Enhanced Retrieve-Edit-Rerank Framework with kNN-MT

Xiaotian Wang[1] **Takuya Tamura**[1] **Takehito Utsuro**[1] **Masaaki Nagata**[2]
[1]Deg. Prog. Sys.&Inf. Eng., Grad. Sch. Sci.&Tech., University of Tsukuba
[2]NTT Communication Science Laboratories, NTT Corporation, Japan
[1]{s2320811, s2120744}@s.tsukuba.ac.jp
[1]utsuro@iit.tsukuba.ac.jp
[2]masaaki.nagata@ntt.com

Abstract

The similar translation sentences has been widely employed in machine translation task (MT), such as the NFR model (Bulte and Tezcan, 2019) and the Retrieve-Edit-Rerank framework (Hossain et al., 2020; Tamura et al., 2023). The source sentences concatenated with its similar translation sentences are used as the input for training the NFR model with the aim of leveraging the attention mechanism to extract beneficial information from these similar translations. However, the retrieved similar translations also contain noise, which may lead to decrease in translation accuracy. In this paper, we introduce kNN Machine Translation (kNN-MT) (Khandelwal et al., 2021; Zheng et al., 2021; Jiang et al., 2022) methods to address the issue caused by retrieved noisy similar translations. We show that kNN-MT methods significantly improve the overall translation accuracy of the framework. Meanwhile, we enhance the Retrieve-Edit-Rerank framework by incorporating the COMET-QE score (Rei et al., 2020, 2021) into the reranking function proposed by Tamura et al. (2023). Our proposed metric achieves the highest accuracy compared to all the previous studies when evaluated using sacreBLEU (Post, 2018). Furthermore, it demonstrates a significant improvement compared to prior research when evaluated using COMET22 (Rei et al., 2022).

1 Introduction

Translation Memory (TM), which is a set that contains high-quality parallel data, has been frequently utilized in recent research. The NFR (Neural Fuzzy Repair) model proposed by Bulte and Tezcan (2019) employs a method where the source sentence retrieves fuzzy matches from the TM and concatenates them as input to the neural machine translation (NMT) model, aiming to enhance the accuracy of machine translation task.

Based on the original NFR model, Hossain et al. (2020) proposed the Retrieve-Edit-Rerank framework, which generates translation candidates by inputting the same source sentence concatenated with different similar target sentences into a trained NFR model, followed by reranking with the objective of maximizing the log-likelihood. Tamura et al. (2023) further improved the Retrieve-Edit-Rerank framework from retrieval and reranking steps. In addition to the existing Python library edit-distance + $SetSimilaritySearch$[1] (ed+sss) method for retrieval, they incorporated mSBERT (Reimers and Gurevych, 2020) and LaBSE (Feng et al., 2022) to generate sentence embeddings, and then performed cross-language retrieval using Faiss[2] (Johnson et al., 2019). Moreover, Tamura et al. (2023) improved the reranking score by considering log-likelihood of output with normalization by token-level sentence length and the similarity between the input and candidate sentences based on sentence embeddings. However, the characteristic of semantic retrieval of LaBSE+Faiss and mSBERT+Faiss introduces a challenge, where the retrieved similar target sentences may not align with the source sentence at the token level, which could potentially have a negative impact when these similar translations are used for translation task.

To tackle this problem, we introduce kNN-MT (Khandelwal et al., 2021; Zheng et al., 2021; Jiang et al., 2022) methods into the editing phase of Retrieve-Edit-Rerank framework. In kNN-MT, the final hidden representations of predicted tokens for all the training data generated by decoder of the trained NMT model (Vaswani et al., 2017) are stored as ground truth tokens' hidden representations into a datastore. When incorporating the kNN-MT method into the NFR model, during the process of creating datastore, although the hidden representation of predicted tokens may deviate from their ground truth values due to the influence of similar translations, they are still recorded as

[1]https://github.com/ardate/SetSimilaritySearch
[2]https://github.com/facebookresearch/faiss

