is a shift from using off-the-shelf search engines to learning task-specific retrievers. Our work draws inspiration from this line of research. However, retrieval-guided generation has so far been mainly investigated for knowledge retrieval in the same language. The memory retrieval in this work is more challenging due to the cross-lingual setting.

NMT using Monolingual Data To our knowledge, the integration of monolingual data for NMT was first investigated by Gulcehre et al. (2015), who separately trained target-side language models using monolingual data, and then integrated them during decoding either through re-scoring the beam, or by feeding the hidden state of the language model to the NMT model. Jean et al. (2015) also explored re-ranking the NMT output with a n-gram language model. Another successful method for leveraging monolingual data in NMT is *back-translation* (Sennrich et al., 2016; Fadaee et al., 2017; Edunov et al., 2018; He et al., 2016), where a reverse translation model is used to translate monolingual sentences from the target language to the source language to generate synthetic parallel sentences. Recent studies (Jiao et al., 2021; He et al., 2019) showed that *self-training*, where the synthetic parallel sentences are created by translating monolingual sentences in the source language, is also helpful. Our method is orthogonal to previous work and bears a unique feature: it can use more monolingual data without re-training (see §4.3).

3 Proposed Approach

We start by formalizing the translation task as a retrieve-then-generate process in §3.1. Then in §3.2, we describe the model design for the cross-lingual memory retrieval model. In §3.3, we describe the model design for the memory-augmented translation model. Lastly, we show how to optimize the two components jointly using standard maximum likelihood training in §3.4 and therein we address the cold-start problem via cross-alignment pre-training.

3.1 Overview

Our approach decomposes the whole translation processing into two steps: retrieve, then generate. The overall framework is illustrated in Figure 1. The Translation Memory (TM) in our approach is a collection of sentences in the target language $\mathcal{Z}$. Given an input x in the source language, the retrieval model first selects a number of possibly helpful sentences $\{z_i\}_{i=1}^{M}$ from $\mathcal{Z}$, where $M \ll |\mathcal{Z}|$, according to a relevance function $f(x, z_i)$. Then, the translation model conditions on both the retrieved set $\{(z_i, f(x, z_i))\}_{i=1}^{M}$ and the original input x to generate the output y using a probabilistic model $p(y|x, z_1, f(x, z_1), \ldots, z_M, f(x, z_M))$. Note that the relevance scores $\{f(x, z_i)\}_{i=1}^{M}$ are also part of the input to the translation model, encouraging the translation model to focus more on more relevant sentences. During training, maximizing the likelihood of the translation references improves both the translation model and the retrieval model.

3.2 Retrieval Model

The retrieval model is responsible for selecting the most relevant sentences for a source sentence from a large monolingual TM. This could involve measuring the relevance scores between the source sentence and millions of candidate target sentences, which poses a serious computational challenge. To address this, we implement the retrieval model using a simple dual-encoder framework (Bromley et al., 1993) such that the selection of the most relevant sentences can be reduced to Maximum Inner Product Search (MIPS). With performant data structures and search algorithms (e.g., Shrivastava and Li, 2014; Malkov and Yashunin, 2018), the retrieval can be done efficiently.

Specifically, we define the relevance score $f(x, z)$ between the source sentence x and the candidate sentence z as the dot product of their dense vector representations:

$$f(x, z) = E_{\text{src}}(x)^{\text{T}} E_{\text{tgt}}(z)$$

where E_{src} and E_{tgt} are the source sentence encoder and the target sentence encoder that map x and z to d-dimensional vectors respectively. We implement the two sentence encoders using two independent Transformers (Vaswani et al., 2017). For an input sentence, we prepend the `[BOS]` token to its token sequence and then feed it into a Transformer. We take the representation at the `[BOS]` token as the output (denoted $\text{Trans}_{\{\text{src,tgt}\}}(\{x, z\})$), and perform a linear projection ($W_{\{\text{src,tgt}\}}$) to reduce the dimensionality of the vector. Finally, we normalize the vectors to regulate the range of relevance scores.

$$E_{\text{src}}(x) = \text{normalize}(W_{\text{src}} \text{Trans}_{\texttt{src}}(x))$$
$$E_{\text{tgt}}(z) = \text{normalize}(W_{\text{tgt}} \text{Trans}_{\texttt{tgt}}(z))$$

The normalized vectors have zero means and unit lengths. Therefore, the relevance scores always fall in the interval $[-1, 1]$. We let θ denote all parameters associated with the retrieval model.

