Exploiting Target Language Data for Neural Machine Translation Beyond Back Translation

Anonymous ACL submission

Abstract

Neural Machine Translation (NMT) suffers from the challenges of translating in new domains and low-resource languages. To address these challenges, researchers have proposed methods to incorporate additional knowledge into NMT, including the integration of translation memories (TMs). However, finding TMs that closely match the input sentence remains difficult, particularly for specific domains. In contrast, monolingual data is widely available in most languages and back-translation is believed as a promising method to utilize target language data. But, it still needs additional training. In this paper, we propose Pseudo-kNN-MT, a method that exploit target language data during the inference phase, without training the NMT model. Also, we further investigate the assistance of large language model (LLM) in NMT. Experimental results show that our method can improve translation quality by a great margin. Interestingly, LLMs are found to be helpful for strong NMT systems.

1 Introduction

Neural Machine Translation (NMT) has witnessed significant advancements with the introduction of deep learning techniques(Sutskever et al., 2014; Bahdanau et al., 2015), especially the transformer model(Vaswani et al., 2017). Despite these advancements, challenges still exist in translating rare words and adapting NMT systems to new domains(Koehn and Knowles, 2017; Saunders, 2022).

To address these challenges, researchers have proposed various methodologies to incorporate external knowledge into NMT. One such approach involves imposing constraints from terminology dictionaries(Dougal and Lonsdale, 2020; Hasler et al., 2018), or the incorporating fuzzy matches retrieved from translation memory (TM)(Eriguchi et al., 2019; Xu et al., 2020; Khandelwal et al., 2021; He et al., 2021; Reheman et al., 2023).

These techniques enhance the NMT systems using bilingual translation knowledge. However, due to the limitations of the bilingual data scale and coverage of domains, it is highly challenging to find sentences that are highly similar to the input sentence, particularly in specific domains or in low-resource languages. One natural idea is utilizing the vast amount of monolingual data, which can offer a pool of highly relevant sentences in terms of semantics. As a promising method, back translation (Sennrich et al., 2016) has been proven to be useful for NMT, especially in low-resource scenarios. However, it still needs some additional training, including the training of a reverse NMT system and retraining the NMT model with the augmented training data.

In this paper, we propose pseudo-kNN-MT, a training-free method that utilizes target language data for translation. For an input sentence, we retrieve its similar target sentences using a cross-lingual retriever. our main goal is how to use these retrieved sentences effectively. First of all, we pair the retrievals with the input sentence and construct pseudo sentence pairs, then conduct nearest neighbor machine translation following Khandelwal et al. (2021). Besides, LLMs are believed to be good compressors (Brown et al., 2020; Radford et al., 2019), which can map texts into representation space effectively, and also demonstrate strong translation capabilities (Zhang et al., 2023; Zhu et al., 2023a; Xu et al., 2023). Consequently, We also investigate the effectiveness of kNN translation model interpolation, whose datastore and the context representation vector are obtained by LLMs, rather than by the NMT model itself. Further, we study the integration of LLMs with the NMT without referring to the target retrieval, from the perspectives of the utilization of the translation ability of LLMs and fluency of the target translation. Experimental results on multi-domain test sets show that our method improves the translation results

with a great margin, with 4.51 sacreBLEU points on average, utilizing the target language data only.

2 Background

In this section, we give some background knowledge about nearest neighbor machine translation, cross-lingual retrieval and the fusion of language models in NMT.

2.1 Nearest Neighbor Machine Translation

The k-Nearest Neighbor Machine Translation (kNN-MT) Khandelwal et al. (2021) is a non-parametric method that uses nearest neighbor retrievals from a vector datastore of translation context representation. It involves two main steps: datastore creation and inference.

Datastore Creation The datastore $\mathcal{D}$ consist of a set of key-value pairs, and the key is the high-dimensional representation of the translation context, which is computed by an auto-regressive MT decoder, and the value is the corresponding ground-truth target token. Here, the combination of source language tokens and the generated target tokens is called translation context. Suppose $(\mathcal{X}, \mathcal{Y})$ is a set of bilingual sentences, and $f(\cdot)$ is a mapping function that transfers the translation context into the high-dimensional representation, using a translation model. For all examples in $(\mathcal{X}, \mathcal{Y})$, the key-value datastore is created as:

$$\begin{aligned}\mathcal{D} \quad = \quad & \{(f(x, y_{1:t-1}), y_t), \forall y_t \in y | \\ & (x, y) \in (\mathcal{X}, \mathcal{Y})\} \end{aligned} \tag{1}$$

Inference During the inference, the translation context representation of each time-step is used as query, $q = f(x, \hat{y}_{1:t-1})$, to retrieve k-nearest neighbors $\mathcal{N}$ from $\mathcal{D}$, using vector distance measuring methods, such as $L2$ distance. A probability distribution, $p_{k\text{NN}}$, over the target vocabulary is then constructed from $\mathcal{N}$ by applying a softmax with temperature to the negative distances and aggregating the same tokens, defined as:

$$\begin{aligned} & p_{k\text{NN}}(y_t | x, \hat{y}_{1:t-1}) = \\ & \frac{\sum_{(k_j, v_j) \in \mathcal{N}} \mathbb{1}_{y_j = v_j} exp(-d(q, k_j)/T)}{\sum_{(k_j, v_j) \in \mathcal{N}} exp(-d(q, k_j)/T)} \end{aligned} \tag{2}$$

where $d(\cdot, \cdot)$ is a distance function that calculates the distance between the two vectors. Here, it is the distance between the query vector and the retrieved neighbors.