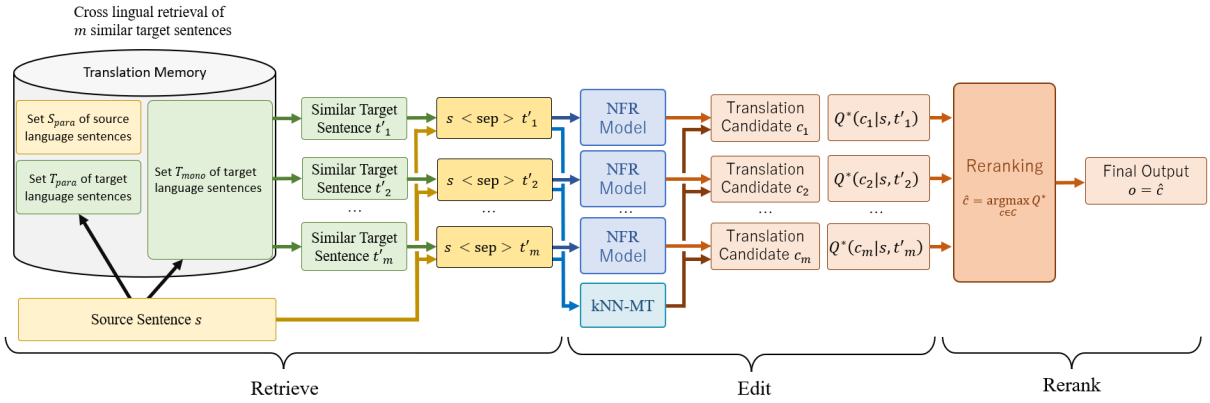

Figure 1: The prediction framework of Retrieve-Edit-Rerank model.

ground truth hidden representations and stored in the datastore. Therefore, during the inference stage, when the hidden representation of tokens is affected by noise from similar translations, it becomes possible to retrieve the ground truth token depending on the deviated representations in the datastore. In addition, the kNN-MT method considers the output distributions of both kNN-MT and the original NMT model, which possibly reduces the negative impact of similar translations acting as noise. In addition to that, this paper introduces COMET-QE score (Rei et al., 2020, 2021) into the reranking function proposed by Tamura et al. (2023), and the coefficients of various indicators are fine-tuned on the validation data.

In summary, our contributions are as below:

1. In the Retrieve-Edit-Rerank framework (Hossain et al., 2020) depicted in Figure 1, during the editing phase, kNN-MT methods (Khandelwal et al., 2021; Zheng et al., 2021; Jiang et al., 2022) are incorporated to slightly elevate the overall accuracy of the framework, while addressing the issue of noise caused by the retrieved similar translations.

2. In the reranking step, we augment the existing reranking function proposed by Tamura et al. (2023) with COMET21-QE (Rei et al., 2021), a metric that enables direct evaluation of candidate quality without reference. Our proposed metric achieves the highest accuracy compared to all previous studies when evaluated using sacreBLEU (Post, 2018). Furthermore, it demonstrates a significant improvement compared to prior research when evaluated using COMET22 (Rei et al., 2022).

2 Related Work

Retrieval-based approaches to NMT have achieved noticeable performance in recent years in machine translation task via providing additional information to the source sentence through retrieving fuzzy matches in Translation Memory (TM). Bulte and Tezcan (2019) introduced a strategy by directly concatenating the entire fuzzy matches alongside the input, leveraging the attention mechanism to extract effective information from similar translations. However, aforementioned approaches rely on searching for similar translations within the bilingual corpus, retrieving in the same language as the input. Cai et al. (2021) presented a method that uses a monolingual corpus in the target language, and put forward a learnable cross-lingual retrieval model which is simultaneously optimized with the NMT model. Hossain et al. (2020) proposed the Retrieve-Edit-Rerank framework, which generate translation candidates via inputting sentences concatenated with multiple similar translations into the trained NFR model and then rerank them based on log-likelihood.

