Simply Trainable Nearest Neighbour Machine Translation with GPU Inference

Anonymous ACL submission

Abstract

Nearest neighbor machine translation is a successful approach for fast domain adaption, which interpolates the pre-trained transformers with domain-specific token-level k-nearest-neighbor (kNN) retrieval without retraining. Despite kNN MT's success, searching large reference corpus and fixed interpolation between the kNN and pre-trained model led to computational complexity and translation quality challenges. In this paper, we propose a simply trainable nearest neighbor machine translation and carry out inference experiments on GPU. In specific, we first adaptively construct a small datastore for each input sentence. Second, we train the interpolation coefficient between the knnMT and pre-trained result to automatically interpolate in different domains. Experimental results on different domains show that our proposed method at least maintains the translation quality of other methods in the literature while being automatic. In addition, our inference results demonstrate that running knnMT is feasible on GPUs with only a 5% speed drop.

1 Introduction

Neural Machine Translation (NMT) has been showing an increasing trend of translation quality owing to the ongoing development of deep neural network models (Vaswani et al., 2017; Kim et al., 2021). However, the quality of these models is limited as soon as the domain of the input test sentences is different than the training data.

To handle this out-of-domain problem, k-nearest neighbor machine translation (kNN-MT) has proven to be successful in many studies (Khandelwal et al., 2021; Zheng et al., 2021a,b; Jiang et al., 2021; Wang et al., 2022; Meng et al., 2022), and thus piqued much attention in the community of machine translation. At the core of kNN-MT, a kNN classifier over an external datastore is built based on cached decoder representations and corresponding target tokens. This classifier is utilized to augment the given NMT model without finetuning leading to improved predictions, especially for domain adaption. Augmenting the NMT model is done via interpolating between the output probability distribution of the NMT model and the kNN classifier output probability distribution.

Despite kNN-MT's noticeable success in alleviating the domain adaption problem, vanilla kNN-MT proposed in (Khandelwal et al., 2021) mainly suffers two challenges slowing down kNN-MT's deployment. First, vanilla kNN-MT requires large datastore sizes resulting in massive storage and expensive latency overheads during inference. For example, (Khandelwal et al., 2021) showed that kNN-MT is two orders of magnitude slower than the base NMT system in a generation speed when retrieving 64 keys from a datastore containing billions of records. Second, the interpolation between the NMT model and the kNN classifier is fixed for all sentences in the test sets and manually tuned to improve translation quality.

Reviewing the literature, various techniques have been proposed to overcome kNN-MT's challenges. For example, (Meng et al., 2022) designed Fast kNN-MT where a subset of the large datastore is created for each source sentence by searching for the nearest token-level neighbors of the source tokens and mapping them to the corresponding target tokens. Building on (Meng et al., 2022), (Dai et al., 2023) proposed a simple and scalable kNN-MT that leverages current efficient text retrieval mechanisms, such as BM25 (Robertson et al., 2009), to obtain a small number of reference samples that have high semantic similarities with the input sentence, and then dynamically construct a tiny datastore by forwarding the samples to the pre-trained model. (Dai et al., 2023) successfully introduced a simple distance-aware interpolation equation to adaptively incorporate kNN retrieval results into the NMT model. However, this simple equation required manual tuning. Along the same line, (Jiang et al., 2022) proposed a trainable interpolation method but with a relatively complicated neural network. To the best of our knowledge, these papers did not integrate kNN-MT into GPU inference to observe the trade-off between accuracy and speed results.

Towards kNN-MT challenges, this paper proposes a simply trainable nearest neighbor machine translation and demonstrates kNN feasibility with GPU Inference. Similar to (Dai et al., 2023), we reduce the large datastore size by extracting online a small number of reference samples that have high semantic similarities with the input test sentence using the efficient BM25 retrieval algorithm (Robertson et al., 2009). Based on these insights, we propose a simply trainable neural network that adaptively interpolates the NMT and knnMT probability distributions per domain in an average of 40 minutes of a single GPU training time. Last but not least, we integrate kNN-MT into FasterTransformer, a highly optimized NMT GPU inference implementation offered by NVIDIA, and observe its speed and accuracy performance on a sparsely activated large-scale MoE. Experimental results show the translation quality effectiveness of our adaptive and automatic interpolation technique and insignificant speed drop of knnMT on GPU.