In practice, the dense representations of all sentences in TM can be pre-computed and indexed using FAISS (Johnson et al., 2019), an open-source toolkit for efficient vector search. Given a source sentence x in hand, we compute the vector representation $v_x = E_{\text{src}}(x)$ and retrieve the top M target sentences with vectors closest to v_x.

3.3 Translation Model

Given a source sentence x, a small set of relevant TM $\{z_i\}_{i=1}^{M}$, and relevance scores $\{f(x, z_i)\}_{i=1}^{M}$, the translation model defines the conditional probability $p(y|x, z_1, f(x, z_1), \ldots, z_M, f(x, z_M))$.

Our translation model is built upon the standard encoder-decoder NMT model (Bahdanau et al., 2015; Vaswani et al., 2017): the (source) encoder transforms the source sentence x into dense vector representations. The decoder generates an output sequence y in an auto-regressive fashion. At each time step t, the decoder attends over both previously generated sequence $y_{1:t-1}$ and the output of the source encoder, generating a hidden state h_t. The hidden state h_t is then converted to next-token probabilities through a linear projection followed by softmax function, i.e., $P_v = \text{softmax}(W_v h_t + b_v)$.

To accommodate the extra memory input, we extend the standard encoder-decoder NMT framework with a memory encoder and allow cross-attention from the decoder to the memory encoder. Specifically, the memory encoder encodes each TM sentence z_i individually, resulting in a set of contextualized token embeddings $\{z_{i,k}\}_{k=1}^{L_i}$, where L_i is the length of the token sequence z_i. We compute a cross attention over all TM sentences:

$$\alpha_{ij} = \frac{\exp(h_t{}^{\text{T}} W_m z_{i,j})}{\sum_{i=1}^{M} \sum_{k=1}^{L_i} \exp(h_t{}^{\text{T}} W_m z_{i,k})} \quad (1)$$

$$c_t = W_c \sum_{i=1}^{M} \sum_{j=1}^{L_i} \alpha_{ij} z_{i,j}$$

where α_{ij} is the attention score of the j-th token in z_i, c_t is a weighted combination of memory embeddings, and W_m and W_c are trainable matrices. The cross attention is used twice during decoding. First, the decoder's hidden state h_t is updated by a weighted sum of memory embeddings, i.e., $h_t = h_t + c_t$. Second, we consider each attention score as a probability of copying the corresponding token (Gu et al., 2016; See et al., 2017). Formally, the next-token probabilities are computed as:

$$p(y_t|\cdot) = (1 - \lambda_t) P_v(y_t) + \lambda_t \sum_{i=1}^{M} \sum_{j=1}^{L_i} \alpha_{ij} \mathbb{1}_{z_{ij}=y_t}$$

where $\mathbb{1}$ is the indicator function and λ_t is a gating variable computed by another feed-forward network $\lambda_t = g(h_t, c_t)$.

Inspired by Lewis et al. (2020a), to enable the gradient flow from the translation output to the retrieval model, we bias the attention scores with the relevance scores, rewriting Eq. (1) as:

$$\alpha_{ij} = \frac{\exp(h_t{}^{\text{T}} W_m z_{i,j} + \beta f(x, z_i))}{\sum_{i=1}^{M} \sum_{k=1}^{L_i} \exp(h_t{}^{\text{T}} W_m z_{i,k} + \beta f(x, z_i))} \quad (2)$$

where β is a trainable scalar that controls the weight of the relevance scores. We let ϕ denote all parameters associated with the translation model.

