Domain-Aware k-Nearest-Neighbor Knowledge Distillation for Machine Translation

Zhexuan Wang[1*] **Shudong Liu**[2*] **Xuebo Liu**[1†] **Miao Zhang**[1] **Derek F. Wong**[2] **Min Zhang**[1]
[1]Institute of Computing and Intelligence, Harbin Institute of Technology, Shenzhen, China
[2]NLP^{2}CT Lab, Department of Computer and Information Science, University of Macau
wangzx1403@gmail.com, nlp2ct.shudong@gmail.com
{liuxuebo,zhangmiao,zhangmin2021}@hit.edu.cn, derekfw@um.edu.mo

Abstract

kNN-MT has utilized neighborhood knowledge for auxiliary decoding, significantly improving translation performance. Subsequently, kNN-KD transitions the use of neighborhood knowledge from the decoding phase to the training phase, to address the temporal and spatial inefficiencies inherent in kNN-MT. However, kNN-KD transfers all the kNN knowledge arbitrarily, which has the potential to restrict the learning of student models. In this paper, we propose a novel domain-aware kNN-KD method, which filters out domain-relevant neighborhood knowledge for learning in the distillation process. Notably, this entire process exclusively utilizes the neighborhood knowledge of the original model, eliminating the need for establishing any additional datastores. Experiments on four domain translation tasks demonstrate that our method achieves state-of-the-art performance, realizing an average gain of 1.55 COMET and 1.42 BLEU scores, by further enhancing the translation of rare words. Source code can be accessed at https://github.com/wangzx1219/Dk-KD.

1 Introduction

The field of neural machine translation (NMT, Vaswani et al., 2017; Ng et al., 2019) has witnessed significant advancements, resulting in noteworthy improvements in various translation tasks. Among these innovations, the introduction of kNN-MT (Khandelwal et al., 2020) stands out as a pioneering approach. This method leverages neighborhood knowledge for assisted decoding, enhancing the translation capabilities of the model. In kNN-MT, the translation process benefits from an expanded contextual understanding, allowing for more accurate and contextually relevant translations. This integration effectively bridges the gap between raw translation output and nuanced, context-aware language understanding.

Although the enhancement of model translation capability by kNN-MT is significant, its retrieval cost in the decoding process and consumption of storage space are non-negligible, which limits the application of kNN-MT in practical scenarios. The existing methods try to transfer knowledge from the kNN datastore into new models. kNN-KD (Yang et al., 2022) employs knowledge distillation (KD, Hinton et al., 2015) for training the network from scratch, which, in some cases, leads to suboptimal and unstable outcomes. INK (Zhu et al., 2023) propose to migrate knowledge into the adapter module (Bapna and Firat, 2019b), which is more efficient due to its smaller parameter size and generally yields better results. However, existing methods indiscriminately transfer knowledge from the datastore, a practice we believe may have its limitations.

Building on the pros and cons of the previous methods, this paper introduces domain-aware kNN-KD (Dk-KD), aimed at extracting the most valuable knowledge from the kNN datastore within the domain. The approach begins by training a more refined teacher model, specifically optimized for storing kNN knowledge, which facilitates the learning process of the adapter. Subsequently, kNN knowledge is selectively distilled from the teacher model into the adapter, ensuring targeted and efficient knowledge transfer. Specifically, we first construct a datastore using the representations from the original NMT model. Subsequently, we utilize domain-relevant knowledge to aid in training the teacher model. Finally, we leverage both the teacher and domain-relevant knowledge to assist in training the final model. Domain-aware knowledge selection is employed to filter domain-relevant knowledge from the retrieved representations.

Experiments on four domain translation tasks indicate that Dk-KD outperforms other advanced models that utilize kNN knowledge for KD. Our

[*] Equal contribution
[†] Corresponding Author

Findings of the Association for Computational Linguistics: ACL 2024, pages 9458–9469
August 11-16, 2024 ©2024 Association for Computational Linguistics

primary contributions are as follows:

- We introduce Dk-KD, which extracts domain-relevant knowledge from a general-domain NMT model by establishing a kNN datastore.

- We employ a two-step distillation process to extract domain-relevant knowledge from the datastore. Experiments show that domain-aware knowledge selection within the datastore is beneficial for model learning.

- Dk-KD focuses on the learning of domain knowledge, improving the translation of domain-specific low-frequency words.

2 Background

2.1 kNN-MT

kNN-MT (Khandelwal et al., 2020) is a method that employs retrieval-augmented techniques in text generation. It operates by retrieving the k closest data points from a vast datastore during the decoding phase, thereby assisting a pre-trained Neural Machine Translation (NMT) model by providing contextually relevant information. The process of kNN-MT can be divided into two main steps:

Building kNN Datastore The datastore is a crucial aspect of the kNN-MT system, where it stores the knowledge of a pre-trained NMT model in the form of key-value pairs. In this context, the key represents the output representation at each time step, and the value corresponds to the target token, which is the accurate translation. Specifically, given a set of training data $(\mathcal{X}, \mathcal{Y})$, we process each sentence pair $(\boldsymbol{x}, \boldsymbol{y})$ to construct the datastore $(\mathcal{K}, \mathcal{V})$. This process can be understood as structuring the knowledge learned by the model in a way that facilitates efficient retrieval and utilization later on. The construction for the datastore is:

$$(\mathcal{K}, \mathcal{V}) = \bigcup_{(\boldsymbol{x},\boldsymbol{y})\in(\mathcal{X},\mathcal{Y})} \{(f(\boldsymbol{x}, \boldsymbol{y}_{<i}), \boldsymbol{y}_i) \ \forall \boldsymbol{y}_i \in \boldsymbol{y}\}. \tag{1}$$

The output representation of the NMT model at a specific time step i, denoted as $f(x, y_{<i})$, serves as the key, while the corresponding reference target token y_i is the value. This is because each time step's output representation corresponds to a target token, and all these representations need to be stored for subsequent retrieval and utilization.