Recent research has expanded beyond word embedding at the token level, prompting exploration to sentence embedding (Reimers and Gurevych, 2019, 2020; Feng et al., 2022; Mao and Nakagawa, 2023), which considers sentences as the fundamental unit of analysis. Reimers and Gurevych (2019) introduced Sentence-BERT (SBERT) and Sentence-RoBERTa (SRoBERTa), based on BERT (Devlin et al., 2019) and RoBERTa (Liu et al., 2019), which utilized Siamese and triplet networks (Schroff et al., 2015) for fine-tuning. The following year, they also released mSBERT (Reimers and Gurevych, 2020), a multilingual version of SBERT. Feng et al.

(2022) proposed LaBSE model, which efficiently leverages negative examples during training and achieves the state-of-art performance in the field of sentence embedding. Mao and Nakagawa (2023) proposed a lightweight version of the LaBSE model through knowledge distillation, which significantly reduces the number of the model's parameters at the cost of a slight decrease in accuracy. Tamura et al. (2023) incorporated sentence embedding into Retrieve-Edit-Rerank framework for cross-lingual retrieval in monolingual corpus, and performed reranking via a more efficacious reranking function.

Furthermore, in recent years, kNN-MT has shown a high potential in the realm of machine translation tasks. Initially proposed by Khandelwal et al. (2021), the kNN-MT approach entails storing representations of all output tokens from the training data as ground truth tokens in a datastore. In the decoding process, the context representation obtained from the NMT model is used to query the datastore to retrieve k nearest neighbors and form the kNN-MT output distribution. Zheng et al. (2021) introduced an enhanced form of the kNN-MT architecture known as Adaptive kNN-MT, which incorporates a set of possible values for k that are constrained within an upper bound K, as opposed to a fixed value for k. The Robust kNN-MT model, proposed by Jiang et al. (2022), is trained with two networks: one utilizes the output distribution of NMT to refine the output distribution of kNN-MT, while the other adjusts the weights of the two output distributions in the final output distribution.

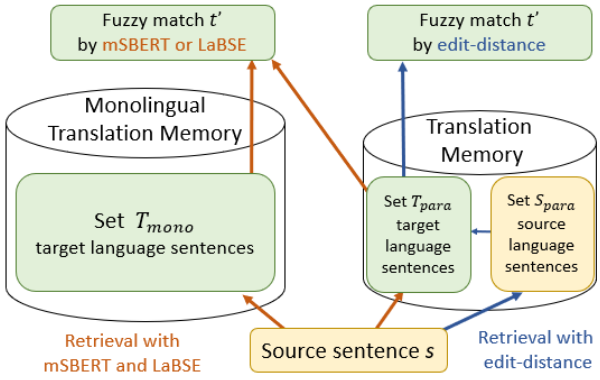

Figure 2: Retrieving Fuzzy Matches from Translation Memory

3 Retrieval

3.1 Translation Memory

Retrieval-based models demonstrate great potential in recent research, using the method of searching for the relevant target sentences in a massive size of high-quality parallel sentence pairs, which have been manually translated in the past, known as Translation Memory (TM). In **C**omputer-**A**ided **T**ranslation (CAT) workflows, when a source language sentence cannot be retrieved with a high fuzzy match surpassing a certain threshold, machine translation (MT) is employed as a backoff mechanism for generating the output. However, even in cases where direct retrieval from TM is not possible, fuzzy matches with high similarity can still be beneficial for MT process. In this paper, we define the target language sentence which is similar to the source language sentence as "similar target sentence or similar translation".

As shown in Figure 2, we define the Translation Memory TM_{para} as a parallel corpus consisting of source sentences set S_{para} and target sentences set T_{para}, As for a monolingual corpus that solely consists of target language sentences set T_{mono}, it is defined as TM_{mono}.