3.4 Training

We optimize the model parameters θ and ϕ using stochastic gradient descent on the negative log-likelihood loss function $-\log p(y^*|x, z_1, f(x, z_1), \ldots, z_M, f(x, z_M))$, where y^* refers to the reference translation. As implied by Eq. (2), TM sentences that improve the likelihood of reference translations should receive higher attention scores and higher relevance scores, so gradient descent on the loss function will improve the quality of the retrieval model as well.

Cross-alignment Pre-training However, if the retrieval model starts from random initialization, all top TM sentences z_i will likely be unrelated to x (or equally useless). This leads to a problem that the retrieval model cannot receive meaningful gradients and improve, and the translation model will learn to completely ignore the TM input. To avoid this cold-start problem, we propose two cross-alignment tasks to warm-start the retrieval model.

The first task is sentence-level cross-alignment. This task aims to find the right translation for a source sentence given a set of other translations, which is directly related to our retrieval function. Concretely, We sample B source-target pairs from the training corpus at each training step. Let X and Z be the $(B \times d)$ matrix of the source and target vectors encoded by E_{src} and E_{tgt} respectively. $S = XZ^T$ is a $(B \times B)$ matrix of relevance scores, where each row corresponds to a source sentence and each column corresponds to a target sentence. Any (X_i, Z_j) pair should be aligned when $i = j$, and should not otherwise. The objective is to maximize the scores along the diagonal of the matrix and henceforth reduce the values in other entries. The loss function can be written as:

$$\mathcal{L}^{(i)}_{\texttt{snt}} = \frac{-\exp(S_{ii})}{\exp(S_{ii}) + \sum_{j \neq i} \exp(S_{ij})}.$$

The second task is token-level cross-alignment, which aims to predict the tokens in the target language given the source sentence representation and vice versa. Formally, we use bag-of-words losses:

$$\mathcal{L}^{(i)}_{\texttt{tok}} = -\sum_{w_y \in \mathcal{Y}_i} \log p(w_y|X_i) + \sum_{w_x \in \mathcal{X}_i} \log p(w_x|Y_i)$$

where $\mathcal{X}_i$ ($\mathcal{Y}_i$) represents the set of tokens in the i-th source (target) sentence and the token probabilities are computed by a linear projection followed by the softmax function. The joint loss for pre-training is $\frac{1}{B}\sum_{i=1}^{B} \mathcal{L}^{(i)}_{\texttt{snt}} + \mathcal{L}^{(i)}_{\texttt{tok}}$. In practice, we find that both the sentence-level and token-level objectives are crucial for achieving superior performance.

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

Table 1: Data statistics for the JRC-Acquis corpus.

Asynchronous Index Refresh To employ fast MIPS, we must pre-compute $E_{\text{tgt}}(z)$ for every $z \in \mathcal{Z}$ and build an index. However, the index cannot remain consistent with the running model during training as θ will be updated over time. One straightforward solution to fix the parameters of E_{tgt} after the pre-training described above and only fine-tune the parameters of E_{src}. However, this may hurt performance since E_{tgt} cannot adapt to the translation objective. Another solution is to asynchronously refresh the index by re-computing and re-indexing all TM sentences at regular intervals. The index is slightly outdated between refreshes, however, we use fresh E_{tgt} in gradient estimate. We explore both options in our experiments.

4 Experiments

We experiment with the proposed approach in three settings: (1) the conventional setting where the available TM is limited to the bilingual training corpus, (2) the low-resource setting where bilingual training pairs are scarce but extra monolingual data is exploited as additional TM, and (3) non-parametric domain adaptation using monolingual TM. Note that existing TM-augmented NMT models are only applicable to the first setting, the last two settings only become possible with our proposed model. We use BLEU score (Papineni et al., 2002) as the evaluation metric.