Model Inference During the inference process, at each decoding step i, kNN-MT transforms the current translation context into a representation, $f(x, y_{<i})$. This representation, f, is then used to query the k nearest neighbors $\mathcal{N}_k^i = \{(k_j, v_j) \mid j \in \{1, 2, \ldots, k\}\}$ from the datastore by comparing the l_2 distance, effectively identifying the k most similar historical contexts to the current one. The distribution for kNN is:

$$p_{k\text{NN}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) \propto \sum_{(k_j,v_j)\in\mathcal{N}_k^i} \mathbb{1}_{y_i=v_j} \exp\left(-\frac{d(k_j, f(\boldsymbol{x}, y_i))}{\tau}\right), \tag{2}$$

where τ is the temperature, and $d(\cdot, \cdot)$ is the l_2 distance function. The final probability distribution predicting the next token y_i is the interpolation of two distributions with a tuning parameter λ:

$$p(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) = (1-\lambda)p_{\text{MT}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) + \lambda p_{\text{kNN}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}). \tag{3}$$

By incorporating external knowledge, the retrieval distribution has been adjusted to refine the initial NMT distribution, thus better performance.

2.2 kNN-KD

kNN-KD was proposed to address the slow decoding speed and large storage space issues of kNN-MT. It shifts the phase of utilizing kNN knowledge from decoding to training. Specifically, this is achieved through the KD method, during the model training process, the kNN distribution stored in the datastore acts as the teacher. For all translation contexts $(\boldsymbol{x}, \boldsymbol{y}_{<i})$ during the training process, kNN-KD treats them as queries and performs retrieval in the datastore, subsequently obtaining the retrieval results $\mathcal{N}$.

Apply kNN to KD Let p_{kNN} represent the kNN distribution retrieved from the datastore. To train an NMT model from scratch using KD, the loss during distillation is:

$$\begin{aligned}\mathcal{L}_{\text{kNN}-\text{KD}} = \sum_{y_i\in\text{V}} p_{\text{kNN}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) \\ \times \log\left(\frac{p_{\text{kNN}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i})}{p_{\text{MT}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i})}\right),\end{aligned} \tag{4}$$

where V represents the target language vocabulary. The final loss is obtained by balancing the cross-entropy (CE) loss and the distillation loss through the parameter δ:

$$\mathcal{L} = (1-\delta)\mathcal{L}_{\text{CE}} + \delta\mathcal{L}_{\text{kNN}-\text{KD}}. \tag{5}$$

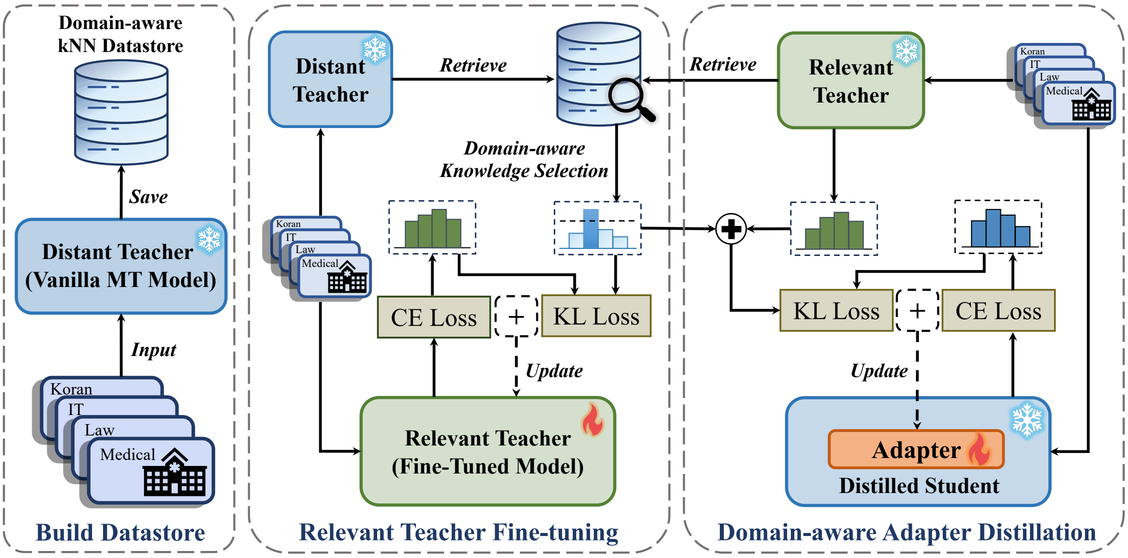
Domain-aware kNN Datastore
Save
Distant Teacher (Vanilla MT Model)
Input
Koran
IT
Law
Medical
Build Datastore
Distant Teacher
Retrieve
Domain-aware Knowledge Selection
CE Loss
+
KL Loss
Update
Relevant Teacher (Fine-Tuned Model)
Relevant Teacher Fine-tuning
Retrieve
Relevant Teacher
KL Loss
CE Loss
Update
Adapter
Distilled Student
Domain-aware Adapter Distillation

Figure 1: An overview of our proposed Dk-KD method.

3 Domain-Aware kNN-KD

3.1 Motivation

Our objective is to maximize the utilization of the datastore established by existing models and inject domain-relevant knowledge into adapter layers through a dual KD process, a technique not implemented in previous methodologies. Recent studies, such as kNN-KD (Yang et al., 2022) and INK (Zhu et al., 2023), have shifted the kNN retrieval from the datastore from the decoding phase to the training process, utilizing kNN representations to facilitate the model's learning of knowledge. However, they overlook the need to tailor knowledge learning to specific domains, which may lead to inadequate learning outcomes for the model regarding domain-specific vocabulary. Therefore, we propose Domain-Aware kNN-KD, which focuses on learning domain-relevant knowledge from kNN representations during the training process. The overall training process is depicted in Figure 1.