3.2 Retrieval based on Sparse Representation

Based on lexical similarity metrics, representative ones include BM25, tf-idf, and edit-distance (Levenshtein, 1966). Edit-distance, defined as $\Delta_{ed}(x, y)$, considers the operational steps required to transform sentence x into sentence y, including deletion, insertion, and substitution. The common characteristic of lexical similarity metrics is limited to retrieval within the language that is the same as the source sentence, and the computation costs are the overhead when used for retrieval in large-scale TM. Furthermore, research by Reimers and Gurevych (2021) suggested that the performance of lexical similarity metrics as sparse representation for retrieval is inferior to retrieval based on dense representation generated by Distil-RoBERTa (Sanh et al., 2019) trained with negative examples.

Therefore, in comparison to cross-lingual retrieval using dense representation, we only consider edit-distance (denoted as "ed" hereafter in this paper) as a representative lexical similarity metric, incorporating the similarity measure proposed by Vanallemeersch and Vandeghinste (2015) that takes into account the length of sentence x and

sentence y.

$$sim(x, y) = 1 - \frac{\Delta_{ed}(x, y)}{\max(|x|, |y|)}$$

where $\Delta_{ed}(x, y)$ is the edit-distance between two sentences x, y, and $|x|$ is the token-level length of sentence x. Moreover, in light of the computational explosion triggered by the computation of edit-distance for all the sentences contained within TM_{para}, we adapt the retrieval approach of Bulte and Tezcan (2019), which only calculates edit-distance for candidate sets of analogous sentences obtained through the utilization of the $containment_{max}$ similarity measure, facilitated by the Python library $SetSimilaritySearch$ (sss):

$$containment_{max}(v_x, v_y) = \frac{||v_x \cap v_y||}{\max(||v_x||, ||v_y||)}$$

where v_x and v_y are defined as the sets of unique tokens contained in the source sentences x and y respectively.

3.3 Retrieval based on Dense Representation

The application of dense representation for retrieval refers to the utilization of a pre-trained Language Model (LM) (Devlin et al., 2018; Reimers and Gurevych, 2020; Feng et al., 2022; Mao and Nakagawa, 2023) to embed sentences from multiple languages, and subsequently, these sentences are mapped to a densely shared vector space, followed by the retrieval using the source sentence x within this vector space. Therefore, when leveraging dense representation for retrieval, it can be utilized without the confines of a single language but extended to encompass diverse languages.

Among several cross-lingual sentence embedding Language Models (LMs) frequently employed in recent researches, LaBSE (Feng et al., 2022) model, which efficiently leverages negative examples during training, and performs state-of-art accuracy in the field of cross-lingual retrieval, is used in our research.

In this paper, LaBSE is introduced for cross-lingual retrieval, and the similarity used between the sentence x and the sentence y is cosine similarity:

$$sim(x, y) = \frac{Emb(x) \cdot Emb(y)}{|Emb(x)||Emb(y)|}$$

where $Emb(x)$ is the sentence embedding for the sentence x, and $|Emb(x)|$ denotes its magnitude.

Regarding the retrieval component, we utilize Faiss (Johnson et al., 2019) , a library for nearest neighbor search in dense space, to extract the similar translations with top-k similarity scores across languages in TM_{mono}.

4 Edit by kNN-MT

This section introduces edit procedures by kNN-MT methods, while their detailed explanations are presented in Section A.

4.1 kNN-MT

Vanilla kNN-MT (Khandelwal et al., 2021) is the initially proposed Nearest Neighbor Machine Translation (kNN-MT), which is a retrieval-based machine translation method. Moreover, the entire process of kNN-MT is independent of the NMT model, indicating that a pre-trained NMT model does not require further training when applying the kNN-MT method. The kNN-MT method consists of two steps. The first step involves establishing a datastore based on the training data of the NMT model. The second one entails using the hidden representation of the predicted token by the NMT model as a query to retrieve k nearest neighbors from the datastore to form the kNN-MT output distribution. Finally, output distribution of the NMT model is combined with the kNN-MT output distribution using the weight λ.

Robust kNN-MT (Jiang et al., 2022) is two trainable networks, where one network is employed to adjust the output distribution of kNN-MT method based on the output distribution of the NMT model, while the other network is used to balance the weight between the kNN-MT method and the NMT model. The Robust kNN-MT method has achieved significant success in handling noise in the datastore.