4.1 Implementation Details

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 3 for the retrieval model, 4 for the memory encoder in the translation model, and 6 for the encoder-decoder architecture in the translation model. 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, 2014) and train models with up to 100K

#	System	Retriever	Es⇒En		En⇒Es		De⇒En		En⇒De	
			Dev	Test	Dev	Test	Dev	Test	Dev	Test
		*Existing NMT systems**								
	Gu et al. (2018)	source similarity	63.16	62.94	-	-	-	-	-	-
	Zhang et al. (2018)	source similarity	63.97	64.30	61.50	61.56	60.10	60.26	55.54	55.14
	Xia et al. (2019)	source similarity	66.37	66.21	62.50	62.76	61.85	61.72	57.43	56.88
		Our NMT systems								
1	*this work*	None	64.25	64.07	62.27	61.54	59.82	60.76	55.01	54.90
2		source similarity	66.98	66.48	63.04	62.76	63.62	63.85	57.88	57.53
3		cross-lingual (fixed)	66.68	66.24	63.06	62.73	63.25	63.06	57.61	56.97
4		cross-lingual (fixed E_{tgt})†	67.66	67.16	63.73	63.22	64.39	64.01	58.12	57.92
5		cross-lingual†	**67.73**	**67.42**	**64.18**	**63.86**	**64.48**	**64.62**	**58.77**	**58.42**

Table 2: Experimental results (BLEU scores) on four translation tasks. *Results are from Xia et al. (2019). †The two variants of our method (model #4 and model #5) are significantly better than other baselines with p-value $<$ 0.01, tested by bootstrap re-sampling (Koehn, 2004).

steps throughout all experiments. When trained with asynchronous index refresh, the re-indexing interval is 3K training steps.[1]

4.2 Conventional Experiments

Following prior work in TM-augmented NMT, we first conduct experiments in a setting where the bilingual training corpus is the only source for TM.

Data We use the JRC-Acquis corpus (Steinberger et al., 2006) for our experiments. The JRC-Acquis corpus 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) and we managed to get the datasets originally preprocessed by Gu et al. (2018), making it possible to fairly compare our results with previously reported BLEU scores. Specifically, we select four translation directions, namely, Spanish⇒English (Es⇒En), En⇒Es, German⇒English (De⇒En), and En⇒De, for evaluation. Detailed data statistics are shown in Table 1.

Models To study the effect of each model component, we implement a series of model variants (model #1 to #5 in Table 2).

1. NMT without TM. To measure the help from TM, we remove the model components related to TM (including the retrieval model and the memory encoder), and only employ the encoder-decoder architecture for NMT. The resulted model is equivalent to the Transformer Base model (Vaswani et al., 2017).
2. TM-augmented NMT using source similarity search. To isolate the effect of architectural changes in NMT models, we replace our cross-lingual memory retriever with traditional source-side similarity search. Specifically, we use the fuzzy match system used in Xia et al. (2019) and many others, which is based on BM25 and edit distance.
3. TM-augmented NMT using pre-trained cross-lingual retriever. To study the effect of end-to-end task-specific optimization of the retrieval model, we pre-train the retrieval model using the cross-alignment tasks introduced in §3.4 and keep it fixed in the following NMT training.
4. Our full model using a fixed TM index; After pre-training, we fix the parameter of E_{tgt} during NMT training.
5. Our full model trained with asynchronous index refresh.

Results The results of the above models are presented in Table 2. We have the following observations: (1) Our full model trained with asynchronous index refresh (model #5) delivers the best performance on test sets across all four translation tasks, outperforming the non-TM baseline (model #1) by 3.26 BLEU points in average and up to 3.86 BLEU points (De⇒En). This result confirms that monolingual TM can boost NMT performance; (2) The end-to-end learning of the retriever model is the key for substantial performance improvement. We can see that using a pre-trained fixed cross-lingual retriever only gives moderate test performance, fine-tuning E_{src} and fixing E_{tgt} significantly boosts the performance, and fine-tuning both E_{src}

[1] Our code is released at `https://github.com/jcyk/copyisallyouneed`.