3.2 Domain-Aware Datastore Construction

As illustrated in the first part of Figure 1, for a specific domain, we employ the original NMT model to perform forced decoding on sentences from the domain's training set $(\mathcal{X}, \mathcal{Y})$. For each sentence, we obtain multiple context representations, which, along with the corresponding target tokens, are saved into a domain-aware kNN datastore $(\mathcal{K}, \mathcal{V})$.

3.3 Domain-Aware Teacher Finetuning

The purpose of this step is to train a teacher model that possesses a profound understanding of specific domain knowledge, termed a domain-aware model. Beginning with a pre-trained NMT model, which lacks a deep understanding of any specific domain, we tackle this issue by adopting a fine-tuning strategy and incorporating a domain-aware datastore. The key aspect here is to perform domain-aware knowledge selection on the kNN representations, enabling us to accurately identify and extract domain-relevant knowledge through this process. This domain-specific knowledge is instrumental in training a teacher model that comprehends the nuances and demands of the specific domain.

During fine-tuning, upon obtaining the kNN distribution, we assess the magnitude of the target token $\bar{y}_i$ within this distribution. If the probability $p_{\text{kNN}}^{\text{DT}}(\bar{y}_i)$ is greater than or equal to a threshold α, it is considered domain-relevant knowledge. Conversely, if $p_{\text{kNN}}^{\text{DT}}(\bar{y}_i)$ is less than α, it is deemed domain-irrelevant knowledge. We calculate the Kullback-Leibler (KL) divergence between the domain-relevant kNN distribution and the output distribution of the NMT model:

$$\begin{aligned}\mathcal{L}_{\text{KL}} = &\sum_{y_i \in \text{V}} p_{\text{kNN}}^{\text{DT}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i}) \\ &\times \log\left(\frac{p_{\text{kNN}}^{\text{DT}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i})}{p_{\text{MT}}^{\text{RT}}(y_i|\boldsymbol{x}, \boldsymbol{y}_{<i})}\right),\end{aligned} \quad (6)$$

Furthermore, the acronyms DT and RT refer to the Distant Teacher and the Relevant Teacher, corresponding to the original NMT model and the domain-aware teacher model, respectively. The final loss for the entire sentence can be formulated as:

$$\mathcal{L}_{\text{Teacher}} = \mathcal{L}_{\text{CE}} + \sum_{i=1}^{I} \mathbb{1}_{(p^{\text{DT}}_{\text{kNN}}(\bar{y}_i)\geq\alpha)}\mathcal{L}^{i}_{\text{KL}}, \tag{7}$$

where I represents the length of the sentence. The indicator function therein indicates the screening process, where only valid kNN knowledge is included in the calculation of the loss.

The parameters of the original NMT model are defined as θ_{N}, and the parameters of the teacher model during fine-tuning are defined as θ_{T}. The training objective for this step is formulated as:

$$\tilde{\theta}_{\text{T}} = \arg\min \mathcal{L}_{\text{Teacher}}\Big((\mathcal{X},\mathcal{Y});(\mathcal{K},\mathcal{V});\theta_{\text{N}}^{\text{❄}};\theta_{\text{T}}^{\text{🔥}})\Big) \tag{8}$$

3.4 Domain-Aware Adapter Distillation

The aim of this step is to refine the domain-aware teacher model through fine-tuning and to further extract domain-relevant knowledge from the domain-aware datastore. This process begins with the adapter layers attached to the original NMT model. These adapter layers are additional layers introduced into the NMT model architecture, including the encoder and decoder, aimed at enhancing the model's adaptability and understanding of specific domain knowledge without significantly altering the overall structure of the model. During this process, domain-aware knowledge selection is still performed on the kNN representations, allowing for the precise extraction of domain-relevant knowledge from the datastore through this filtering mechanism. This domain-relevant knowledge is then used to optimize the adapter layers.

We utilize the relevant teacher model to retrieve kNN distribution from the datastore. Subsequently, we compare the probability of the golden label in this distribution with α and filter out distributions where the probability is less than α, in order to isolate knowledge with stronger domain-relevant. The distilled distributions are then integrated with the distribution output by the teacher model. The final representation of the teacher probability distribution p_{T} is:

$$p_{\text{T}} = \begin{cases} \lambda p^{\text{RT}}_{\text{MT}} + (1-\lambda)p^{\text{RT}}_{\text{kNN}}, & \text{if } p^{\text{RT}}_{\text{kNN}}(\bar{y}_i) \geq \alpha, \\ p^{\text{RT}}_{\text{MT}}, & \text{if } p^{\text{RT}}_{\text{kNN}}(\bar{y}_i) < \alpha. \end{cases} \tag{9}$$

The method for calculating the final loss is:

$$\mathcal{L}_{\text{KL}} = \sum_{y_i \in \text{V}} p_{\text{T}}(y_i|\boldsymbol{x},\boldsymbol{y}_{<i}) \log\left(\frac{p_{\text{T}}(y_i|\boldsymbol{x},\boldsymbol{y}_{<i})}{p^{\text{STU}}_{\text{MT}}(y_i|\boldsymbol{x},\boldsymbol{y}_{<i})}\right). \tag{10}$$

Dataset	IT	Koran	Medical	Law
Train	222,927	17,982	248,009	467,309
Dev	2,000	2,000	2,000	2,000
Test	2,000	2,000	2,000	2,000

Table 1: Statistics of four domain translation datasets.

$$\mathcal{L}_{\text{Student}} = \mathcal{L}_{\text{CE}} + \beta\sum_{i=1}^{I}\mathcal{L}^{i}_{\text{KL}}. \tag{11}$$

where STU denotes the final student model and β is a balanced coefficient.

