Neural Machine Translation with Contrastive Translation Memories

Xin Cheng [1,2], **Shen Gao** [3], **Lemao Liu** [4], **Dongyan Zhao** [1,2,5,6*], **Rui Yan** [7*]
[1] Wangxuan Institute of Computer Technology, Peking University
[2] Center for Data Science, Peking University [4] Tencent AI Lab
[3] School of Computer Science and Technology, Shandong University
[5] State Key Laboratory of Media Convergence Production Technology and Systems
[6]Beijing Institute of General Artificial Intelligence (BIGAI)
[7] Gaoling School of Artificial Intelligence, Renmin University of China
chengxin1998@stu.pku.edu.cn shengao@sdu.edu.cn
zhaody@pku.edu.cn redmondliu@tencent.com ruiyan@ruc.edu.cn

Abstract

Retrieval-augmented Neural Machine Translation models have been successful in many translation scenarios. Different from previous works that make use of mutually similar but redundant translation memories (TMs), we propose a new retrieval-augmented NMT to model contrastively retrieved translation memories that are holistically similar to the source sentence while individually contrastive to each other providing maximal information gains in three phases. First, in TM retrieval phase, we adopt a contrastive retrieval algorithm to avoid redundancy and uninformativeness of similar translation pieces. Second, in memory encoding stage, given a set of TMs we propose a novel Hierarchical Group Attention module to gather both local context of each TM and global context of the whole TM set. Finally, in training phase, a Multi-TM contrastive learning objective is introduced to learn salient feature of each TM with respect to target sentence. Experimental results show that our framework obtains improvements over strong baselines on the benchmark datasets.

1 Introduction

Translation memory (TM) is basically a database of segmented and paired source and target texts that translators can access in order to re-use previous translations while translating new texts (Christensen and Schjoldager, 2010). For human translators, such similar translation pieces can lead to higher productivity and consistency (Yamada, 2011). For machine translation, early works mainly contributes to employ TM for statistical machine translation (SMT) systems (Simard and Isabelle, 2009; Utiyama et al., 2011; Liu et al., 2012, 2019). Recently, as neural machine translation (NMT) model (Sutskever et al., 2014; Vaswani et al., 2017) has achieved impressive performance in many translation tasks, there is also an emerging interest (Gu et al., 2018) in retrieval-augmented NMT model.

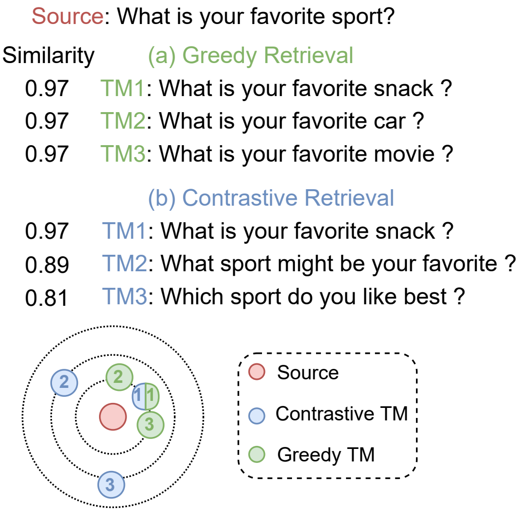
Source: What is your favorite sport?
Similarity
(a) Greedy Retrieval
0.97 TM1: What is your favorite snack ?
0.97 TM2: What is your favorite car ?
0.97 TM3: What is your favorite movie ?
(b) Contrastive Retrieval
0.97 TM1: What is your favorite snack ?
0.89 TM2: What sport might be your favorite ?
0.81 TM3: Which sport do you like best ?
2
2
1
1
3
3
Source
Contrastive TM
Greedy TM

Figure 1: An example of *Greedy Retrieval* and *Contrastive Retrieval*. The similarity score is computed by edit distance detailed in Section 3.1. And the target side of TM is omitted for brevity.

The key idea of retrieval-augmented NMT mainly includes two steps: a retrieval metric is used to retrieve similar translation pairs (i.e., TM), and the TM is then integrated into an NMT model. In the first step, a standard retrieval method greedily chooses the most similar translation memory one by one solely based on similarity with the source sentence (namely *Greedy Retrieval*). This method would inevitably retrieve translation memories that are mutually similar but redundant and uninformative as shown in Figure 1. Intuitively, it is promising to retrieve a diverse translation memory which would offer maximal coverage of the source sentence and provide useful cues from different aspects. Unfortunately, empirical experiments in Gu et al. (2018) show that a diverse translation memory only leads to negligible improvements. As a

[*] Corresponding author.

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3591–3601
December 7-11, 2022 ©2022 Association for Computational Linguistics

result, greedy retrieval is adopted in almost all later studies (Cao and Xiong, 2018; Xia et al., 2019; Xu et al., 2020; He et al., 2021; Cai et al., 2021; Khandelwal et al., 2020).