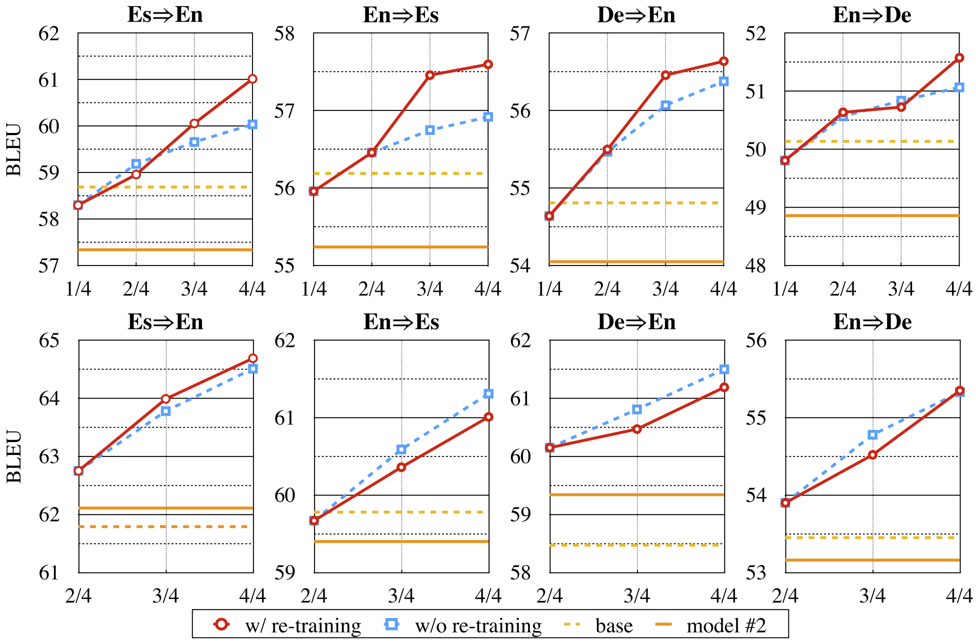

Figure 2: Test results with 1/4 bilingual pairs (upper) and 2/4 bilingual pairs (lower) across different TM sizes.

and E_{tgt} leads to the strongest performance (model #5>model #4>model #3); (3) Cross-lingual retrieval (model #4 and model #5) can obtain better results than that of the source similarity search (model #2). This is remarkable since the cross-lingual retrieval only requires monolingual TM, while the source similarity search relies on bilingual TM. We attribute the success, again, to the end-to-end adaptability of our cross-lingual retriever. This is manifested by the fact that model #3 even slightly underperforms model #2 in some of translation tasks.

Contrast to Previous Bilingual TM Systems We also compare our results with the best previously reported models.[2] We can see that our results significantly outperform previous arts. Notably, our best model (model #5) surpasses the best reported model (Xia et al., 2019) by 1.69 BLEU points in average and up to 2.9 BLEU points (De⇒En). This result verifies the effectiveness of our proposed models. In fact, we can see that our translation model using traditional similarity search (model #2) already outperforms the best previously reported results, which reveals that the architectural design of our translation model is surprisingly effective despite its simplicity.

4.3 Low-Resource Scenarios

One most unique characteristic of our proposed model is that it uses monolingual TM. This motivates us to conduct experiments in low-resource scenarios, where we use extra monolingual data in the target language to boost translation quality.

Data We create low-resource scenarios by randomly partitioning each training set in JRC-Acquis corpus into four subsets of equal size. We set up two series of experiments: (1) We only use the bilinguals pairs in the first subset and gradually enlarge the TM by including more monolingual data in other subsets. (2) Similar to (1), but we instead use the bilingual pairs in the first two subsets.