The parameters of the adapter layer attached to the NMT model are defined as θ_A, hence the training objective for this step can be represented as:

$$\hat{\theta}_A = \arg\min \mathcal{L}_{\text{Student}}\Big((\mathcal{X},\mathcal{Y});(\mathcal{K},\mathcal{V});\tilde{\theta}_{\text{T}}^{\text{❄}};\theta_{A}^{\text{🔥}})\Big) \tag{12}$$

3.5 Discussion

The original NMT model is generally not specialized to a specific domain since it is trained on data from various domains. However, after training on a dataset specific to a domain, we consider the teacher model to be focused on that domain. Similarly, the distribution obtained from the datastore, after filtering, should also be focused on the domain. This can make our final model align with the domain as closely as possible. The final model includes the original model with adapter layers added, while keeping the parameters of the original model's embedding layer, encoder, and decoder unchanged. Therefore, our model can outperform vanilla kNN-MT systems in terms of inference time and memory costs, achieving a balance between time, memory space, and model effectiveness.

4 Experiments

4.1 Experimental Setup

Testbed Model and Dataset We selected the winning model from the WMT'19[1] German-to-English (Ng et al., 2019) news translation task as the basis for constructing our NMT model, utilizing it for both translation and datastore. In this paper, we refer to it as the domain-agnostic model. We conduct experiments on the multi-domain datasets, which include Koran, IT, Medical, and Law. The statistics of these datasets are shown in Table 1. We follow Ng et al. (2019) to tokenize the sentence into subword units.

[1] `https://github.com/facebookresearch/fairseq/tree/main/examples/wmt19`

Parameters	IT	Koran	Medical	Law
Domain-Aware Teacher Finetuning				
k	16	16	16	16
τ	10	10	10	100
max-epochs	100	60	100	100
max-tokens	4096	4096	4096	8192
Domain-Aware Adapter Distillation				
k	8	16	8	4
τ	100	10	50	10
λ	0.1	0.2	0.1	0.1
max-epochs	80	80	80	80
max-tokens	8192	8192	8192	8192
adapter-ffn	8192	512	8192	8192

Table 2: Dk-KD settings for different datasets.

Baselines For the purpose of comparison, we outline the performance of the original model here and also test two methods of applying neighborhood knowledge during the decoding phase. Moreover, we also compare with the methods distilling neighborhood knowledge into the model.

- **Vanilla NMT** (Vaswani et al., 2017) introduces the Transformer model, which adopts a self-attention mechanism.
- **Vanilla kNN-MT** (Khandelwal et al., 2020) utilizes neighborhood knowledge to enhance translation during the decoding phase.
- **Robust kNN-MT** (Jiang et al., 2022) enhances kNN-MT performance by dynamically adjusting decoding hyperparameters.
- **kNN-KD** (Yang et al., 2022) aims to train an NMT model from scratch that distills kNN knowledge into it.
- **INK** (Zhu et al., 2023) smooths the representation space using kNN knowledge from an asynchronously refreshed datastore.

Evaluation During inference, we set beam size as 4 and length penalty as 0.6. We used the following two metrics for MT evaluation:

- **COMET** (Rei et al., 2020), a machine learning-based evaluation tool specifically designed for assessing the quality of machine translations. It quantifies the accuracy and fluency of translations by comparing the translated text with reference texts translated by humans. We report the COMET score, calculated using the publicly available wmt20-comet-da[2] model.
- **BLEU** (Papineni et al., 2002), a widely-used metric for automatically evaluating the quality of machine translations. We present our results using case-sensitive, detokenized sacrebleu[3].

4.2 Implementation Details

We employed the fairseq[4] toolkit for model implementation, leveraging $k$nn-box[5] (Zhu et al., 2024) for the construction and kNN retrieval of the datastore. Furthermore, we utilized FAISS[6] for efficient search operations. In the first phase, we set the learning rates for the Koran, IT, and Medical domains to 1e-4, respectively. For the Law domain, the learning rate was adjusted to 3e-4.In the second phase, during the training of the adapter layers, we standardized the learning rates across all four domains to 3e-4, respectively, and set β to 2.0.All Dk-KD systems were trained on A100 GPUs. In the two-stage distillation, we set the threshold α to 0.5. Throughout the entire training process, we set the label smoothing parameter to 0.1, the weight decay to 0.0001, used the Adam optimizer with betas of (0.9, 0.98), and adopted an inverse square root learning rate scheduler with a warm-up of 4000 updates. We selected the model with the highest validation BLEU score for testing.

The parameters involved in the distillation process of Dk-KD are outlined in Table 2, including those for kNN retrieval, max-epochs, and batch-size for training. During the fine-tuning phase of the teacher model, due to the small size of the Koran dataset, we appropriately reduced the max-epochs to prevent the model from overfitting to the validation set, which could lead to suboptimal results. With the larger Law dataset, to ensure thorough learning, we adjusted its batch size. Similarly, during the distillation adapter phase, we also reduced the dimension of the adapter layers specifically for the characteristics of the Koran dataset. To ensure the learning effectiveness and training speed of the adapter layers, we standardized the max-tokens to 8192.