This paper aims to ask an important question *whether diverse translation memories are beneficial for retrieval-augmented NMT*. To this end, we propose a powerful retrieval-augmented NMT model called Contrastive Memory Model which takes into account diversity in translation memory from three ways. Specifically, (1) during TM retrieval, inspired by Maximal Marginal Relevance (MMR) (Carbonell and Goldstein, 1998), we introduce a conceptually simple while empirically useful retrieval method called *Contrastive Retrieval* to find informative translation memories. The core is to retrieve a cluster of translation memories that are similar to the source sentence while contrastive to each other keeping inner-cluster uniformity in the latent semantic space, as shown in Figure 1. (2) In TM encoding, given multiple translation memories, the local and global information should both be captured by the translation model. Separately encoding (Gu et al., 2018; He et al., 2021; Cai et al., 2021) or treating them as a long sequence (Xu et al., 2020) would inevitably lose such hierarchical structure information. Thus, to facilitate the direct communication between different translation memories for local information and gather the global context via message passing mechanism, we propose a Hierarchical Group Attention (HGA) module to encode the diverse memories. (3) In the model training phase, to learn salient and distinct features of each TM with respect to target sentence, we devise a novel Multi-TM Contrastive Learning objective (MTCL), which further contributes to a uniformly distributed translation memory cluster by forcing representation of every translation memory to approach the sentence to be translated while keep away from each other.

To verify the effectiveness of our framework, we conduct extensive experiments on four benchmark datasets, and observe substantial improvement over strong baselines, proving that diverse translation memories is indeed useful to NMT. Our main contributions are:

- We answer an important question about retrieval-augmented NMT, i.e., is diverse translation memory beneficial for retrieval-augmented NMT?
- We propose a diverse-TM-aware framework to improve a retrieval-augmented NMT system from three ways including TM retrieval, TM encoding and model training.
- We conduct extensive experiments on four translation directions, observing substantial performance gains over strong baselines with greedy retrieval.

2 Related Work

TM-augmented NMT Augmenting neural machine translation model with translation memories is an important line of work to boost the performance of the NMT model with non-parametric method. Feng et al. (2017) stores memories of infrequently encountered words and utilizes them to assist the neural model. Gu et al. (2018) uses an external memory network and a gating mechanism to incorporate TM. Cao and Xiong (2018) uses an extra GRU-based memory encoder to provide additional information to the decoder. Xia et al. (2019) adopts a compact graph representation of TM and perform additional attention mechanisms over the graph when decoding. Bulté and Tezcan (2019) and Xu et al. (2020) directly concatenate TM with source sentence using cross-lingual vocabulary. Zhang et al. (2018) augments the model with an additional bonus given to outputs that contain the collected translation pieces. There is also a line of work that trains a parametric retrieval model and a translation model jointly (Cai et al., 2021) and achieves impressive results. Recently, with rapid growth of computational power, a more fine grained token level translation memories are use in Khandelwal et al. (2020). This approach gives the decoder direct access to billions of examples at test time, achieving state-of-the-art result even without further training.

Contrastive Learning The key of contrastive learning (Hadsell et al., 2006; Mikolov et al., 2013) is to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors. Chen et al. (2020) and He et al. (2020) show that contrastive learning can boost the performance of self-supervised and semi-supervised learning in computer vision tasks. In natural language processing, Word2Vec (Mikolov et al., 2013) uses noise-contrastive estimation to learn better word representation. Gao et al. (2021) adopts contrastive learning with a simple token level dropout to greatly advance the state-of-the-art sentence embeddings. Liu and Liu (2021) uses contrastive loss to post-rank generated summaries and achieves promising results in benchmark datasets. Lee et al. (2020) and Pan et al. (2021) also use contrastive learning in translation tasks and observe consistent improvements.

3 Proposed Framework

Preliminary Assuming we are given a source sentence $x = \{\mathrm{x}_1, ..., \mathrm{x}_s\}$ and its corresponding target sentence $y = \{\mathrm{y}_1, ..., \mathrm{y}_t\}$ where s, t are their respective length. For a TM-augmented NMT model, a set of similar translation pairs $M = \{(x^m, y^m)\}_{m=1}^{|M|}$ are retrieved based on certain criterion $\mathbb{C}$ and NMT models the conditional probability of target sentence y conditioned on both source sentence x and translation memories M in a left-to-right manner:

$$P(Y = y|X = x) = \prod_{t=1}^{|T|} P(\mathrm{y}_t|\mathrm{y}_0, ...\mathrm{y}_{t-1}; x; M) \tag{1}$$

Overview Given a source sentence x and informative translation memories M , the translation model defines the conditional probability similar to the Equation 1. At the high level, our framework, as shown in Figure 2, consists of *contrastive retrieval*, which searches a diverse translation memory, *source encoder* which transforms source sentence x into dense vector representations z^x, *memory encoder* with hierarchical group attention module to jointly encode $|M|$ translation memories into a series dense representation z^m and *decoder* which attends to both z^x and z^m and generates target sentence y in an auto-regressive fashion, and *contrastive learning* which effectively trains the memory encoder as well as source encoder and decoder. *Among all these five modules, contrastive memory (§3.1), memory encoder (§3.3) and contrastive learing (§3.5) are key in our framework compared with existing work of TM-augmented NMT.*