In the end, the final probability distribution is obtained by linear interpolating the two distributions, $p_{k\text{NN}}$ and p_{NMT}, using a tuned hyperparameter λ:

$$\begin{aligned} p(y_t | x, \hat{y}_{1:t-1}) = & \lambda p_{k\text{NN}}(y_t | x, \hat{y}_{1:t-1}) + \\ & (1 - \lambda) p_{\text{NMT}}(y_t | x, \hat{y}_{1:t-1}) \end{aligned} \tag{3}$$

2.2 Cross-lingual Retrieval

Cross-lingual retrieval is the technique of retrieving information from multilingual sources (Feng et al., 2022a; Li et al., 2023; Gao et al., 2023). Its core is a pretrained cross-lingual sentence embedding model, which maps the sentences from different languages into a shared semantic space. In usage, they return the embedding of "CLS" tokens or using mean pooling strategy for all token embeddings in the sentence to acquire the representation of a sentence. This is useful in various cross-lingual applications, such as information retrieval and machine translation (MT). We use it to retrieve similar sentences from the target language, taking the source sentence as query, in this paper.

3 Methodology

In this section, we will introduce retrieving similar sentences from target dataset (§3.1), as well as the proposed method of pseudo kNN-MT (§3.2) and the large language model integration (§3.3 and §3.4) in detail.

3.1 Retrieving Similar Sentences from Target Language Dataset

Given a source input sentence x, a target language dataset $\mathcal{Y} = \{y^1, y^2, ..., y^n\}$, and a cross-lingual sentence embedding model e. First of all, the distributed representation of the target dataset, $h_{\mathcal{Y}} = \{h_1, h_2, ..., h_n\}$, is obtained by feeding $\mathcal{Y}$ into the cross-lingual model, formulated as:

$$h_{\mathcal{Y}} = e(\mathcal{Y}) \tag{4}$$

In the same way, we acquire the distributed representation of x by $h_x = e(x)$. After that, we calculate the distances of each item in $h_{\mathcal{Y}}$ from h_x by the distance function $d(\cdot)$:

$$D = d(h_x, h_{\mathcal{Y}}) \tag{5}$$

where $D = \{d_1, d_2, ..., d_n\}$ is the distance of each sentence in y from x in the vector space. Finally, we acquire the top-k similar sentences by ranking the sentences by their distances and select k-nearest of them as the final retrieval, namely the k-nearest neighbors. Our work is primarily centered on the utilization of this target retrievals.

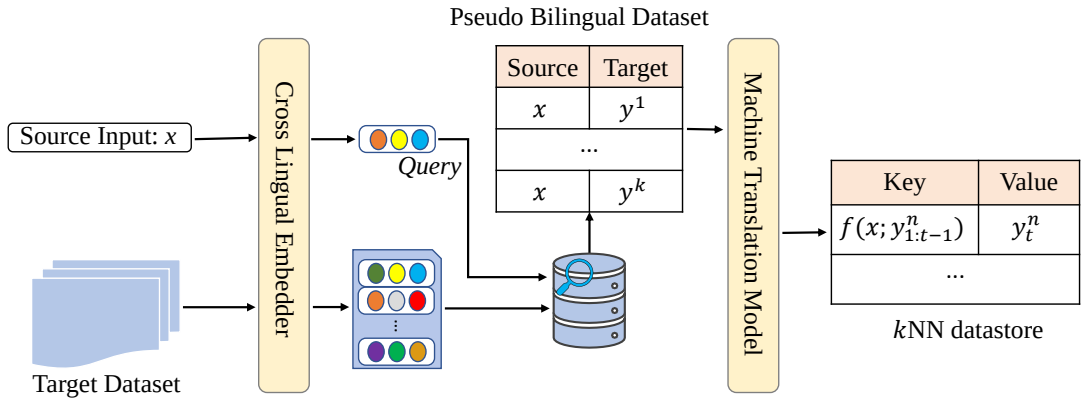

Figure 1: pseudo datastore creation process. Function $f(\cdot)$ returns the last hidden state of MT decoder at every time-step.

3.2 kNN-MT with Pseudo Datastore

After obtaining similar sentences from the target dataset, we endeavor to construct bilingual data in order to align with the decoding behavior of the MT model. A well-known method is to back-translate them into the source language, and pair the corresponding sentences (Sennrich et al., 2016). But, this needs to train an additional reverse NMT model, which is trained to translate from target to source. Here, we take another option. Due to the semantic similarity between the retrieved sentences with the input sentence, we pair them and construct bilingual data. After this, we explore whether this pseudo bilingual data can effectively facilitate the translation, following the approach of kNN-MT (Khandelwal et al., 2021).

First, we build a key-value datastore $\mathcal{D}_{\text{pse}}$ on the pseudo bilingual data. Suppose $\mathcal{Y}_{\text{sim}} = \{y^1, y^2, ..., y^k\}$ is the target retrieval for the input sentence x, the pseudo bilingual data is constructed by pairing x with each sentence in $\mathcal{Y}_{\text{sim}}$, as $(\mathcal{X}, \mathcal{Y})_{\text{pse}} = \{(x, y_i) | y_i \in \mathcal{Y}_{\text{sim}}, i \in [1, k]\}$. The kNN-MT datastore on $(\mathcal{X}, \mathcal{Y})_{\text{pse}}$ is built using the equation 1, defined as:

$$\mathcal{D}_{\text{pse}} = \{(f(x, y_{1:t-1}), y_t), \forall y_t \in y | (x, y) \in (\mathcal{X}, \mathcal{Y})_{\text{pse}}\} \quad (6)$$

where $f(\cdot)$ also is the mapping function from translation context to last hidden state of the MT decoder and t is the time-step of decoding. Figure 1 shows the pseudo datastore creation process.

During the inference, we construct the target token distribution from $\mathcal{D}_{\text{pse}}$ and interpolate it with the NMT distribution, using equation 2 and equation 3, same as kNN-MT (Khandelwal et al., 2021).