[2] https://github.com/Unbabel/COMET
[3] https://github.com/mjpost/sacrebleu
[4] https://github.com/facebookresearch/fairseq
[5] https://github.com/NJUNLP/knn-box
[6] https://github.com/facebookresearch/faiss

Method	IT		Koran		Medical		Law		Avg.	
	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU
Vanilla NMT	39.29	38.35	-1.36	16.26	46.91	39.99	57.54	45.48	35.60	35.02
FT	67.34	49.95	6.69	21.80	59.71	57.58	71.24	63.60	51.25	48.23
Adapter	66.82	48.50	2.49	21.76	60.61	57.17	70.18	61.00	50.03	47.11
Training: Vanilla | Decoding: kNN Datastore Retrieval										
Vanilla kNN-MT	51.76	45.64	3.01	20.82	52.84	54.23	66.47	61.40	43.52	45.52
Robust kNN-MT	58.10	48.62	2.07	19.65	57.54	57.27	69.82	63.79	46.88	47.33
Training: kNN Knowledge Distillation | Decoding: Vanilla										
kNN-KD	57.07	44.30	-31.64	15.79	56.14	55.92	67.93	62.32	37.38	44.58
INK	67.26	48.31	6.60	22.56	60.45	57.33	71.31	61.54	51.41	47.44
Dk-KD	**69.41**	**50.30**	**9.41**	**23.37**	**60.75**	**57.80**	**72.26**	**63.96**	**52.96**	**48.86**

Table 3: Main results on the four domain translation tasks. Both COMET score and BLEU score reflect that our Dk-KD method achieves significant improvements compared with INK on all the tasks ($p < 0.1$).

Method	IT		Koran		Medical		Law		Avg.	
	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU
Dk-KD	**69.41**	**50.30**	**9.41**	**23.37**	**60.75**	**57.80**	**72.26**	**63.96**	**52.96**	**48.86**
w/o Relevant Teacher	68.19	49.13	7.60	22.12	60.43	57.67	72.09	63.40	52.08	48.08

Table 4: "w/o Relevant Teacher" indicates the step of training the relevant teacher model is omitted, allowing for direct training of the adapter using domain-specific knowledge from the datastore.

4.3 Main Results

The comparative results of different systems are presented in Table 3. Given the original NMT model's lack of domain sensitivity, its performance on tests across the four domains was relatively moderate. While the conventional kNN-MT model demonstrates improved translation quality, its decoding phase incurs significant time and storage costs, which limit its widespread application. The kNN-KD approach requires training from scratch, resulting in unstable outcomes. Meanwhile, the INK training framework demands asynchronous updates to the datastore, resulting in a high demand for disk interaction during training. Dk-KD optimizes the balance between memory usage and method effectiveness, eliminating the need for extra disk operations during training, ensuring a more stable and consistent learning environment, thereby validating the method's rationality. Through testing on four domain tasks, Dk-KD achieved commendable translation outcomes, demonstrating the effectiveness of Dk-KD.

5 Analysis

5.1 Method Justification

Effect of Domain-Aware Teacher Finetuning To validate the significance of training the teacher model, we omitted the distillation process in Step 1. Specifically, we directly extracted the kNN probability distribution from the datastore during the training process, applied the same threshold to select domain-relevant knowledge, and used the filtered results for KD on the adapter layer. Although there was an improvement compared to adapter baseline, the results were still not sufficiently excellent and stable. Table 4 indicates that distilling the adapter layer using a teacher model significantly enhances translation quality. These findings underscore the effectiveness of integrating domain-specific knowledge into a teacher model, highlighting it as an effective strategy for extracting domain-relevant knowledge from the datastore.

Effect of Domain-Aware Knowledge Selection To validate the effectiveness of the domain-aware distillation approach, we conducted four sets of experiments for two-stage KD on the Koran and IT datasets, with results depicted in Figure 2. Here, Dk-KD denotes "Learn $\geq \alpha$", wherein the learning objective targets the distribution of neighborhood knowledge with golden labels greater than or equal to a threshold, representing domain-relevant knowledge; "Learn $< \alpha$" represents the opposite scenario, targeting distributions where golden labels are below the threshold, representing domain-irrelevant knowledge; "Learn all" refers to using

Threshold	IT		Koran		Medical		Law		Avg.	
	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU
0.2	69.14	49.15	6.66	22.73	61.18	58.10	71.59	63.47	52.14	48.36
0.3	68.01	49.13	9.26	23.49	60.56	57.39	72.41	63.32	52.56	48.33
0.4	68.44	49.56	8.25	23.20	60.96	58.31	71.85	63.48	52.38	48.64
0.5	69.41	50.30	9.41	23.37	60.75	57.80	72.26	63.96	52.96	48.86
0.6	68.95	49.62	7.85	23.17	60.97	57.96	72.22	63.45	52.50	48.55
0.7	67.40	49.01	6.57	22.80	61.88	58.46	72.78	63.99	52.16	48.57
0.8	67.88	49.49	8.85	23.36	61.88	58.69	71.96	63.77	52.64	48.83

Table 5: Experimental effects of different α.

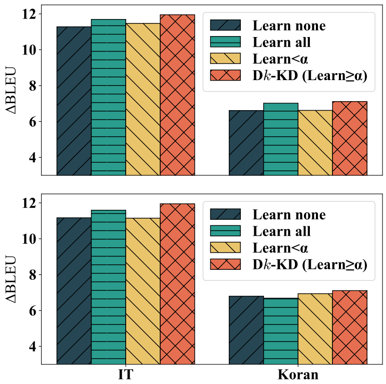

Figure 2: For Dk-KD in the context of two-step distillation, we conducted three comparative experiments for each step on the Koran and IT datasets, respectively. The bars in the histogram indicate the increase in BLEU scores relative to the baseline NMT model.

all retrieved neighbor knowledge kNN distributions for distillation without filtering, meaning all retrieved probability distributions are utilized for distillation; "Learn none" indicates that neighbor knowledge kNN distributions are not used for distillation. It can be intuitively observed that learning from domain-relevant knowledge can enhance the performance of the final model compared to learning from the complete set of neighborhood knowledge, while learning from domain-irrelevant knowledge can diminish the performance of the final model. Furthermore, the effectiveness and rationality of selectively filtering knowledge are highlighted by the improved performance of the final model after executing domain-aware knowledge selection, compared to indiscriminate learning.