4.2 Candidates Generation with kNN-MT

Preprocess. The complete process of the Retrieve-Edit-Rerank framework is illustrated in Figure 1. First, we train the NFR model as the n-best framework used by Bulte and Tezcan (2019). Specifically, for given source sentence s, we extract n-best similar translations $t'_1, t'_2, \ldots, t'_n$ in TM based on ed or LaBSE, and concatenate the source sentence s with each corresponding similar translation sentence t'_k respectively using the special token "$\langle$sep$\rangle$" (i.e., "$s\langle$sep$\rangle t'_i$") as the input

Source Sentence	This paper examines a method to divide the images into regions where the statistical properties are resemble and a coding method which describes the region as a previous step of the regional division encoding.
Noisy Similar Translation (NST)	原画の局所領域を特異値分解して得た固有ベクトルをその領域の構造情報とみなし，構造情報の比較によりカテゴリーに分類する手法を提案した。 (The eigenvector obtained by singular value decomposition of a local region in the original image is regarded as the structural information of the region, and a method is proposed to classify the region into categories by comparing the structural information.)
Reference Sentence	ここでは，領域分割符号化の前段階として，画像を統計的性質の似た領域に分割する手法とその領域を記述する符号化手法を検討した。

	Output Sentences	Sentence-BLEU
Benchmark Translation by NFR w/o NST	まず，統計的特徴を記述する領域と，類似の領域を記述するコーディングと，地域区分符号化の前段階として記述する方法について述べた。 (First, the regions for describing statistical properties, the coding for describing similar regions, and a method for describing them as a previous step of the regional division encoding are described.)	19.45
Output Translation by NFR w/NST	統計的特徴が類似している地域，及び地域熱心化の既報の領域として記述される符号化方法について考察した。 (Regions with similar statistical properties and a coding method that can be described as the previously reported region of regional eagerness are discussed.)	1.59e-78
Output Translation by NFR+Vanilla kNN-MT w/NST	統計的特徴が類似している地域に分ける像と，地域区分の前段階として記述される領域を記述する符号化方法について考察した。 (Images that are divided into regions with similar statistical properties and a coding method which describes the regions which is described as a previous step of the regional division encoding are discussed.)	20.08
Output Translation by NFR+Robust kNN-MT w/NST	統計的特徴が似た地域に分けるべき画像と，地域分割符号化の前段階としてその領域を記述する符号化方法について考察した。 (Images that should be divided into regions with resemble statistical properties and a coding method which describes the regions as a previous step of the regional division encoding are discussed.)	**31.16**

Table 1: Examples of the Noisy Similar Translation and the results edited by kNN-MT methods (For ASPEC, discussions on the underlined portions "統計的特徴が類似している地域" (regions with similar statistical properties) are given in section 4.2.
)

format for training the NFR model. For each of the trained NFR models with ed or LaBSE, and the NMT model (the Vanilla Transformer (Vaswani et al., 2017)), we create a datastore based on the training data, and apply the three kNN-MT methods introduced in Section 4.1 to the validation data for determining the optimal hyperparameters of Vanilla kNN-MT as well as obtaining the networks of Robust kNN-MT.

Edit. First, for each retrieval method, we designate the output sentence obtained by directly inputting the source sentence s which is not concatenated with any similar translation into the trained NFR model as the "benchmark translation" $o_*^{w/o\,sim}$ for s, where $*$ represents a retrieval method. Subsequently, during the inference stage, we define the translation obtained by inputting s concatenated with its k-th most similar translation t'_k into the NFR model as "output translation" o_*^{w/sim_k}. For determining whether the retrieved similar translation is noise, we evaluate both benchmark translation $o_*^{w/o\,sim}$ and output translation o_*^{w/sim_k} referring to the reference translation o^{ref}. In cases where the evaluation value of o_*^{w/sim_k} is lower than $o_*^{w/o\,sim}$, the corresponding similar translation is defined as a "**N**oisy **S**imilar **T**ranslation (NST)" in this paper. For a given source sentence s_i, we concatenate it with k-th most similar translation "$t'_{(*,i),k}$" $(k = 1, \ldots, N)$ out of the top-N most similar translations and input them into the NFR model for obtaining output translations $o_{(*,i)}^{w/sim_1}, o_{(*,i)}^{w/sim_2}, \ldots, o_{(*,i)}^{w/sim_N}$. The proportion of NST is then calculated as "**R**ate of **N**oise (RoN)", which represents the percentage of these N output translations that show a decrease in accuracy when compared to the benchmark translation $o_{(*,i)}^{w/o\,sim}$.