3.3 kNN-MT with LLM Pseudo Datastore

As a dual-model method, kNN-MT is the combination of NMT model and the kNN translation model from datastore. Unlike the naive implementation that uses NMT model's own hidden states to construct key-value datastore and retrieve during inference, any MT model can be used to finish this procedure. In addition, previous works reported that LLMs has the distinct translation and multilingual modeling capability(Zhang et al., 2023; Zhu et al., 2023a; Xu et al., 2023). Here, we explore marrying NMT model with the kNN translation model whose datastore is constructed by an LLM.

Due to the differences between utilizing LLMs for translation tasks and NMT, where LLMs require instructions to specify the desired translation task, including the support for zero-shot and few-shot learning, the translation context here differs from that in NMT. Taking the zero-shot scenario of De-En translation task as an example, presuming x to be the source sentence, and $y_{1:t-1}$ representing the target string of previously generated tokens. The translation context for LLM can be written as "Translate this from Geman into English.\nGerman: <x>\nEnglish: <$y_{1:t-1}$>". In few-shot scenario, few-shot examples come after the instruction.

First of all, we construct the key-value datastore on the pseudo bilingual dataset obtained previously, using an LLM. For all the bilingual sentences, we feed them into the LLM using specific prompt and extract the hidden states of the translation context at each time step as the key and the corresponding target token as value. For zero-shot, we use the prompt as "Translate this from *source language* into *target language*.\n*source language*: <*source sentence*>\n*target language*:". It is worth mention-

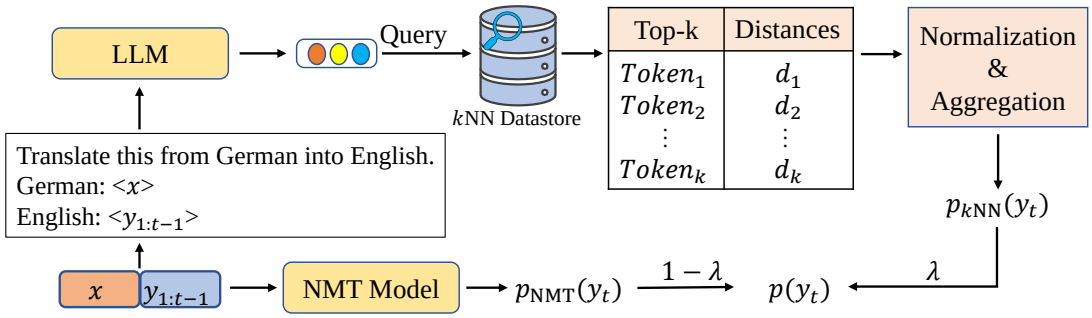

Figure 2: Illustration of decoding using LLM datastore. Here, we take zero shot prompt as an example. The kNN datastore is constructed offline using the LLM on the pseudo bilingual dataset.

ing that when constructing the key-value datastore from the pseudo bilingual dataset, we should use the same prompt that was used during the inference to maintain key representation consistency.

At each time step of inference phase, we first construct the translation context using the prompt. The translation context is then fed into the LLM to extract the hidden state. Then, we take this hidden state as a query to search the k-nearest neighbors from the datastore, and get the kNN probability, which is interpolated with NMT probability afterward. The illustration of the inference phase is given in Figure 2.

3.4 Large Language Model Integration

Being a key component of statistical machine translation (SMT), a language model trained on target language data is used to predict sentence probability. while NMT models the translation task in end-to-end way, and no need to train a language model explicitly. A target language model incorporation in the inference can not help much to the NMT (Gülçehre et al., 2015).

But, in LLMs, things changed a lot. They not only possess capabilities beyond merely generating continuations based on prefix text but also can process multilingual information according to human instructions, such as translating. With this knowledge, we further explore LLM integration without additional data. For an input sentence x and previously generated target tokens $y_{1:t-1}$, our method operates as follows.

LLM Translator Interpolation In this method, we leverage the translation capabilities of the LLM. At each time-step of inference, we utilize both x and $y_{1:t-1}$ to construct the prompt for the LLM. The prompt here is the same as the translation context for LLM described in Section 3.3. Subsequently, the constructed prompt is fed into the LLM, which in turn generates its probability distribution for y_t. Finally, we combine the LLM probability p_{LLM} and the NMT probability p_{NMT} through interpolation using a hyperparameter λ:

$$p(y_t|x, \hat{y}_{1:t-1}) = \lambda p_{\text{NMT}}(y_t|x, \hat{y}_{1:t-1}) + (1-\lambda) p_{\text{LLM}}(y_t|x, \hat{y}_{1:t-1}, pr) \quad (7)$$

where pr represents the prompt template.

LLM Continuation Generator Fusion In this method, we leverage the continue generation capabilities of the LLM, without referencing the source language information That is to say, next token generation is only conditioned on $y_{1:t-1}$. At each time step of inference, we feed $y_{1:t-1}$ into the LLM and get its next token generation probability p_{LLM}. The final translation probability for y_t is calculated by adding this probability with the generation probability of NMT, p_{NMT}, using a hyperparameter λ as:

$$p(y_i|x, \hat{y}_{1:t-1}) = p_{\text{NMT}}(y_t|x, \hat{y}_{1:t-1}) + \lambda p_{\text{LLM}}(y_t|y_i|\hat{y}_{1:t-1}) \quad (8)$$

4 Experiments

In this section, we will introduce our experiments, including the main experiment and the comprehensive analysis from various perspectives.

Datasets and Evaluation Metrics We evaluated the effectiveness of our proposed method on publicly available datasets. For domain adaptation, we performed the experiments on IT, Koran, Law, and Medical domains of multi-domain datasets provided by Aharoni and Goldberg (2020). To measure the translation quality, we used sarcreBLEU (Post, 2018) and COMET (Rei et al., 2022). The data statistics are given in table 1.