In addition, the choice of different thresholds is also noteworthy, we conducted experiments on different α while fixing other parameters and the results of the tests are shown in Table 5. As can be seen by their average scores in the four domains, for both assessment metrics, all test results are higher than the current state-of-the-art methods. These experiments demonstrate that, despite the variability introduced by different threshold settings, the method consistently enhances performance compared to the state-of-the-art (namely, INK), indicating flexibility and robustness in threshold selection.

5.2 How Does Dk-KD Enhance Domain Translation?

Enhanced Domain-Aware Sentence Representation We input the test set sentences into the model, extract the outputs from the model's encoder, and perform average pooling. Then, by applying t-SNE for dimensionality reduction on the sentence representations, we obtain the clustering results as shown in Figure 3. The Koran dataset exhibits the best clustering effect, which may be attributed to the presence of a larger number of unique terms within the Koran domain. Conversely, in the clustering of the Law, Medical, and IT datasets, there is always a degree of overlap, likely due to the intersecting content across these domains. We also calculated the average intra-cluster distance for each method, and the results indicate that Dk-KD has the smallest average cluster distance, demonstrating the best clustering effect for distinguishing domain shifts. Given that our sentence representations are derived from the average pooling of individual word representations, the model's effectiveness in clustering sentence representations is directly related to its performance in representing words. Low-frequency words, due to their sparse occurrence, may have vector representations that are less accurate or robust, leading to suboptimal clustering performance.

Enhanced Rare Word Translation In multi-domain datasets, low-frequency words are often terminologies specific to their respective domains. We examined word frequencies of WMT19 German-English training sets across domains, classifying words as low-frequency (less than 50 occurrences),

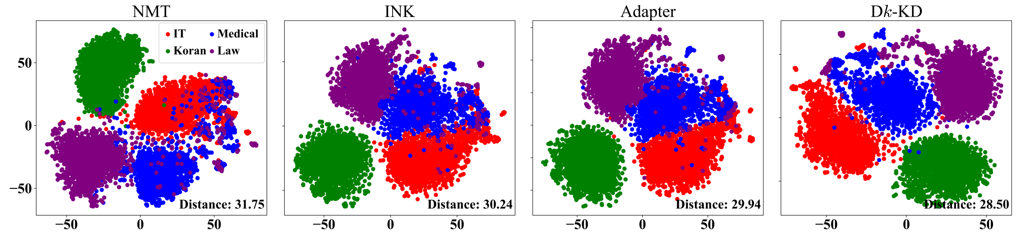

Figure 3: Visualization of sentence representations from test sets across four datasets for different systems.

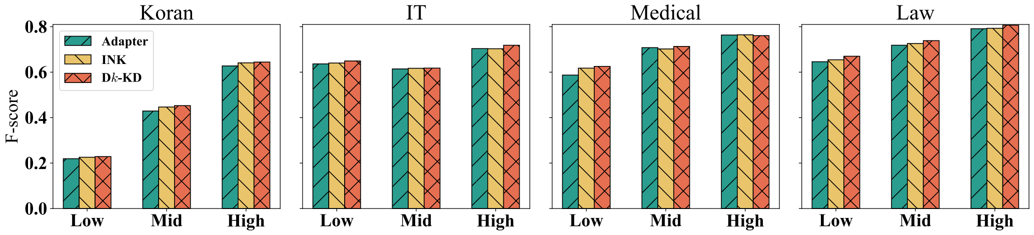

Figure 4: Comparing the translation performance of tokens with varying frequencies across four datasets.

medium-frequency (51-999 occurrences), and high-frequency (over 1000 occurrences). To evaluate our approach, we used a baseline that involves direct training of adapters and employed compare-mt (Neubig et al., 2019) to calculate F-scores, as shown in Figure 4. Our research demonstrates that, in comparison to the adapter baseline and INK, Dk-KD exhibits a significant improvement in the inference accuracy of low-frequency words across various domains. This improvement is particularly notable in the medical and law fields, where the enhancement in inference accuracy for low-frequency words exceeds that of medium and high-frequency words. We hypothesize that the improvement of translating low-frequency words is associated with the size of the datastore, suggesting that a larger and more comprehensive datastore facilitates better learning outcomes for low-frequency words. The robust performance of Dk-KD across different domains, especially the translation of low-frequency words, validates the efficacy of our approach.

5.3 Dk-KD with kNN Datastore Retrieval

Following the training process of Dk-KD, we constructed a new datastore using the representations from the final model and applied Robust kNN-MT to it. The resulting COMET and BLEU scores are shown in Table 6. We found that applying R-kNN to the Koran dataset actually deteriorates translation performance, which may be attributed to the smaller size of the Koran dataset leading to a sparser datastore, making it challenging to extract an effective kNN distribution to assist in generation. The sparse datastore for the Koran dataset indicates a need for further methodological adjustments or enhancements when dealing with limited or highly specialized data collections. Despite the challenges observed with the Koran dataset, the overall results illustrate that, in most scenarios, the combined application of Dk-KD with R-kNN still leads to an improvement in translation quality.