2 Background: kNN-MT

2.1 Vanilla-kNN

In Vanilla-kNN, a datastore is created to convert a bilingual sentence into a set of key-value pairs. These keys and values are defined in Equation 1.

$$K, V = F(x, y_{<t}), y_t \tag{1}$$

where (x, y) ∈ (X, Y) define the reference corpus for which the pretrained NMT model generates the context representation $F(x, y_{<t}$ at each time step t. Then we collect the output hidden state $F(x, y_{<t})$ as key and y_t as value to construct the whole datastore (K, V).

At inference time, the current context representation $F(x, \hat{y}_{<t})$ at decoding step t, as well as the already generated words, are leveraged to generate a retrieval distribution $P_{knn}(y_t|y_{<t}, x)$ over the entire vocabulary:

$$P_{knn}(y_t|x, \hat{y}_{<t}) \propto \sum_{(h_i,v_i)\in N_t} I_{y_t=v_i} exp(\frac{-L_2(hi, F(x, \hat{y}_{<t}))}{T}) \tag{2}$$

where L2 is the Euclidean distance between the current context embedding and the embedding of a token from the data store. In vanilla KNN-MT, a predefined interpolation weight λ is fixed as a hyperparameter. This weight interpolates between the probability distribution computed from KNN and the probability distribution generated from the pretrained NMT model (see Equation 3).

$$\begin{aligned} P(y_t|x, \hat{y}_{<t}) &= \lambda * P_{mt}(y_t|x, \hat{y}_{<t}) \\ &+ (1-\lambda) * P_{knn}(y_t|x, \hat{y}_{<t}) \end{aligned} \tag{3}$$

2.2 SK-MT

In SK-MT (Dai et al., 2023), Elasticsearch is used for semantic retrieval components to create a sentence adaptive datastore instead of a static and extensive datastore used in Vanilla kNN-MT. In specific, Elasticsearch does two main operations: Index & Search; storing parallel sentences in indexes format, and then retrieving 32 sentences per input sentence with the highest relevance score from the training corpus.

Also, SK-MT provided a successful way of setting the interpolation coefficient in Equation 4.

$$\lambda = Relu(1 - \frac{d_0}{T}) \tag{4}$$

where d_0 is the top-1 L2 distance during the nearest neighbor search, T is the temperature parameter and is typically fixed.

3 Inference with Trainable *k*NN Retrieval

3.1 Trainable Interpolation

Even though Sk-MT introduced a simple solution that derives the interpolation weight from the distance, a fixed parameter T for all datasets is tuned to produce the best results. A fixed temperature may not be optimal for all domains and datasets. For example, Figure 1 shows the BLEU score from the Koran dataset when varying T from 100 to 500 with a step size of 100. As seen in the figure, $T = 300$ provides the best BLEU score and the optimal value varies with the dataset. This observation motivates a simple and trainable method to find the optimal temperature for each dataset.

The proposed simple neural network consists of a single layer trained to predict the interpolation weight given the distance of the retrieved kNN candidates. This is in contrast to other adaptive interpolation methods e.g. (Jiang et al., 2022) which use more complex architectures and are harder to train. We use the development set of each domain to optimize our single-layer network.

Our training objective is designed to provide better translation quality. Knowing the ground truth token, we can choose the best interpolation weight that produces the best probability distribution that we can get from the interpolation between P_{mt} and P_{knn}. Thus, our final objective is to create a sharper final probability distribution toward our ground truth token.