Split	Multi-domain				WMT19
	IT	Koran	Law	Medical	
Train	223K	17K	467K	248K	33M
Valid	2000	2000	2000	2000	6002
Test	2000	2000	2000	2000	2000

Table 1: Statistics of datasets.

Models We used the winner model of the WMT19 De-En news translation task, submitted by Facebook, as the pretrained base NMT model (Ng et al., 2019). For LLM, we use various versions of LLAMA 2 (Touvron et al., 2023), including the base version LLAMA-2-7B, dialogue optimized version LLAMA-2-7B-chat and ALMA-7B (Advanced Language Model-based trAnslator), a translation optimized model from LLAMA-2-7B, from Xu et al. (2023), respectively. We encountered difficulties when integrating the NMT model with Llama 2. The native version of wmt19 cannot be assisted by LLM directly, because they used different tokenization strategiesword granularity, and different training data, which results in the difference in the dictionary of the two models. So, we trained another NMT model on WMT19 De-En training data, using the dictionary of Llama 2. Also, we trained a decoder-only transformer language model (Radford et al., 2019) with 12 layers and a model dimension of 768 on the target data of the WMT19 De-En dataset and Llama 2 dictionary as well, ensuring a fair comparison. Before training, We cleaned WMT19 training data by applying punctuation normalization and language identification filtering. After that, we tokenized them using `llama.tokenizer`.

Settings We use the cross-lingual embedding model LaBSE (Feng et al., 2022b) to transfer both the source and target datasets into the embedding representations, then we use dense vector similarity search library FAISS (Johnson et al., 2021) to perform cross-lingual retrieval. For k-nearest neighbor searching from the kNN datastore, we also use FAISS. In all experiments, for retrieving top-k similar sentences from the target dataset, we set this k' to 32. For models that perform retrieval, we retrieve $k = 8$ neighbors from the translation context vector datastore. For the kNN temperature, we followed the optimized settings from Zheng et al. (2021), and set it to 100 for Koran, and 10 for other domains. Except for the kNN-MT method that used in LLM datastore, which searches the interpolation hyperparameter from $\lambda \in \{0.2, 0.3, 0.4\}$, other methods searches from $\lambda \in \{0.1, 0.2, 0.3, 0.4, 0.5, 0.6, 0.7, 0.8, 0.9\}$. For decoding, we set the $beam\ size$ to 5, and $length\ penalty$ to 1.0.

We take vanilla NMT (Base-NMT) and vanilla kNN-MT (kNN-MT) as the baselines. For simulating the usage of monolingual data, we take the target language training data as the monolingual dataset. The other compared methods are as follows:

Pseudo-kNN-MT: the method that introduced in Section 3.2.

Mono-bt-kNN-MT: a kNN-MT method. Its datastore is created from a bilingual dataset whose source sentences are obtained by translating the target dataset back into source language by WMT19 En-De model (Ng et al., 2019).

Retrieve-bt-kNN-MT: a variant of Pseudo-kNN-MT. In this method, the retrieved target sentences are back translated into the source language sentences by WMT19 en-de model (Ng et al., 2019), then constructed bilingual sentence pairs, from which the datastore is constructed afterward.

4.1 Main Experiment

In this experiment, we tested our method on the testset of the multi-domain dataset. The base model for NMT is Facebook's WMT19 De-En model. For back translation, we employed Facebook's WMT19 En-De model. The experimental results with sacreBLEU scores are given in Table 2. We give the COMET scores for this experiment in the Appendices B. In this experiment, we compare our proposed method with the vanilla kNN-MT (Khandelwal et al., 2021).

The experimental results indicate that although the performance is not as good as the vanilla KNN-MT, our proposed Pseudo-KNN-MT method can improve BLEU scores by an average of 4.51 BLEU points compared to the baseline. This seems reasonable intuitively because the pseudo-bilingual sentences are constructed by pairing the retrieved target sentences with the source ones, they are similar or relevant in semantics, but not the exact match to each other. However, the datastore of kNN-Mt is constructed from fully aligned bilingual data. To address this, we use the reverse model to translate the retrieved target language data back into the source language and build the bilingual data. This approach further boosts BLEU scores by 0.75

Methods	IT	Koran	Law	Medical	Average
NMT	38.43	17.07	45.99	41.97	35.86
kNN-MT	$46.74_{(0.7)}$	$21.93_{(0.7)}$	$61.92_{(0.9)}$	$56.40_{(0.8)}$	46.75
Pseudo-kNN-MT	$40.63_{(0.3)}$	$18.46_{(0.4)}$	$53.03_{(0.4)}$	$49.36_{(0.5)}$	40.37
Retrieve-bt-kNN-MT	$41.53_{(0.8)}$	$19.44_{(0.8)}$	$54.49_{(0.8)}$	$49.02_{(0.8)}$	41.12
Mono-bt-kNN-MT	$41.58_{(0.5)}$	$20.35_{(0.7)}$	$54.43_{(0.9)}$	$49.47_{(0.7)}$	41.46

Table 2: SacreBLEU scores of Facebook's WMT19 De-En model on the multi-domain test sets. The numbers in the parentheses at the bottom-right indicate that the model yielded the best translation performance when the hyperparameter lambda for interpolation is this value.

points. Additionally, translating the entire target data into the source language using the reverse translation model, followed by kNN-MT on this bilingual data, can yield an additional improvement of 0.34 BLEU points. However, this also implies a higher computational cost.