6 Related Work

6.1 Domain Adaptation for NMT

The prevailing approaches in this area can generally be categorized into model-centric and data-centric methods. The former focuses on carefully designing NMT model architectures to learn translation knowledge of target domain (Wang et al., 2017; Zeng et al., 2018; Bapna and Firat, 2019a; Guo et al., 2021) or improving the training process to better utilize relevant context (Wuebker et al., 2018; Bapna and Firat, 2019b; Lin et al., 2021; Liang et al., 2021). The latter, on the other hand, involves leveraging target domain monolingual corpora (Zhang and Zong, 2016; Zhang et al., 2018b), synthetic corpora (Hoang et al., 2018; Hu et al., 2019; Wei et al., 2020), parallel corpora (Chu et al., 2017), or learn the similar data various domains to intensify the model for low-resource machine translation (Zhan et al., 2021), to enhance machine translation models through model fine-tuning. In this paper, we propose a novel approach by starting with building a datastore of domain knowledge, extracting domain-relevant knowledge from stored representations.

Method	IT		Koran		Medical		Law	
	COMET	BLEU	COMET	BLEU	COMET	BLEU	COMET	BLEU
Dk-KD	69.41	50.30	**9.41**	**23.37**	60.75	57.80	72.26	63.96
R-kNN	58.10	48.62	2.07	19.65	57.54	57.27	69.82	63.79
Dk-KD+R-kNN	**69.69**	**50.75**	1.64	21.65	**60.83**	**59.23**	**73.22**	**65.79**

Table 6: Results of Dk-KD with kNN Decoding.

6.2 kNN-MT

Example-based approaches have been introduced into machine translation, demonstrating the utility of leveraging token-level or sentence-level examples to improve translation performance (Zhang et al., 2018a; Gu et al., 2018). kNN-MT significantly improves performance by retrieving examples, serving as a non-parametric retrieval-augmented method that offers an alternative to traditional fine-tuning methods (Khandelwal et al., 2020). Adaptive kNN-MT further enhances performance through the training of a network that dynamically adjusts parameters (Zheng et al., 2021a; Jiang et al., 2022). kNN-MT has shown promising progress in various machine translation tasks, including domain adaptation (Zheng et al., 2021b; Cao et al., 2023), interactive machine translation (Wang et al., 2022b), Transfer Learning (Li et al., 2022; Liu et al., 2023) and speech translation (Du et al., 2022). Other researchers have enhanced kNN-MT retrieval efficiency by pruning the datastore (Wang et al., 2022a), dynamically constructing the datastore (Meng et al., 2022; Wang et al., 2021; Dai et al., 2023), and reducing the number of steps required for retrieval (Martins et al., 2022a,b). To eliminate the need for retrieval during inference, the approach of distilling the knowledge from the datastore into the model parameters has been proposed, presenting a novel alternative to kNN retrieval (Yang et al., 2022; Zhu et al., 2023). We observe that not all the knowledge contained in a domain-specific datastore is beneficial for the learning of student models. In this paper, we present a two-step distillation process to extract domain-relevant knowledge from the neighboring datastore, effectively distilling the knowledge into the adapter.

7 Conclusion and Future Work

In this paper, we introduce a method named domain-aware kNN-KD, which extracts domain-relevant knowledge from a domain-aware datastore constructed by a domain-agnostic model. In Dk-KD, we design a two-stage knowledge distillation process that initially trains a domain-relevant teacher with the aid of domain-relevant knowledge, and then utilizes the teacher model to distill domain-relevant knowledge into the adapter layers, achieving improvements in translation performance. Notably, Dk-KD significantly enhances the source representation and the translation accuracy of domain-specific low-frequency words.

Future work includes: 1) Attempting to shift the filtering of domain-focused knowledge from the training phase to the datastore construction phase, aiming to reduce the storage footprint of the datastore and alleviate retrieval times; 2) Identifying more effective and efficient methods for filtering domain-relevant knowledge to enhance the efficacy of kNN knowledge distillation.

Limitations

While the effectiveness of our model is promising, employing the entire datastore for knowledge distillation during the training phase results in longer training times. Furthermore, we need to conduct two rounds of training for each domain, thereby increasing the overall training expense. Additionally, we observed that even with the final model, kNN knowledge continues to aid in inference performance. This suggests that our selection of domain-relevant knowledge is not sufficiently precise, or the distillation method may be unsuitable, indicating potential areas for improvement. Due to the large scale of large language models, current methods of kNN knowledge distillation, including Dk-KD, are not readily transferable to machine translation based on large language models, awaiting further development in GPU memory capacity.

Acknowledgment

This work was supported in part by the National Natural Science Foundation of China (Grant Nos. 62206076, 62306084, 62261160648), Guangdong Basic and Applied Basic Research Foundation (Grant No. 2024A1515011491), Shenzhen Science and Technology Program (Grant Nos. ZDSYS20230626091203008, KJZD20231023094700001, RCBS20221008093121053), Shenzhen College Stability Support Plan (Grant Nos. GXWD20220811173340003, GXWD20220817123150002), the Science and Technology Development Fund, Macau SAR (Grant Nos. FDCT/060/2022/AFJ, FDCT/0070/2022/AMJ), the Multi-year Research Grant from the University of Macau (Grant No. MYRG-GRG2023-00006-FST-UMDF), and Tencent AI Lab Rhino-Bird Gift Fund (Grant No. EF2023-00151-FST). Xuebo Liu was sponsored by CCF-Tencent Rhino-Bird Open Research Fund. We would like to thank the anonymous reviewers and meta-reviewer for their insightful suggestions.

References

Ankur Bapna and Orhan Firat. 2019a. Non-parametric adaptation for neural machine translation. 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 1921–1931, Minneapolis, Minnesota. Association for Computational Linguistics.

Ankur Bapna and Orhan Firat. 2019b. Simple, scalable adaptation for neural machine translation. 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 1538–1548, Hong Kong, China. Association for Computational Linguistics.

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. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5841–5853, Toronto, Canada. Association for Computational Linguistics.