Models As shown in §4.2, the model trained with asynchronous index refresh (model #5) is slightly better than the model using fixed E_{tgt} (model #4), however, the computational cost of training model #5 is much bigger. For simplicity and environmental consideration, we only test model #4 in low-resource scenarios. Nevertheless, we note there are still two modeling choices: (1) train the model once with the TM limited to training pairs and only enlarge the TM during testing; (2) re-train the model with every enlarged TM. Note that when using the first choice, the model may retrieve a TM sentence that has never been seen during training. To measure the performance improvements from additional monolingual TM, we also include a Transformer Base baseline (model #1, denoted as

[2] Some recent work used different datasets other than JRC-Acquis with unspecified data split, which makes it hard to make an exhaustive comparison. However, note that our in-house baseline (model #2) is quite strong.

Data	Model	Es⇒En		En⇒Es		De⇒En		En⇒De	
		dev	test	dev	test	dev	test	dev	test
1/4 bilingual + 4/4 monolingual	Ours	61.46	61.02	57.86	57.40	56.77	56.54	51.11	51.58
	BT	62.47	61.99	60.28	59.59	57.75	58.20	52.47	52.96
	Ours+BT	**65.98**	**65.51**	**62.48**	**62.22**	**62.22**	**61.79**	**56.75**	**56.50**
2/4 bilingual + 4/4 monolingual	Ours	65.17	64.69	61.31	61.01	61.43	61.19	55.55	55.35
	BT	63.82	63.10	61.59	60.83	59.17	59.26	54.18	54.29
	Ours+BT	**66.95**	**66.38**	**63.22**	**62.90**	**63.68**	**63.10**	**57.69**	**57.40**

Table 3: Comparison with back-translation (BT).

	Medical	Law	IT	Koran	Subtitle	Avg.	Avg. Δ
#Bilingual Pairs	61,388	114,930	55,060	4,458	124,992	-	-
#Monolingual Sents	184,165	344,791	165,181	13,375	374,977	-	-
Using Bilingual Pairs Only							
Transformer Base	47.81	51.40	33.90	14.64	21.64	33.88	-
Ours	47.52	51.17	34.64	15.49	22.66	34.30	+0.42
+ Monolingual Memory							
Ours + domain-specific	**50.32**	53.97	**35.33**	**16.26**	**22.78**	**35.73**	**+1.85**
Ours + all-domains	50.23	**54.12**	35.24	16.24	**22.78**	35.72	+1.84

Table 4: Test results on domain adaptation.

base) and a bilingual TM baseline (model #2).

Results Figure 2 shows the main results on the test sets. The general patterns are consistent across all experiments: the larger the TM becomes, the better translation performance the model achieves. When using all available monolingual data (4/4), the translation quality is boosted significantly. Interestingly, the performance of models without re-training is comparable to, if not better than, those with re-training. We also observe that when the training pairs are very scarce (only 1/4 bilingual pairs are available), a small size of TM even hurts the model performance. The reason could be overfitting. We speculate that better results would be obtained by tuning the model hyper-parameters according to different TM sizes.

Contrast to Back-Translation We compare our models with back-translation (BT) (Sennrich et al., 2016), a popular way of utilizing monolingual data for NMT. We train a target-to-source Transformer Base model using bilingual pairs and use the resultant model to translate monolingual sentences to obtain additional synthetic parallel data. As shown in Table 3, our method performs better than BT with 2/4 bilingual pairs but performs worse with 1/4 bilingual pairs. Interestingly, the combination of BT and our method yields significant further gains, which demonstrates that our method is not only orthogonal but also complementary to BT.

4.4 Non-parametric Domain Adaptation

Lastly, the *"plug and play"* property of TM further motivates us to domain adaptation, where we adapt a *single* general-domain model to a specific domain by using domain-specific monolingual TM.

Data To simulate a diverse multi-domain setting, we use the data splits in Aharoni and Goldberg (2020) originally collected by Koehn and Knowles (2017). It includes German-English parallel data for train/dev/test sets in five domains: Medical, Law, IT, Koran and Subtitles. Similar to the experiments in §4.3, we only use one fourth of bilingual pairs for training. The target side of the remaining data is treated as additional monolingual data for building domain-specific TM, and the source side is discarded. The data statistics can be found in the upper block of Table 4. The dev and test sets for each domain contains 2K instances.