4.2 LLM Integration

In this experiment, we validate the effectiveness of the integration methods of NMT and LLM on the multi-domain test set. To ensure the consistency of the vocabulary between NMT and LLM for interpolation, for base NMT model, we used the WMT19 De-En model, which is trained on WMT19 De-En training data and the vocabulary of the Llama-2 model, as mentioned in Subsection 4. We examine various LLM integration methods, including the interpolation via kNN-MT whose pseudo datastore is constructed by the LLM, via the translation ability of the Llama model itself, and the fusion using LLM as a continuation generator, on Llama2, Llama2-chat, and ALMA models, respectively. The experimental results are given in Tabel 3.

From the results of the base models, it's clear that all three Llama models perform weaker in translation compared to the NMT model, even the translation-optimized ALMA model. Since the base NMT model is trained Utilizing the Llama dictionary, its performance averaged a loss of 1.67 BLEU points compared to Facebook's WMT19 model. In this experiment, in order to fair comparing with the method of kNN-MT with LLM pseudo datastore, we also experimented Pseudo-kNN-MT. Compared to base NMT, Pseudo-kNN-MT still significantly improves translation performance, with an average increase of 4.37 BLEU points on a slightly weaker NMT model. Retrieve-bt-kNN-MT and Mono-bt-kNN-MT can further improve over Pseudo-kNN-MT.

Unlike the NMT counterpart that constructs the datastore by NMT itself, our attempt to construct the datastore using LLM failed except the Llama2-zero-shot, resulting in a lower BLEU score than the base NMT. This indicates that LLMs are not good at compressing effective translation knowledge from pseudo-bilingual data. Besides, even the ALMA model, which has better translation capabilities, achieved similar BLEU scores to the other two Llama models. Moreover, the interpolation ratio λ was consistent with the other two Llama models, suggesting that the construction of a translation knowledge datastore from pseudo-bilingual data is not strongly correlated with the translation capabilities of LLMs.

Within the interpolation of the LLM translators, all three models can improve NMT translation to varying degrees on zero-shot and few-shot scenarios, with such enhancement being notably obvious in the more proficient ALMA model. Concurrently, optimal translation results are achieved on larger λ values for the stronger LLM translators, which means the latter can provide more translation knowledge to the NMT.

In the experimentation of fusing language models as text continuators, the Llama2 model, owing to its robust generative capability, aids in generating better translations, exhibiting an average improvement of 1.07 BLEU points over the base NMT. Conversely, conventional generative language models decrease the average BLEU score by 0.83 points compared to the base NMT. These results indicate that a language model solely trained for next token generation, if powerful enough, can be directly integrated during decoding and contribute to better translation. Furthermore, fine-tuned language models on validation sets in each domain also prove effective in achieving a similar impact.

Methods	IT	Koran	Law	Medical	Average
Base Models					
NMT	36.39	16.76	44.29	39.34	34.19
+ kNN-MT	$45.46_{(0.7)}$	$21.68_{(0.6)}$	$60.24_{(0.9)}$	$55.17_{(0.8)}$	45.64
+ Pseudo-kNN-MT	$38.97_{(0.3)}$	$18.14_{(0.4)}$	$51.14_{(0.4)}$	$47.19_{(0.5)}$	38.56
+ Retrieve-bt-kNN-MT	$39.59_{(0.5)}$	$19.26_{(0.5)}$	$52.14_{(0.6)}$	$46.71_{(0.6)}$	39.43
+ Mono-bt-kNN-MT	$40.22_{(0.7)}$	$20.14_{(0.6)}$	$52.40_{(0.7)}$	$46.85_{(0.7)}$	39.90
Llama2	34.19	11.71	37.52	33.96	29.35
Llama2-chat	29.03	12.97	28.54	33.83	26.09
ALMA	36.20	15.66	36.25	40.05	32.04
kNN-MT with LLM Pseudeo Datastore					
+Llama2-zero-shot	$35.53_{(0.3)}$	$17.91_{(0.3)}$	$44.39_{(0.3)}$	$42.09_{(0.3)}$	34.98
+Llama2-three-shot	$35.58_{(0.3)}$	$17.34_{(0.3)}$	$40.38_{(0.3)}$	$40.71_{(0.4)}$	33.50
+Llama2-chat-zero-shot	$35.58_{(0.3)}$	$17.33_{(0.3)}$	$40.33_{(0.3)}$	$40.53_{(0.4)}$	33.44
+Llama2-chat-three-shot	$35.49_{(0.3)}$	$17.37_{(0.3)}$	$40.50_{(0.3)}$	$40.89_{(0.4)}$	33.56
+ALMA-zero-shot	$35.48_{(0.3)}$	$17.37_{(0.3)}$	$40.46_{(0.3)}$	$42.52_{(0.4)}$	33.96
+ALMA-three-shot	$35.71_{(0.3)}$	$17.67_{(0.3)}$	$40.62_{(0.3)}$	$40.65_{(0.4)}$	33.66
LLM Translator Interpolation					
+Llama2-zero-shot	$37.68_{(0.2)}$	$17.36_{(0.2)}$	$44.94_{(0.1)}$	$39.95_{(0.1)}$	34.98
+Llama2-three-shot	$37.75_{(0.1)}$	$17.74_{(0.3)}$	$45.10_{(0.2)}$	$39.77_{(0.1)}$	35.09
+Llama2-chat-zero-shot	$37.73_{(0.1)}$	$17.21_{(0.1)}$	$44.86_{(0.2)}$	$40.03_{(0.2)}$	34.96
+Llama2-chat-three-shot	$38.33_{(0.2)}$	$17.41_{(0.2)}$	$45.17_{(0.2)}$	$40.44_{(0.3)}$	35.34
+ALMA-zero-shot	$38.67_{(0.4)}$	$17.70_{(0.4)}$	$45.78_{(0.3)}$	$41.10_{(0.5)}$	35.81
+ALMA-three-shot	$38.67_{(0.3)}$	$17.76_{(0.4)}$	$45.78_{(0.3)}$	$41.13_{(0.3)}$	35.84
Language Model Continuation Generator Fusion					
+Llama2-7B	$37.15_{(0.2)}$	$18.38_{(0.7)}$	$45.10_{(0.3)}$	$40.41_{(0.5)}$	35.26
+LM	$34.20_{(0.1)}$	$17.00_{(0.1)}$	$43.37_{(0.1)}$	$38.89_{(0.1)}$	33.36
+fine-tuned-LM	$35.79_{(0.1)}$	$18.35_{(0.3)}$	$47.03_{(0.2)}$	$42.69_{(0.2)}$	35.96

Table 3: SacreBLEU scores of WMT19 Llama-dictionary De-En model on the testsets of multi-domain data. The numbers in the parentheses at the bottom-right indicate same meaning as in Table 2.