Algorithm 1 Training Interpolation Layer

$len \leftarrow length(y)$
$t \leftarrow 0$
while $t \neq len$ **do**
  $gt \leftarrow$ ground truth Index
  generate $Pknn(y_{<t}, x)$
  generate $Pmt(y_{<t}, x)$
  if $Pknn(gt|y_{<t}, x) \geq Pmt(gt|y_{<t}, x)$ **then**
    $label = 1$ ▷ favoring Pknn
  else
    $label = 0$
  end if
  $\lambda_{pred} = Sigmoid(W * D_0 + B)$
  $loss \leftarrow CrossEntropy(\lambda_{pred}, label)$
  update W, B
  $t = t + 1$
end while

As shown in Algorithm 1, our training procedure is divided into two stages at each decoding step. The first stage examines the probability of the ground truth token in both distributions P_{knn} and P_{mt}. If the probability of the ground truth token P_{knn} is higher then we set the label to 1 otherwise we set the label to 0. The second stage trains our single-layer network using binary loss.

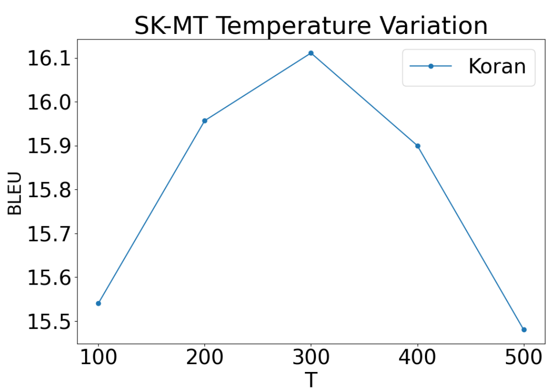

Figure 1: Koran Temperature Variation.

4 Experimental Results

4.1 Experimental Setup

Input stimuli and Datasets: We test our methodology in 2 language directions: German-English (deen), and English-Czech (encs). For deen, we employ the multi-domain dataset as the baseline (Khandelwal et al., 2021) in addition to an e-commerce domain. For encs, we utilize two other domains: finance and medpharma. Our evaluation metrics are the SacreBLEU (Post, 2018) and COMET-22 (wmt22-COMET-da) (Rei et al., 2022), a reference-based metric that combines direct assessments (DA), sentence-level scores, and word-level tags from Multidimensional Quality Metrics (MQM) error annotations.

Models: Three transformer models are used in our experiments. The first two of the three are used to measure the translation quality. The first two transformer models are constructed from 12 encoder layers and 12 decoder layers with 512 hidden dimensions and 2048 feedforward layer hidden dimensions with 8 multi-head attention heads. The third transformer is the ZCode M3 model reviewed and presented in (Kim et al., 2021). ZCode M3 is constructed from 24 encoder layers and 12 decoder layers with 1024 hidden dimensions and 4096 feedforward layer hidden dimensions with 16 multi-head attention heads. The ZCode M3 has 32 experts, 5B parameters, and 128,000 vocab size.

Baselines: The model without knnMT is one baseline. We also compare with the SK-MT method that uses a distance-aware adapter (Dai et al., 2023). In (Dai et al., 2023), the authors compared with other methods and showed success so we use (Dai et al., 2023) as a proxy to compare with other methods.

GPU Inference Hardware and Environment: Inference and speed evaluation experiments are carried out on a single NVIDIA Tesla V100 GPU. Our inference environment is the highly optimized FasterTransformer from NVIDIA. Without loss of generality, we utilize the SK-MT with its fixed distance-aware adapter in these experiments to measure speed. Because experimental results between SK-MT and the proposed trainable method show similar translation quality performance, the speed numbers should not change.

4.2 Trainable *k*NN Retrieval Results

Table 1 shows the translation quality performance comparison between the proposed trainable method and other baselines. As shown in the table, our pro-

Table 1: Translation quality of the proposed method versus other methods at Beam=5 and K=2.

Domain	**BLEU**			**WMT22-COMET-da**		
	NMT	**SK-MT**	**Trainable**	**NMT**	**SK-MT**	**Trainable**
IT	38	45.5	46.1	83.0	85.0	85.0
Law	49.6	62.8	62.7	86.7	88.3	88.0
Koran	12.2	15.5	16.4	69.1	70.0	70.6
Medical	42.7	57.1	51.0	83.9	85.2	85.0
e-commerce	52.5	58.1	58.5	90.7	90.9	90.9
finance	48.6	53.3	53.3	70.6	94.2	93.9
medpharma	41.6	47.4	45.5	92.2	92.0	92.5
AVERAGE	40.8	48.5	47.6	82.4	86.5	86.6

Table 2: GPU Inference Results on ZCode M3 Model.