Models We first train a Transformer Base baseline (model #1) on the concatenation of bilingual pairs in all domains. As in §4.3, we train our model using fixed E_{tgt} (model #4). One advantage of our approach is the possibility of training a single model which can be adapted to any new domain at the inference time without any re-training, by just switching the TM. When adapting to a new TM, we do not re-train our model. As the purpose here is to verify that our approach can tackle domain adaptation *without any domain-specific training*, we leave the comparison and combination of other domain adaptation techniques (Moore and Lewis, 2010; Chu and Wang, 2018) as future work.

Results The results are presented in Table 4. We can see that when only using the bilingual data, the TM-augmented model obtains higher BLEU scores in domains with less data but slightly lower scores in other domains compared to the non-TM baseline. However, as we switch the TM to domain-specific TM, the translation quality is significantly boosted in all domains, improving the non-TM baseline by an average of 1.85 BLEU points, with improvements as large as 2.57 BLEU points on Law and 2.51 BLEU point on Medical. We also attempt to combine all domain-specific TMs to one and use it for all domains (the last row in Table 4). However, we do not obtain noticeable improvement. This reveals that the out-of-domain data can provide little help so that a smaller in-domain TM is sufficient, which is also confirmed by the fact that about 90.21% of the retrieved sentences come from the corresponding domain in the combined TM.

4.5 Running Speed

With the help of FAISS in-GPU index, search over millions of vectors can be made incredibly efficient (often in tens of milliseconds). In our implementation, the memory search performs even faster than naive BM25[3]. For the results in Table 2, taking the vanilla Transformer Base model (model #1) as the baseline. The inference latency of our models (both model #4 and model #5) is about 1.36 times of the baseline (all use a single Nividia V100 GPU). Note that the corresponding number for the previous state-of-the-art model (Xia et al., 2019) is 1.80. As for training cost, the averaged time cost per training step of model #4 and model #5 is 2.62 times and 2.76 times of the baseline respectively, which are on par with traditional TM-augmented baselines (model #2 is 2.59 times) (all use two Nividia V100 GPUs). Table 5 presents the results. In addition, we also observe that memory-augmented models converge much faster than vanilla models in terms of training steps.

5 Conclusion

We introduced an effective approach that augments NMT models with monolingual TM. We show that a task-specific cross-lingual memory retriever can be learned by end-to-end MT training. Our approach achieves new state-of-the-art results on several datasets, leads to large gains in low-resource scenarios where the bilingual data is limited, and can specialize a NMT model for specific domains without further training.

#	Model	Training	Inference
1	Transformer Base	1.00x	1.00x
2	source similarity	2.59x	-
4	cross-lingual (fixed E_{tgt})	2.62x	1.36x
5	cross-lingual	2.76x	1.36x
-	Xia et al. (2019)	-	1.80x

Table 5: Latency cost for training and inference. For training, we measure the averaged time cost per training step. The number of Xia et al. (2019) is inferred from their paper.

Future work should aim to build over our proposed framework. Two obvious directions are: (1) Even though our experiments validated that the whole framework can be learned from scratch using standard MT corpora, it is possible to initialize each model component in our framework with massively pre-trained models for performance enhancement; and (2) The NMT model can benefit from aggregating over a set of diverse memories, which is not explicitly encouraged in current design.

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.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR*.

Jane Bromley, Isabelle Guyon, Yann LeCun, Eduard Säckinger, and Roopak Shah. 1993. Signature verification using a" siamese" time delay neural network. In *Proceedings of the 6th International Conference on Neural Information Processing Systems*, pages 737–744.

T. Brown, B. Mann, Nick Ryder, Melanie Subbiah, J. Kaplan, P. Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, G. Krüger, Tom Henighan, R. Child, Aditya Ramesh, D. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, E. Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, J. Clark, Christopher Berner, Sam McCandlish, A. Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *ArXiv*, abs/2005.14165.

[3] Elasticsearch Implementation: `https://www.elastic.co/`

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.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2019a. Skeleton-to-response: Dialogue generation guided by retrieval memory. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1219–1228.

Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, and Shuming Shi. 2019b. Retrieval-guided dialogue response generation via a matching-to-generation framework. 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 1866–1875.

Qian Cao, Shaohui Kuang, and Deyi Xiong. 2019. Learning to reuse translations: Guiding neural machine translation with examples. *arXiv preprint arXiv:1911.10732*.

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.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading Wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Chenhui Chu and Rui Wang. 2018. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186.

Sergey Edunov, Myle Ott, Michael Auli, and David Grangier. 2018. Understanding back-translation at scale. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 489–500.

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.

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.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor O.K. Li. 2016. Incorporating copying mechanism in sequence-to-sequence learning. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1631–1640.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *AAAI*, pages 5133–5140.

Caglar Gulcehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loic Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *arXiv preprint arXiv:1503.03535*.

Kelvin Guu, Tatsunori B. Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. 2020. Realm: Retrieval-augmented language model pre-training. *arXiv preprint arXiv:2002.08909*.

Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems*, pages 10052–10062.

Di He, Yingce Xia, Tao Qin, Liwei Wang, Nenghai Yu, Tie-Yan Liu, and Wei-Ying Ma. 2016. Dual learning for machine translation. *Advances in neural information processing systems*, 29:820–828.

Junxian He, Jiatao Gu, Jiajun Shen, and Marc'Aurelio Ranzato. 2019. Revisiting self-training for neural sequence generation. In *International Conference on Learning Representations*.

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*.

Yifan He, Yanjun Ma, Josef van Genabith, and Andy Way. 2010. Bridging smt and tm with translation recommendation. In *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics*, pages 622–630.

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*.

Sébastien Jean, Orhan Firat, Kyunghyun Cho, Roland Memisevic, and Yoshua Bengio. 2015. Montreal neural machine translation systems for wmt'15. In *Proceedings of the Tenth Workshop on Statistical Machine Translation*, pages 134–140.

Wenxiang Jiao, Xing Wang, Zhaopeng Tu, Shuming Shi, R. Michael Lyu, and Irwin King. 2021. Self-training sampling with monolingual data uncertainty for neural machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics*.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 6769–6781.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. *arXiv preprint arXiv:2010.00710*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

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

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 conference on empirical methods in natural language processing*, pages 388–395.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. *arXiv preprint arXiv:1706.03872*.

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

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Mike Lewis, Marjan Ghazvininejad, Gargi Ghosh, Armen Aghajanyan, Sida Wang, and Luke Zettlemoyer. 2020a. Pre-training via paraphrasing. *Advances in Neural Information Processing Systems*, 33.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.

Yury A Malkov and Dmitry A Yashunin. 2018. Efficient and robust approximate nearest neighbor search using hierarchical navigable small world graphs. *IEEE transactions on pattern analysis and machine intelligence*.

Robert C. Moore and William Lewis. 2010. Intelligent selection of language model training data. In *Proceedings of the ACL 2010 Conference Short Papers*, pages 220–224.

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.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. 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.

Anshumali Shrivastava and Ping Li. 2014. Asymmetric lsh (alsh) for sublinear time maximum inner product search (mips). *Advances in neural information processing systems*, 27:2321–2329.

Ralf Steinberger, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaz Erjavec, Dan Tufis, and Dániel Varga. 2006. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. *arXiv preprint cs/0609058*.

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, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in neural information processing systems*, 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 (Volume 1: Long Papers)*, pages 11–21.

Jason Weston, Emily Dinan, and Alexander Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. In *Proceedings of the 2018 EMNLP Workshop SCAI: The 2nd International Workshop on Search-Oriented Conversational AI*, pages 87–92.

Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. 2019. Response generation by context-aware prototype editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7281–7288.

Mengzhou Xia, Guoping Huang, Lemao Liu, and Shuming Shi. 2019. Graph based translation memory for neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, 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.

Masaru Yamada. 2011. The effect of translation memory databases on productivity. *Translation research projects*, 3:63–73.

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.