4.3 The Influence of Nearest Neighbors Numbers for Per Query

The performance of kNN-MT is sensitive to the k, which is the number of the retrieved nearest neighbors. To investigate the impact of k on our approach, we conducted experiments on the Medical and Law test sets with varying values of k. In this experiment, the cross-lingual retrieval remains at 32. We only vary the number of neighbors retrieved from the kNN datastore. The experimental results in Figure 3 show that, as the k increases, both methods exhibit a trend of initially improving before declining, consistent with the findings in kNN-MT (Khandelwal et al., 2021), which suggest that appropriately increasing the number of neighbors is beneficial for translation but too many neighbors introduce noise and degrade translation quality. Moreover, on the Medical dataset, starting from k=8, Pseudo-kNN-MT surpasses Retrieve-bt-kNN-MT, which means Pseudo-kNN-MT is strong competitor to its back-translation counterpart.

4.4 The Influence of Cross-retrieval Similarity on Translation

To explore the applicability of our approach, we conducted experiments under low-resource settings on the WMT21 Is-En and Cs-En news translation tasks. The results indicated that Pseudo-kNN-MT failed to enhance translation quality, while Retrieve-bt-kNN-MT can improve it slightly. Details of the experiments are provided in the Appendix A.1.

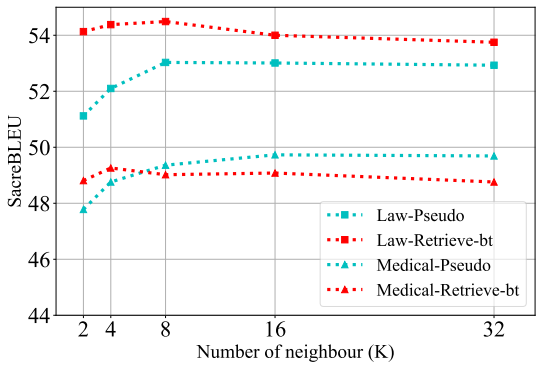

Figure 3: Impact of nearest neighbor numbers on the translation.

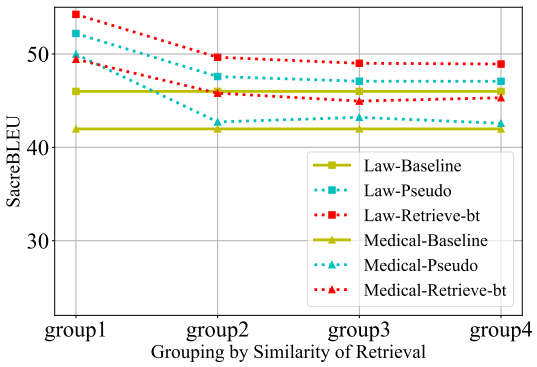

Figure 4: Impact of retrieval similarity on the translation results.

From these experiments, we observed that the similarity between the retrieval and source language is crucial. To investigate the impact of retrieval similarity on translation results, we conducted this experiment on Medical and Law test sets. We partitioned the retrieved 32 target sentences into four groups according to their similarity. Here, we measure the similarity of two vectors by using $L2$ distance from FAISS library, and closer distances indicate greater similarities. Each group consisted of eight sentences, which were used as retrievals of each group. We set k for kNN search to 4, while keeping other experimental settings consistent with the main experiment. The results are presented in Figure 4. The average distances of retrieval from Group 1 to Group 4 are as follows: for Medical (0.5764, 0.6834, 0.7232, 0.7491) and for Law (0.5798, 0.6648, 0.6964, 0.7167). This means that the similarity decreases sequentially from Group 1 to Group 4. This indicates that the higher the similarity of target language retrieval, the more significant the improvement in translation performance.

5 Related Works

As a mature and widely known method, kNN-MT(Khandelwal et al., 2021) has many variants. Zheng et al. (2021) introduce adaptive knn-mt, which can adaptively choose k to decrease noisy neighbors. Deguchi et al. (2023) introduce subset kNN-MT, which accelerates inference speed since it only retrieves in a small subset according to source similarity. We also leverage subset retrieval while relying cross language similarity. Wang et al. (2022) introduces cluster-based kNN-MT, which adopts a compact network to prune feature datastore extremely. Martins et al. (2022) introduces chuck-based kNN-MT, which transforms retrieve granularity from single token to chunk. Dai et al. (2023) introduces a fast kNN-MT method, which combines subset kNN-MT and distance-aware λ together. Liu et al. (2023) introduced kNN-TL, which explores how to combine the transfer learning method and kNN-MT in low-resource scenery. Zhu et al. (2023b) introduces INK, which is a training framework refines the representation space of an NMT model according to the extracted kNN knowledge to avoid the expensive inference cost of kNN-MT method.Also, Wang et al. (2023) explores non-parametric kNN-MT method can improve machine translation models at the fine-tuning stage. Cao et al. (2023) introduces a method to deal with the gap between the upstream NMT model and downstream domains datastore, which makes kNN-MT adopt better for downstream tasks by reconstructing datastore.