3.1 Contrastive Retrieval

In this stage, following previous work (Gu et al., 2018) we first employ an off-the-shelf full-text search engine, Apache Lucene, to get a preliminary translation memory set $K = \{(x^k, y^k)\}_{k=1}^{|K|}$ $(|K| \gg |M|)$ for every source sentence. Notice that both source sentence x and translation memory set K are from training set $\mathcal{D} = \{(x^n, y^n)\}_{n=1}^{N}$, which means we do not introduce any extra data during training. Then to be directly comparable with previous works (Gu et al., 2018; He et al., 2021) as discussed in Section 5 and considering the core of our method is similarity function-agnostic as detailed below, we adopt a sentence-level similarity function:

$$\mathrm{sim}(x, x') = 1 - \frac{D_{\text{edit}}(x, x')}{\max(|x|, |x'|)} \tag{2}$$

where D_{edit} is the edit distance between two sentences and $|x|$ is the length of x. Specifically, we would select $|M|$ translation memories incrementally and in every step we do not only measure the similarity between current translation memory and the source sentence but also take into consideration the edit distance with those already retrieved ones balanced by a hyperparameter α (namely contrastive factor). Different from MMR (Carbonell and Goldstein, 1998), we treat retrieved translation memories as a whole and take the average similarity score as a penalty term:

$$\underset{x^i \in K \setminus M}{\arg\max} \, [\mathrm{sim}(x, x^i) - \frac{\alpha}{|M|} \sum_{x^j \in M} \mathrm{sim}(x^i, x^j)] \tag{3}$$

where M is the post-ranked translation memory set. Finally, for every source sentence x, by ignoring the source side of M due to information redundancy we have translation memories $M = \{y^m\}_{m=1}^{|M|}$.

3.2 Source Encoder

For a source sentence $x = \{\mathrm{x}_1, ..., \mathrm{x}_s\}$, our *source encoder* is built upon the standard Transformer (Vaswani et al., 2017) architecture composed of a token embedding layer, a sinusoidal positional embedding Layer and stacked transformer encoder layers. Specifically we prepend a <*bos*> token to source sentence and get the dense vector representation z^x as follows:

$$z^x = \mathrm{SrcEnc}(\text{<bos>}, \mathrm{x}_1, ...\mathrm{x}_s) \tag{4}$$

3.3 Memory Encoder

Given a set of translation memories, the local context of each TM and the global context of the whole TM set should be captured by the model to fully utilize this hierarchical structure information. Separately encoding (Gu et al., 2018; Cai et al., 2021) or treating them as a long sequence (Xu et al., 2020)