Domain	**beam=1, batch=1**			**beam=2, batch=20**		
	BLEU		**Speed Drop (%)**	**BLEU**		**Speed Drop (%)**
	NMT	SK-MT		NMT	SK-MT	
IT	37.6	43.8	4.9	37.4	43.7	6.5
Medical	45.6	55.6	5.0	45.8	56.3	9.1
Law	54.1	61.8	5.8	54.1	62.2	5.5
AVERAGE	**45.7**	**53.7**	**5.2**	**45.7**	**54.0**	**7.0**

posed trainable method improves the NMT baseline translation quality by a large margin. In addition, the proposed method at least maintains the translation quality relative to SK-MT on average in terms of the BLEU and COMET scores except for the Medical and medpharma domains. This result demonstrates the ability to at least maintain the performance of SK-MT while being trainable in a simple manner. For Medical and medpharma, SK-MT outperforms our proposed method because the datastore built by the dev set does not have any semantic similarity to the training set leading to imbalanced binary labeling, whereas the test does not have this imbalanced binary labeling. To overcome this challenge, we suggest that we add weights to the binary cross-entropy training loss function. With this weighted loss function, our trainable method achieves 57.2 BLEU, 85 COMET in Medical, and 48.1 BLEU, 92.5 COMET in medphrama. These results increase our average results to 48.9 BLEU, and 86.6 COMET, respectively.

Table 1 shows that the translation quality in terms of COMET using SK-MT or our proposed method is not always significantly improved as noticed in BLEU. For example, e-commerce has an improvement of roughly 6 BLEU points relative to NMT, while the improvement is 0.3 COMET score points. For the domain problem, we believe this phenomenon exists in some domains because COMET could focus slightly more on fluency, while BLEU could focus more on adequacy and text completeness.

4.3 GPU Inference Results

Table 2 depicts the speed results of ZCode M3 inference and corresponding BLEU scores in three domains under test namely, IT, Medical, and Law. The results for beam=1, batch=1 SK-MT setting on the large scale MoE improves the NMT baseline with a large margin while dropping the speed by only 5.2% on average. Similarly, SK-MT has an improved translation quality with only a drop of 7% relative to NMT as beam and batch increase to 2 and 20, respectively. These results show the potential of deploying the knnMT domain adaption approach in such a large-scale model as ZCode M3.

Conclusion

This paper proposes a simply trainable nearest-neighbor machine translation and carries out experiments on large-scale models to demonstrate kNN feasibility with GPU Inference. Experimental results show the translation quality effectiveness of our adaptive and automatic interpolation technique and insignificant speed drop of knnMT on GPU inference.

References

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 (ICLR)*.

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*, pages 5468–5477, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Qingnan Jiang, Mingxuan Wang, Jun Cao, Shanbo Cheng, Shujian Huang, and Lei Li. 2021. Learning kernel-smoothed machine translation with retrieved examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7280–7290, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations (ICLR)*.

Young Jin Kim, Ammar Ahmad Awan, Alexandre Muzio, Andres Felipe Cruz Salinas, Liyang Lu, Amr Hendy, Samyam Rajbhandari, Yuxiong He, and Hany Hassan Awadalla. 2021. Scalable and efficient moe training for multitask multilingual models. *arXiv preprint arXiv:2109.10465*.

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.

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

Ricardo Rei, José GC De Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova, Alon Lavie, Luisa Coheur, and André FT Martins. 2022. Comet-22: Unbabel-ist 2022 submission for the metrics shared task. In *Proceedings of the Seventh Conference on Machine Translation (WMT)*, pages 578–585.

Stephen Robertson, Hugo Zaragoza, et al. 2009. The probabilistic relevance framework: Bm25 and beyond. *Foundations and Trends® in Information Retrieval*, 3(4):333–389.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need.

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.

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.