6 Conclusion and Future Work

In this paper, we propose the pseudo-kNN-MT method, and achieve significant improvements on domain adaptation task, validating the effectiveness of incorporating target monolingual data in the kNN-MT. Within this method, we employ a cross-lingual retrieval model to retrieve semantically similar sentences from the target language data and pair them with the input sentences to construct pseudo-bilingual data, which is then used to build a key-value datastore. We also explore methods of utilizing large language models to construct the key-value datastore. In future work, we will further explore LLM prompts suitable for this scenario and explore the potential of LLMs in this

context.

7 Limitation

Our proposed pseudo-kNN-MT method is significantly influenced by the similarity of the retrieved target language sentence. If the retrieved target sentence matches the source sentence semantically, it can help the translation; otherwise, it may not, and could even degrade translation performance. Therefore, its applicability is limited. Specifically, when translating in a particular domain, the target language data used should also belong to that domain to ensure similarity in retrieval. If this target language data can cover the domain extensively, then our method can perform even better.

References

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

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 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Zhiwei Cao, Baosong Yang, Huan Lin, Suhang Wu, Xiangpeng Wei, Dayiheng Liu, Jun Xie, Min Zhang, and Jinsong Su. 2023. Bridging the domain gaps in context representations for k-nearest neighbor neural machine translation. *arXiv preprint arXiv:2305.16599*.

Yuhan Dai, Zhirui Zhang, Qiuzhi Liu, Qu Cui, Weihua Li, Yichao Du, and Tong Xu. 2023. Simple and scalable nearest neighbor machine translation. In *The Eleventh International Conference on Learning Representations*.

Hiroyuki Deguchi, Taro Watanabe, Yusuke Matsui, Masao Utiyama, Hideki Tanaka, and Eiichiro Sumita. 2023. Subset retrieval nearest neighbor machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 174–189, Toronto, Canada. Association for Computational Linguistics.

Duane K. Dougal and Deryle Lonsdale. 2020. Improving NMT quality using terminology injection. In *Proceedings of the Twelfth Language Resources and Evaluation Conference*, pages 4820–4827, Marseille, France. European Language Resources Association.

Akiko Eriguchi, Spencer Rarrick, and Hitokazu Matsushita. 2019. Combining translation memory with neural machine translation. In *Proceedings of the 6th Workshop on Asian Translation*, pages 123–130, Hong Kong, China. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022a. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 878–891. Association for Computational Linguistics.

Fangxiaoyu Feng, Yinfei Yang, Daniel Cer, Naveen Arivazhagan, and Wei Wang. 2022b. Language-agnostic BERT sentence embedding. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, May 22-27, 2022*, pages 878–891. Association for Computational Linguistics.

Pengzhi Gao, Liwen Zhang, Zhongjun He, Hua Wu, and Haifeng Wang. 2023. Learning multilingual sentence representations with cross-lingual consistency regularization. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: EMNLP 2023 - Industry Track, Singapore, December 6-10, 2023*, pages 243–262. Association for Computational Linguistics.

Çaglar Gülçehre, Orhan Firat, Kelvin Xu, Kyunghyun Cho, Loïc Barrault, Huei-Chi Lin, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. 2015. On using monolingual corpora in neural machine translation. *CoRR*, abs/1503.03535.

Eva Hasler, Adrià de Gispert, Gonzalo Iglesias, and Bill Byrne. 2018. Neural machine translation decoding with terminology constraints. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, pages 506–512, New Orleans, Louisiana. Association for Computational Linguistics.

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.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2021. Billion-scale similarity search with gpus. *IEEE Trans. Big Data*, 7(3):535–547.

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

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

Ziheng Li, Shaohan Huang, Zihan Zhang, Zhi-Hong Deng, Qiang Lou, Haizhen Huang, Jian Jiao, Furu Wei, Weiwei Deng, and Qi Zhang. 2023. Dual-alignment pre-training for cross-lingual sentence embedding. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 3466–3478. Association for Computational Linguistics.

Shudong Liu, Xuebo Liu, Derek F. Wong, Zhaocong Li, Wenxiang Jiao, Lidia S. Chao, and Min Zhang. 2023. kNN-TL: k-nearest-neighbor transfer learning for low-resource neural machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1878–1891, Toronto, Canada. Association for Computational Linguistics.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2022. Chunk-based nearest neighbor machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 4228–4245, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, August 1-2, 2019 - Volume 2: Shared Task Papers, Day 1*, pages 314–319. Association for Computational Linguistics.

Matt Post. 2018. A call for clarity in reporting BLEU scores. In *Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, Belgium, Brussels, October 31 - November 1, 2018*, pages 186–191. Association for Computational Linguistics.

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

Abudurexiti Reheman, Tao Zhou, Yingfeng Luo, Di Yang, Tong Xiao, and Jingbo Zhu. 2023. Prompting neural machine translation with translation memories. *Proceedings of the AAAI Conference on Artificial Intelligence*, 37(11):13519–13527.

Ricardo Rei, José G. C. de Souza, Duarte M. Alves, Chrysoula Zerva, Ana C. Farinha, Taisiya Glushkova, Alon Lavie, Luísa Coheur, and André F. T. Martins. 2022. COMET-22: unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation, WMT 2022, Abu Dhabi, United Arab Emirates (Hybrid), December 7-8, 2022*, pages 578–585. Association for Computational Linguistics.

Danielle Saunders. 2022. Domain adaptation and multi-domain adaptation for neural machine translation: A survey. *Journal of Artificial Intelligence Research*, 75:351–424.

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, ACL 2016, August 7-12, 2016, Berlin, Germany, Volume 1: Long Papers*. The Association for Computer Linguistics.