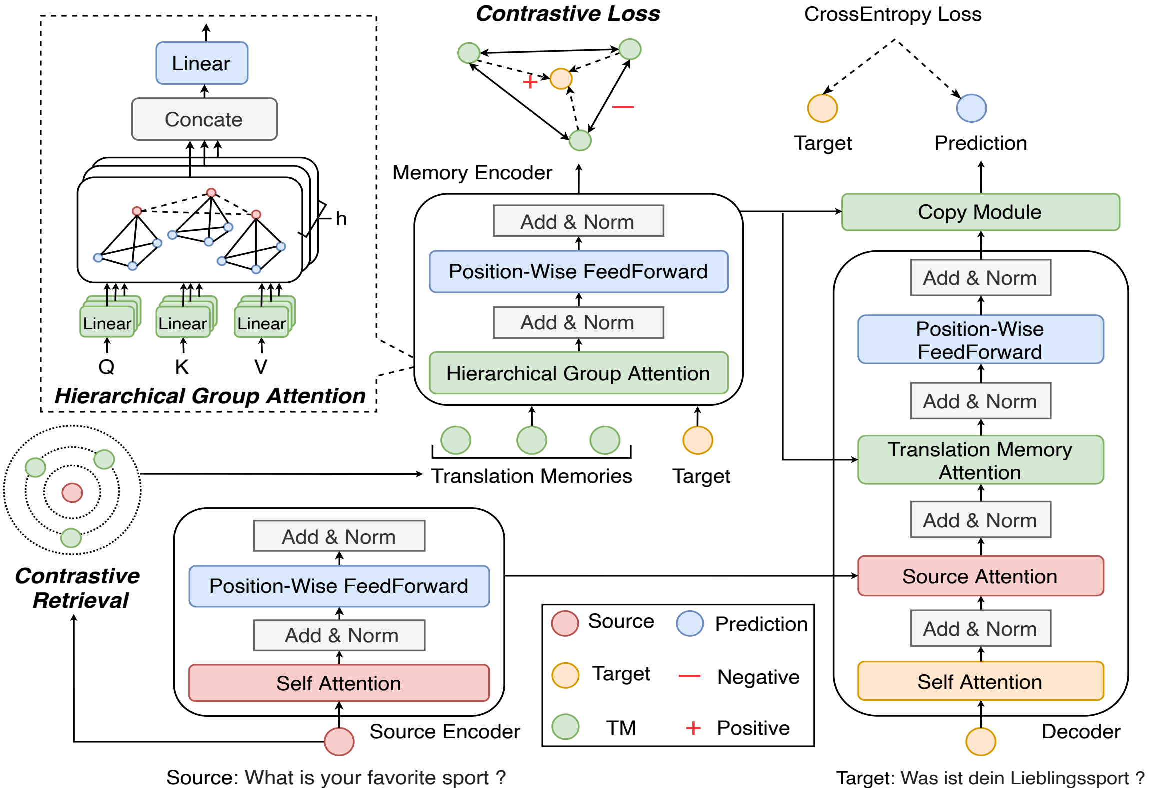

Figure 2: Overview of our framework: (1) ***Contrastive Retrieval*** (2) Source Encoder; (3) Memory Encoder with ***Hierarchical Group Attention*** module (we only show three translation memories for brevity); (4) Decoder; (5) ***Contrastive Learning***.

would inevitably mask the model with this kind of local and global schema. In this section, to facilitate the direct communication between different translation memories for local information and gather the global context via message passing mechanism, we propose a Hierarchical Group Attention (HGA) module. Formally, given a cluster of translation memories $M = \{y^m\}_{m=1}^{|M|}$, where each $y^m = \{y_0^m, ..., y_{n_m}^k\}$ is composed of n_m tokens, for each y^m we would like to create a fully connected graph $G^m = (V^m, E^m)$ where V^m is the token set. To facilitate inter-memory communication, we also create a super node v_*^m by connecting it with all other nodes (namely trivial node) in that graph and then connect all super nodes together contributing to information flow among different translation memories in a hierarchical way as shown in Figure 2. Then we adopt multi-head self attention mechanism (Vaswani et al., 2017) as message passing operator (Gilmer et al., 2017). For every node v_i^m in the graph, their hidden state in time step $t+1$ is updated by the hidden states of its neighbours $\phi(v_i^m)$ in time step t:

$$v_i^m|_{t+1} = \text{SelfAttn}(\phi(v_i^m|_t), v_i^m|_t) \qquad (5)$$

To be computationally efficient, we use mask mechanism to block communication between nodes in different graphs. For each trivial node v_i^m in G^m, they update their hidden states by attending to all trivial nodes as well as super node v_*^m. For v_*^m, it does not only exchange information within the graph G^m, but also communicate with all other super nodes $\{v_*^i\}_{i=1}^{|M|}$. To stabilize training, we also add residual connection and feed-forward Layer after HGA module. After stacking multiple layers, we get dense representation of translation memories:

$$z^m = \text{MemEnc}(\text{Concate}\{y^m\}_{m=1}^{|M|}) \qquad (6)$$

where $|m|$ is the total length of $|M|$ translation memories and $z^m \in \mathbb{R}^{|m| \times d}$.

3.4 Fusing TM in Decoding

To better incorporate the information from both source sentence z^x and translation memories z^m, we introduce a multi-reference decoder architecture. For a target sentence y , we get a hidden representation $h = \{h_1, ..., h_t\}$ after token embedding layer and masked self-attention layer, then we use a cross attention layer to fuse information from source sentence:

$$\hat{h} = \text{CrossAttn}(\text{Add\&Norm}(h), z^x, z^x) \tag{7}$$

Then for translation memories, we employ another cross attention layer:

$$\overline{h} = \text{CrossAttn}(\text{Add\&Norm}(\hat{h}), z^m, z^m) \tag{8}$$

After stacking multiple decoder layers, to further exploit translation memories, we apply a copy module (See et al., 2017; Gu et al., 2016) using the attention score from the second cross attention layer in the last sub-layer of decoder as a probability of directly copying the corresponding token from the translation memory. Formally, with $t-1$ previous generated tokens and hidden state h_t, the decoder computes t-th token probability as:

$$p(y_t|\cdot) = (1 - p_{\text{copy}})p_v(y_t) + p_{\text{copy}} \sum_{i=1}^{|z^m|} \alpha_i \mathbb{1}_{z_i^m = y_t} \tag{9}$$

where $p_{\text{copy}} = \sigma(MLP(h_t, y_{t-1}, \alpha \otimes z^m))$, α is the attention score, $\otimes$ is a Hadamard product and $\mathbb{1}$ is the indicator function.

3.5 Multi-TM Contrastive Learning

The key of contrastive learning is to learn effective representation by pulling semantically close neighbors together and pushing apart non-neighbors (Hadsell et al., 2006; Mikolov et al., 2013). As indicated in (Lee et al., 2020), simply choosing in-batch negatives would yield meaningless negative examples that are already well-discriminated in the embedding space and would even cause performance degradation in translation task (Lee et al., 2020), which also holds true in our preliminary experiments. So how to devise effective contrastive learning objective for a translation model with a cluster of translation memories to learn salient features with respect to the target sentence remains unexplored and challenging.