Chenhui Chu, Raj Dabre, and Sadao Kurohashi. 2017. An empirical comparison of domain adaptation methods for neural machine translation. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 385–391, Vancouver, Canada. Association for Computational Linguistics.

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

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

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search engine guided neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5133–5140. AAAI Press.

Demi Guo, Alexander Rush, and Yoon Kim. 2021. Parameter-efficient transfer learning with diff pruning. 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 4884–4896, Online. Association for Computational Linguistics.

Geoffrey Hinton, Oriol Vinyals, and Jeff Dean. 2015. Distilling the knowledge in a neural network. *ArXiv preprint*, abs/1503.02531.

Vu Cong Duy Hoang, Philipp Koehn, Gholamreza Haffari, and Trevor Cohn. 2018. Iterative back-translation for neural machine translation. In *Proceedings of the 2nd Workshop on Neural Machine Translation and Generation*, pages 18–24, Melbourne, Australia. Association for Computational Linguistics.

Junjie Hu, Mengzhou Xia, Graham Neubig, and Jaime Carbonell. 2019. Domain adaptation of neural machine translation by lexicon induction. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 2989–3001, Florence, Italy. Association for Computational Linguistics.

Hui Jiang, Ziyao Lu, Fandong Meng, Chulun Zhou, Jie Zhou, Degen Huang, and Jinsong Su. 2022. Towards robust k-nearest-neighbor machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, December 7-11, 2022*, pages 5468–5477. Association for Computational Linguistics.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, April 26-30, 2020*. OpenReview.net.

Zhaocong Li, Xuebo Liu, Derek F. Wong, Lidia S. Chao, and Min Zhang. 2022. ConsistTL: Modeling consistency in transfer learning for low-resource neural machine translation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 8383–8394, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jianze Liang, Chengqi Zhao, Mingxuan Wang, Xipeng Qiu, and Lei Li. 2021. Finding sparse structures for domain specific neural machine translation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 13333–13342. AAAI Press.

Zehui Lin, Liwei Wu, Mingxuan Wang, and Lei Li. 2021. Learning language specific sub-network for multilingual 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 293–305, Online. 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 Martins, Zita Marinho, and Andre Martins. 2022a. Efficient machine translation domain adaptation. In *Proceedings of the 1st Workshop on Semiparametric Methods in NLP: Decoupling Logic from Knowledge*, pages 23–29, Dublin, Ireland and Online. Association for Computational Linguistics.

Pedro Henrique Martins, Zita Marinho, and André F. T. Martins. 2022b. 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.

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

Graham Neubig, Zi-Yi Dou, Junjie Hu, Paul Michel, Danish Pruthi, and Xinyi Wang. 2019. compare-mt: A tool for holistic comparison of language generation systems. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics (Demonstrations)*, pages 35–41, Minneapolis, Minnesota. 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 (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

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

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. COMET: A neural framework for MT evaluation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 2685–2702, Online. Association for Computational Linguistics.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

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

Dongqi Wang, Haoran Wei, Zhirui Zhang, Shujian Huang, Jun Xie, and Jiajun Chen. 2022b. Non-parametric online learning from human feedback for neural machine translation. In *Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22 - March 1, 2022*, pages 11431–11439. AAAI Press.

Rui Wang, Masao Utiyama, Lemao Liu, Kehai Chen, and Eiichiro Sumita. 2017. Instance weighting for neural machine translation domain adaptation. In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1482–1488, Copenhagen, Denmark. Association for Computational Linguistics.

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

Hao-Ran Wei, Zhirui Zhang, Boxing Chen, and Weihua Luo. 2020. Iterative domain-repaired back-translation. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 5884–5893, Online. Association for Computational Linguistics.

Joern Wuebker, Patrick Simianer, and John DeNero. 2018. Compact personalized models for neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 881–886, Brussels, Belgium. Association for Computational Linguistics.

Zhixian Yang, Renliang Sun, and Xiaojun Wan. 2022. Nearest neighbor knowledge distillation for neural machine translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5546–5556, Seattle, United States. Association for Computational Linguistics.

Jiali Zeng, Jinsong Su, Huating Wen, Yang Liu, Jun Xie, Yongjing Yin, and Jianqiang Zhao. 2018. Multi-domain neural machine translation with word-level domain context discrimination. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 447–457, Brussels, Belgium. Association for Computational Linguistics.

Runzhe Zhan, Xuebo Liu, Derek F. Wong, and Lidia S. Chao. 2021. Meta-curriculum learning for domain adaptation in neural machine translation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 14310–14318. AAAI Press.

Jiajun Zhang and Chengqing Zong. 2016. Exploiting source-side monolingual data in neural machine translation. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 1535–1545, Austin, Texas. Association for Computational Linguistics.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018a. Guiding neural machine translation with retrieved translation pieces. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 1325–1335, New Orleans, Louisiana. Association for Computational Linguistics.

Zhirui Zhang, Shujie Liu, Mu Li, Ming Zhou, and Enhong Chen. 2018b. Joint training for neural machine translation models with monolingual data. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 555–562. AAAI Press.

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

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

Wenhao Zhu, Jingjing Xu, Shujian Huang, Lingpeng Kong, and Jiajun Chen. 2023. INK: Injecting kNN knowledge in nearest neighbor machine translation. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 15948–15959, Toronto, Canada. Association for Computational Linguistics.

Wenhao Zhu, Qianfeng Zhao, Yunzhe Lv, Shujian Huang, Siheng Zhao, Sizhe Liu, and Jiajun Chen. 2024. kNN-BOX: A unified framework for nearest neighbor generation. In *Proceedings of the 18th Conference of the European Chapter of the Association for Computational Linguistics: System Demonstrations*, pages 10–17, St. Julians, Malta. Association for Computational Linguistics.