Ilya Sutskever, Oriol Vinyals, and Quoc V Le. 2014. Sequence to sequence learning with neural networks. *Advances in neural information processing systems*, 27.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

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

Jiayi Wang, Ke Wang, Yuqi Zhang, Yu Zhao, and Pontus Stenetorp. 2023. Non-parametric, nearest-neighbor-assisted fine-tuning for neural machine translation. *arXiv preprint arXiv:2305.13648*.

Haoran Xu, Young Jin Kim, Amr Sharaf, and Hany Hassan Awadalla. 2023. A paradigm shift in machine translation: Boosting translation performance of large language models.

Jitao Xu, Josep-Maria Crego, and Jean Senellart. 2020. Boosting neural machine translation with similar translations. In *Annual Meeting of the Association for Computational Linguistics*, pages 1570–1579. Association for Computational Linguistics.

Biao Zhang, Barry Haddow, and Alexandra Birch. 2023. Prompting large language model for machine translation: A case study. In *International Conference on Machine Learning, ICML 2023, 23-29 July 2023, Honolulu, Hawaii, USA*, volume 202 of *Proceedings of Machine Learning Research*, pages 41092–41110. PMLR.

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. Adaptive nearest neighbor machine translation. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, pages 368–374, Online. Association for Computational Linguistics.

Wenhao Zhu, Hongyi Liu, Qingxiu Dong, Jingjing Xu, Lingpeng Kong, Jiajun Chen, Lei Li, and Shujian Huang. 2023a. Multilingual machine translation with large language models: Empirical results and analysis. *CoRR*, abs/2304.04675.

Wenhao Zhu, Jingjing Xu, Shujian Huang, Lingpeng Kong, and Jiajun Chen. 2023b. Ink: Injecting knn knowledge in nearest neighbor machine translation. *arXiv preprint arXiv:2306.06381*.

A Other Experiments

A.1 Low Resource Settings

To verify the performance of our method in low-resource scenarios, we conducted experiments on the datasets from Is-En and Cs-En news translation tasks of WMT 21. For data selection, we combined all datasets except for the bilingual obtained from machine translation, and then performed uniform sampling on the cleaned bilingual data to obtain a bilingual dataset. The monolingual target language data utilized the news2021 data from `news-crawl/en`. After cleaning, we also used uniform sampling to obtain final monolingual data. In the back-translation method, following Sennrich et al. (2016), we initially trained a reverse NMT model from bilingual data to translate target language monolingual data back into the source language, resulting in 1 million synthetic-bilingual data. Subsequently, we mixed this data with the original bilingual data and trained an NMT model on this combined dataset. Data statistics are presented in Table 4, and experimental results are provided in Table 5.

B Comet Scores

Here we present the COMET evaluation results for the main experiment and the LLM integration experiments. Specifically, Table 6 corresponds to Table 2 in the main text, and Table 7corresponds to Table 3 in the main part of the paper.

Split	Is-En	Cs-En	En
Train	500K	500K	1M
Valid	2004	2082	-
Test	1000	1000	-

Table 4: Statistics of datasets for low resource translation scenario.

Split	Is-En	Cs-En
NMT	21.46	21.46
Back-translation	25.69	23.68
Mono-bt-kNN-MT	22.26	22.54
Retrieve-bt-kNN-MT	21.79	21.97

Table 5: SacreBLEU scores for low resource translation scenario.

Methods	IT	Koran	Law	Medical	Average
NMT	.8246	.7257	.8538	.8316	.8089
kNN-MT	.8489	.7352	.8717	.8486	.8261
Pseudo-kNN-MT	.8251	.7224	.8468	.8243	.8046
Retrieve-bt-kNN-MT	.8264	.7314	.8611	.8384	.8143
Mono-bt-kNN-MT	.8296	.7300	.8596	.8393	.8146

Table 6: COMET scores of Facebook's WMT19 De-En model on the multi-domain test sets.

Methods	IT	Koran	Law	Medical	Average
Base Models					
NMT	.8236	.7244	.8547	.8335	.8090
+ kNN-MT	.8616	.7342	.8748	.8541	.8311
+ Pseudo-kNN-MT	.8338	.7208	.8492	.8252	.8072
+ Retrieve-bt-kNN-MT	.8346	.7239	.8630	.8409	.8156
+ Mono-bt-kNN-MT	.8354	.7304	.8653	.8428	.8184
Llama2	.7456	.6827	.7678	.8035	.7499
Llama2-chat	.7548	.7773	.7954	.7894	.7792
ALMA	.7700	.7643	.7985	.8049	.7844
kNN-MT with LLM Pseudeo Datastore					
+Llama2-zero-shot	.7747	.7830	.8127	8064	.7942
+Llama2-three-shot	.7760	.7859	.7992	.8017	.7907
+Llama2-chat-zero-shot	.7720	.7858	.8000	.8003	.7895
+Llama2-chat-three-shot	.7773	.7863	.7991	.8010	.7909
+ALMA-zero-shot	.8240	.7857	.8000	.8083	.8045
+ALMA-three-shot	.7760	.7860	.8000	.8006	.7907
LLM Translator Interpolation					
+Llama2-zero-shot	.7819	.7932	.8189	.8085	.8006
+Llama2-three-shot	.7823	.7933	.8189	.8087	.8008
+Llama2-chat-zero-shot	.7869	.7978	.8204	.8146	.8049
+Llama2-chat-three-shot	.7889	.7975	.8208	.8142	.8054
+ALMA-zero-shot	.7877	.7992	.8206	.8124	.8050
+ALMA-three-shot	.7866	.7977	.8204	.8144	8048
Language Model Continuation Generator Fusion					
+Llama2-7B	.7782	.7913	.8177	.8096	.7992
+LM	.7748	.7856	.8128	.8058	.7947
+fine-tuned-LM	.7772	.7891	.8165	.8093	.7980

Table 7: COMET scores of WMT19 Llama-dictionary De-En model on the testsets of multi-domain data.