In this work, to make every translation memory learn distinct and useful feature representations with respect to current target sentence, we propose a novel Multi-TM Contrastive Learning (MTCL) objective which do not simply treat in-batch samples as negative but instead keep aligned with the principle of our *contrastive retrieval*, making every translation memory approach the ground truth translation while pushing apart from each other. Formally, given a source sentence x, its corresponding target sentence y and translation memories $M = \{y^m\}_{m=1}^{|M|}$. The goal of MTCL is to minimize the following loss:

$$\mathcal{L}_{\text{MTCL}} = -\sum_{y^i \in M} log \frac{e^{\text{sim}(y^i,y)/\tau}}{\sum_{y^j \in M} e^{\text{sim}(y^j,y)/\tau}} \tag{10}$$

where $\text{sim}(y^i, y)$ is the cosine similarity between the representation of target sentence y and translation memory y^i given by *memory encoder* and τ is a temperature hyperparameter which controls the difficulties of distinguishing between positive and negative samples (Pan et al., 2021). Notice that the representation of each translation memory is the super node v_*^m given by HGA module in Section 3.3, which communicates with both intra-memory and inter-memory nodes. Intuitively, by maximizing the softmax term $e^{\text{sim}(y^i,y)/\tau}$, the contrastive loss would force the representation of each translation memories to approach the ground truth while push apart from each other, delivering a uniformly distributed representation around the target sentence in latent semantic space. In MTCL, all negative samples are not from in-batch data but are different translation memories for one source sentence, which make up of non-trivial negative samples and help the model to learn the subtle difference between multiple translation memories.

During the training phase, the model can be optimized by jointly minimizing the MTCL loss and Cross Entropy loss as shown:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \mathcal{L}_{\text{MTCL}} \tag{11}$$

where λ is a balancing coefficient to measure the importance of different objectives in a multi-task learning scenario (Sener and Koltun, 2018).

4 Experimental Setup

4.1 Dataset and Evaluation

We use the JRC-Acquis (Steinberger et al., 2006) corpus to evaluate our model. This corpus is a collection of parallel legislative text of European union Law applicable in the EU member states. Highly related and well structured data make this corpus an ideal test bed to evaluate the proposed TM-augmented translation model. Following previous work, we use the same split of train/dev/test set as in (Gu et al., 2018; Xia et al., 2019; Cai et al., 2021; Xu et al., 2020; He et al., 2021). For evaluation, we use *SacreBLEU*.

4.2 Implementation Details

Our model is named Contrastive Memory Model (CMM). To implement CMM, we use transformer as building block of our model. Specifically, we adopt the base configuration and the default optimization configuration as in Vaswani et al. (2017). We use joint BPE encoding (Sennrich et al., 2016) with vocab size 35000. We also adopt label smoothing as 0.1 in all experiments. The number of tokens in every batch is 10000, which includes both source sentence and translation memories. The memory size and contrastive factor is set to be 5 and 0.7 across all translation directions. The contrastive temperature τ is $\{0.1, 0.08, 0.05, 0.15\}$ for Es→En, En→Es, De→En and En→De directions. The balancing factor λ is set to be 1 [1].

4.3 Baselines

CMM is compared with the following baselines:

- Vaswani et al. (2017): this is the original implementation of base transformer.
- Gu et al. (2018): this is a pioneer work of integrating translation memories into NMT system using an external memories networks to separately encode every translation memory
- Xu et al. (2020): this paper augments source sentence with concatenation of TM and euqip the model with different language embedding (FM^{+}).
- Xia et al. (2019): this work uses a compact graph to encode translation memories and is also based on transformer architecture.
- Zhang et al. (2018): this work equips a NMT model with translation pieces and extra bonus given to outputs that contain the collected translation pieces.
- Cai et al. (2021): this model first retrieves translation memories by source side similarity and adopts a dual encoder architecture.
- He et al. (2021): this model incorporates one most similar translation memory with proposed example layer.

In addition, considering that Gu et al. (2018) is based on Memory Network and RNN architecture, to be fairly compared with transformer based model, we re-implement two more direct baselines (i.e., **BaseGreedy** and **T-Ada**) on top of Transformer with the same configuration as our CMM. Specifically, in both baselines the original Memory Network is replaced by a transformer encoder sharing weights with source encoder. BaseGreedy employs greedy retrieval and it does not take diversity of TM into account. In contrast, T-Ada adopts adaptive retrieval, which finds the translation memories via maximizing the token coverage of source sentence, and it promotes the diversity in retrieved memory to some extent as CMM.