$$RoN_*(s_i) = \sum_{k=1}^{N} \mathbb{1}_{QE(o_{(*,i)}^{w/sim_k}) < QE(o_{(*,i)}^{w/o\,sim})} / N$$

Here, $QE(\cdot)$ represents the quality evaluation by Sentence-BLEU or COMET22, $*$ represents ed or LaBSE, and in this paper, $N = 32$ is used. Finally, for the test data consisting of n sentence pairs, we calculate the "**M**ean **R**ate of **N**oise (MRoN)" and compare them across retrieval methods.

$$MRoN_* = \sum_{i=1}^{n} \frac{RoN_*(s_i)}{n}$$

Table 1 shows a concrete example of the noisy similar translation, and the results edited by kNN-MT methods. For the output translation of the "NFR w/NST" model, it can be observed that the underlined "統計的特徴が類似している地域" (regions with similar statistical properties) portion is the same as the output translations edited by Vanilla kNN-MT. However, thereafter, discrepancies emerge, leading to a significant decrease in translation accuracy of the original NFR model. This phenomenon can be attributed to the impact of similar translation acting as noise for the original NFR model. Nevertheless, the decoding process is guided back on the right track after the application of kNN-MT methods, and we show this process in Figure 3. During the ninth step of decoding, the NFR model predicts a higher probability for