		CMM	T-Ada	BaseGreedy
Avg. TM Size		5	5.68	5
Avg. Coverage		84.01%	92.11 %	81.13%
Avg. Similarity		0.89	0.84	0.91
Training Latency		1.21x	1.25x	1.21x
Inference Latency		1.44x	1.56x	1.44x
BLEU	Es→En	**67.76**†	67.08	66.84
	En→ES	**64.04**†	63.56	63.18
	De→En	**64.33**†	63.81	63.84
	En→De	**58.69**†	57.28	57.02

Table 1: Comparison between CMM, T-Ada and BaseGreedy. The TM Size, Coverage and Similarity is averaged among four translation directions. Coverage means the token level coverage of all translation memories with respect to source sentence. Similarity score is calculated as described in Section 3.1. † means CMM is significantly better than baselines with *p-value* < 0.01.

5 Experiment Results

5.1 Main results

Is diverse translation memory helpful? We make a comparison with the direct baseline T-Ada because both the proposed CMM and T-Ada promote the diversity in translation memory. As shown in Table 1, T-Ada yields modest gains (about +0.2 BLEU points on average) over BaseGreedy on four translation tasks, which is in line with the results in Gu et al. (2018) on the RNN architecture. We conjecture that it is because *Adaptive Retrieval* only partially maximize the word coverage while neglecting the overall semantics of the whole sentence thus injecting undesirable noise into the retrieval phase. In contrast, the proposed CMM takes both token-level coverage and sentence-level similarity into consideration and consistently outperforms T-Ada, gaining about 0.5-1.4 BLEU points on four tasks in translation quality with smaller TM size and lower latency in both training and inference phase. This fact shows the following findings: 1) NMT augmented with diverse translation memory can yield consistent improvements in translation quality; 2) how to model and learn the diverse translation memory is important in addi-

[1] Code and data is available at https://github.com/Hannibal046/NMT_with_contrastive_memories

System	Es→En		En→Es		De→En		En→De	
	Dev	Test	Dev	Test	Dev	Test	Dev	Test
Vaswani et al., 2017†	64.08	64.63	62.02	61.80	60.18	60.16	54.65	55.07
Gu et al., 2018	57.62	57.27	60.28	59.34	55.63	55.33	49.26	48.80
Zhang et al., 2018	63.97	64.30	61.50	61.56	60.10	60.26	55.54	55.14
Xu et al., 2020*	66.44	65.90	-	-	-	-	-	-
Xia et al., 2019	66.37	66.21	62.50	62.76	61.85	61.72	57.43	56.88
He et al., 2021(@s)	67.23	67.26	-	-	-	-	-	-
Cai et al., 2021(#2)	66.98	66.48	63.04	62.76	63.62	63.85	57.88	57.53
CMM	**67.48**	**67.76**	**63.84**	**64.04**	**64.22**	**64.33**	**58.94**	**58.69**

Table 2: BLEU points on four translation directions of JRC-Acquis dataset. † denotes that the model is implemented by ourselves. @s means the model is trained under standard training criterion and * means results are from He et al. (2021). #2 is the second model proposed in Cai et al. (2021) using source retrieval.

System	Model Size	Training	Inference	Es→En		En→Es		De→En		En→De	
				Dev	Test	Dev	Test	Dev	Test	Dev	Test
T-Para	101M	2.76x	1.36x	**67.73**	67.42	**64.18**	63.86	**64.48**	**64.62**	58.77	58.42
CMM	68M	1.21x	1.44x	67.48	**67.76**	63.84	**64.04**	64.22	64.33	**58.94**	**58.69**

Table 3: Translation quality and running efficiency compared with the strong model T-Para.

	BLEU	Chrf	TER	BertScore	BartScore
BaseGreedy	66.84	78.45	25.39	0.9686	0.1209
CMM	**67.76**	**79.01**	**24.43**	**0.9698**	**0.1329**

Table 4: Evaluation results with different metrics.

	top-25%	top-50%	top-75%	ALL
Vaswani et al. (2017)	29.17	43.48	56.07	64.63
BaseGreedy	34.17	48.77	59.94	66.84
CMM	**35.38**	**49.37**	**60.53**	**67.76**

Table 5: Evaluation results in terms of BLEU in different difficulty range.

tion to promoting diversity in translation memory. Because of the potential problem of high BLEU test (Callison-Burch et al., 2006), we conduct another two experiments. First, We use metrics other than BLEU to evaluate our high BLEU systems. We compare our model CMM and BaseGreedy in JRC/EsEn dataset. We use both model-free and model-based metrics as shown in Table 4. A clear patent here is that our higher-BLEU model CMM outperforms BaseGreedy model in all these metrics. Second, we disengage our model from high BLEU range by picking the hard sentences from the test set of JRC/EsEn according to the sentence-level BLEU for a vanilla Transformer model. The evaluation results for top-25%, top-50%, top-75% hardest subsets are shown in Table 5. We can see that the proposed CMM still outperforms baselines on the top-25% subset whose BLEU is in the range of 30s.

Comparing with other baselines Since our CMM involves the heuristic metric (i.e., TF-IDF and normalized edit distance) for retrieval, we first compare our methods with other works using the same retrieval metric. The result is presented in Table 2. As can be seen, our method yields consistent better results than all other baseline models across four tasks in terms of BLEU. Substantial improvement by an average 3.31 BLEU points and up to 4.29 in En→De direction compared with transformer baseline model demonstrates the effectiveness of incorporating translation memories into NMT model. In comparison with previous works either using greedy retrieval (Gu et al., 2018; Zhang et al., 2018; Xia et al., 2019; Cai et al., 2021), which introduces redundant and uninformative translation memories, or using top1 similar translation memory (Xu et al., 2020; He et al., 2021), which causes omission of potentially useful cues, our framework equipped with contrastive translation memories can deliver consistent improvement in both development set and test set among four translation directions.

Unlike the above work, there is also another line of work that retrieve translation memory with a learnable metric. Cai et al. (2021) proposes a

		En-De		En-Es	
		→	←	→	←
Dev	T w/o MTCL	58.55	64.14	63.26	67.30
	T w/o HGA	58.06	63.85	62.74	67.28
	T-Greedy	58.01	63.72	63.10	66.98
	T-MMR	58.20	64.10	62.66	67.25
	CMM	**58.94**	**64.22**	**63.84**	**67.48**
Test	T w/o MTCL	58.37	64.29	63.92	67.49
	T w/o HGA	58.06	64.19	62.74	67.28
	T-Greedy	57.66	63.57	63.28	67.16
	T-MMR	57.95	64.27	63.10	67.15
	CMM	**58.69**	**64.33**	**64.04**	**67.76**

Table 6: Ablation study in four translation tasks with respect to each key component in our framework.

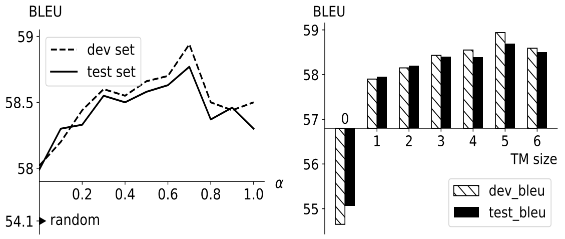

Figure 3: Effect of contrastive factor and TM size.

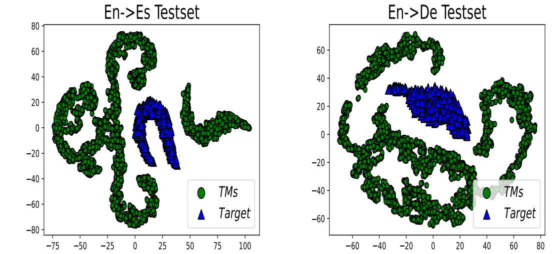

Figure 4: Visualization of Translation Memories and Target sentence in EnEs and EnDe testset by t-SNE.

powerful framework (namely T-Para) which jointly trains the retrieval metric and translation model in an end-to-end fashion, leading to state-of-the-art performance in translation quality. We also compare our method with this strong model and result is shown in Table 3. Notice that our model gives comparable results with T-Para, which is actually remarkable considering that our model has much smaller model size and training latency. In particular, our work about contrastive translation memory is orthogonal to Cai et al. (2021) and it is promising to apply our idea into their framework, which remains a future work.

5.2 Analysis

Ablation Study We also implement several variants of our framework: (1) T w/o MTCL: this model uses the same model configuration as CMM but without MTCL loss. (2) T w/o HGA: in this setting, $|M|$ translation memories are concatenated together as a long sequence without Hierarchical Group Attention module. (3) T-Greedy: this model replaces the *Contrastive Retrieval* in CMM by *Greedy Retrieval*. (4) T-MMR: this model replaces the *Contrastive Retrieval* in CMM by Maximal Marginal Relevance (Carbonell and Goldstein, 1998) while the setting of translation model keeps the same as CMM. The result is shown in Table 6 and we have the following observations. Simply replacing *Contrastive Retrieval* by *Greedy Retrieval* or MMR while keeping the setting of translation model unchanged yields worse results than our model which demonstrates that the informative translation memories serve as key ingredient in a TM-augmented NMT model. Interestingly, direct removal of HGA module while maintaining MTCL objective (i.e., T w/o HGA) gives consistent worse results in four translation directions. We suspect that a pull-and-push game brought by contrastive learning causes performance degradation without modeling the fine-grained interaction among multiple translation memories. Combining HGA and MTCL, which facilitates communication between different translation memories and helps the model to learn the subtle difference between them, performs better than all other baseline models revealing the fact that properly designed contrastive learning objective and HGA module is complementary to each other.

Memory Size and Contrastive Factor To verify the effectiveness of fusing multiple contrastive translation memories, we choose En→De dataset and make the following experiments in both TM retrieval and TM fusion stage: In retrieval stage, we explore the contrastive factor α which is supposed to decide the degree of currently retrieved translation memory contrasting to those already retrieved. A larger α indicates that the retrieved translation memories are less similar to the source sentence while more contrastive to each other. And in fusion stage, the size $|M|$ of translation memories is considered. The effect of different α is shown in Figure 3. The *random* point is the result of a NMT model with $|M|$ randomly retrieved translation memories and it even underperforms a non-TM translation model (Vaswani et al., 2017) shown in Table 2. We assume it is due to much noise injected by random memories. When contrastive factor α is set to 0, it is essentially greedy retrieval, and an important observation is that the translation quality of our model increases with the α until it drops at some certain point. We suspect that too large α would yield mutually contrastive TM that divert too much from the original source sentence. Similar phenomenon can be verified in the Figure 3, when TM size equals to 0, it is a non-TM translation model delivering worst result while too large TM size also hurts the model performance which is also observed in Bulté and Tezcan (2019); Xia et al. (2019).

To further demonstrate the intuition behind our framework, we randomly sample 1,000 examples from test sets of En→De and En→Es directions and use t-SNE (Van der Maaten and Hinton, 2008) to visualize the sentence embedding of translation memories and target sentence encoded by our CMM. The result is shown in Figure 4 and one interesting observation is that although the target side of testset is never exposed to the model, the representation of translation memories are uniformly distributed around the target sentence in the latent semantic space.

6 Conclusion

In this work, we introduce an approach to incorporate contrastive translation memories into a NMT system. Our system demonstrates its superiority in retrieval, memory encoding and training phases. Experimental results on four translation datasets verify the effectiveness of our framework. In the future, we plan to exploit the potential of this general idea in different retrieval-generation tasks.

7 Limitations

This paper propose a framework for Retrieval-augmented Neural Machine Translation and it relies on holistically similar but mutually contrastive translation memories which makes it work mostly for corpora in the same domain. How to apply this general idea to other scenario like low resource NMT remains a future challenge.

8 Acknowledgement

This work was supported by National Natural Science Foundation of China (NSFC Grant No. 62122089 and No. 61876196). Rui Yan is supported by Tencent Collaborative Research Fund.

References

Bram Bulté and Arda Tezcan. 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In *57th Annual Meeting of the Association-for-Computational-Linguistics (ACL)*, pages 1800–1809.

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.

Chris Callison-Burch, Miles Osborne, and Philipp Koehn. 2006. Re-evaluating the role of Bleu in machine translation research. In *11th Conference of the European Chapter of the Association for Computational Linguistics*, pages 249–256, Trento, Italy. Association for Computational Linguistics.

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.

Jaime Carbonell and Jade Goldstein. 1998. The use of mmr, diversity-based reranking for reordering documents and producing summaries. In *Proceedings of the 21st annual international ACM SIGIR conference on Research and development in information retrieval*, pages 335–336.

Ting Chen, Simon Kornblith, Mohammad Norouzi, and Geoffrey Hinton. 2020. A simple framework for contrastive learning of visual representations. In *International conference on machine learning*, pages 1597–1607. PMLR.

Tina Paulsen Christensen and Anne Schjoldager. 2010. Translation-memory (tm) research: what do we know and how do we know it? *Hermes-Journal of Language and Communication in Business*, (44):89–101.

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.

Tianyu Gao, Xingcheng Yao, and Danqi Chen. 2021. Simcse: Simple contrastive learning of sentence embeddings. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6894–6910.

Justin Gilmer, Samuel S Schoenholz, Patrick F Riley, Oriol Vinyals, and George E Dahl. 2017. Neural message passing for quantum chemistry. In *International conference on machine learning*, pages 1263–1272. PMLR.

Jiatao Gu, Zhengdong Lu, Hang Li, and Victor OK 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 *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Raia Hadsell, Sumit Chopra, and Yann LeCun. 2006. Dimensionality reduction by learning an invariant mapping. In *2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'06)*, volume 2, pages 1735–1742. IEEE.

Kaiming He, Haoqi Fan, Yuxin Wu, Saining Xie, and Ross Girshick. 2020. Momentum contrast for unsupervised visual representation learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9729–9738.

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.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Seanie Lee, Dong Bok Lee, and Sung Ju Hwang. 2020. Contrastive learning with adversarial perturbations for conditional text generation. In *International Conference on Learning Representations*.

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

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.

Yixin Liu and Pengfei Liu. 2021. Simcls: A simple framework for contrastive learning of abstractive summarization. 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 1065–1072.

Tomas Mikolov, Ilya Sutskever, Kai Chen, Greg S Corrado, and Jeff Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in neural information processing systems*, pages 3111–3119.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural 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 1: Long Papers)*, pages 244–258.

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.

Ozan Sener and Vladlen Koltun. 2018. Multi-task learning as multi-objective optimization. *Advances in neural information processing systems*, 31.

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

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

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*.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. In *Advances in neural information processing systems*, pages 3104–3112.

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

Laurens Van der Maaten and Geoffrey Hinton. 2008. Visualizing data using t-sne. *Journal of machine learning research*, 9(11).

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.

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 M 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 NAACL-HLT*, pages 1325–1335.