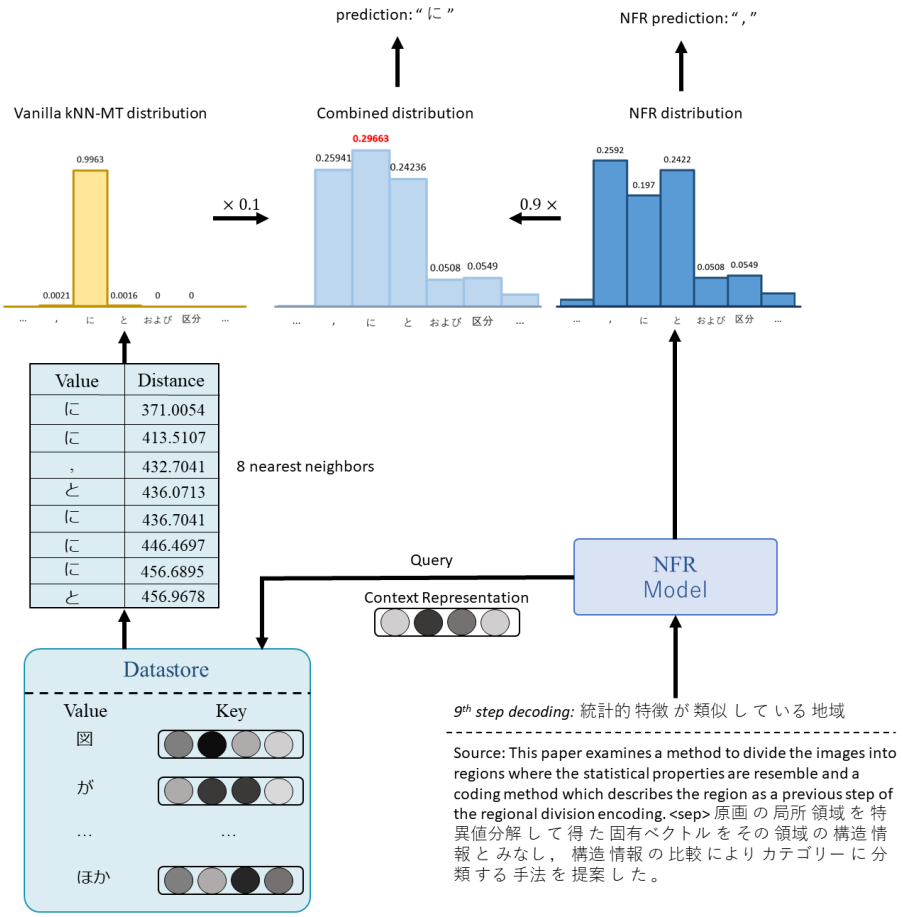

Figure 3: Decoding Process of Vanilla kNN-MT

"," than for "に". However, upon incorporating the Vanilla kNN-MT method to form the final output distribution, the final output probability of "に" becomes larger than ",", which may lead the subsequent decoding process towards a more accurate track.

Generation. In this paper, for any given source sentence s in the test data, we retrieve m similar translations $t'_1, t'_2, \ldots, t'_m$ in the TM using two different strategies, searching on sparse representation or dense representation, and then concatenate s and t'_k $(k = 1, \ldots, m)$ as in Figure 1 respectively and input into the trained NFR model, with the edit of kNN-MT methods for m times decoding to gain the candidates $c_1, c_2, \ldots, c_m$, and their adjusted output probability, preparing for the reranking step.

5 Rerank Candidates by Proposed Reranking Functions

During the editing stage, we input the source sentence s concatenated with m similar translations $t'_1, t'_2, \ldots, t'_m$ respectively, to generate m translation candidates $c_1, c_2, \ldots, c_m$, and we define the set of translation candidates as $\mathcal{C}$. In the reranking step, the main objective is to obtain a final output $\hat{c}$ from these m translation candidates by maximizing the score function Q^*.

$$\hat{c} = \arg\max_{c \in \mathcal{C}} Q^*(s, t', c)$$

In this paper, we introduce the following five reranking functions:

- **Q$^{\text{(Hossain)}}$** (Hossain et al., 2020), computed based solely on the log-likelihood of the output probabilities.

$$Q^{(\text{Hossain})} = \log_2 p(c|s, t')$$

- **Q$^{\text{(Tamura)}}$** (Tamura et al., 2023), based on the average log-likelihood normalized by sentence length $|\text{deSW}(c)|$, which is the number of words after detokenizing the subwords of the translation candidate c, and the similarity between the source sentence s and the candidate c using LaBSE sentence embeddings.

$$\begin{aligned} Q^{(\text{Tamura})} &= \alpha \frac{\log_2 p(c|s,t')}{|\text{deSW}(c)|} + (1-\alpha)\,\text{sim}(s,c) \\ &\triangleq \alpha L(s|t', c) + (1-\alpha)E(s,c) \end{aligned}$$

- **Q**$^{(\text{COMET}_{21})}$ (Rei et al., 2021), a reference-free quality evaluation score obtained through a pre-trained COMET-QE model, which achieves remarkable performance in the WMT 2021 Metrics Shared Task (Freitag et al., 2021)).
- **Q**$^{(\text{Proposed})}$, the reranking function proposed in this paper, introducing the COMET21-QE score $CM(s,c)$ into $Q^{(\text{Tamura})}$, with all the terms being linearly combined.

$$Q^{(\text{proposed})} = \alpha L(s|t',c) + \beta E(s,c) + \gamma CM(s,c)$$

- **Oracle**, the reference translation is used to identify the candidate with the highest value for the evaluation metric, serving as the absolute maximum evaluation score.

The final output probability is calculated as follows:

$$p(c|s,t') = \lambda p_{\text{kNN}}(c|s,t') + (1-\lambda) p_{\text{NFR}}(c|s,t')$$

where the value of λ depends on the specific kNN-MT method being used and those procedures are presented in section A. Here, $p_{\text{NFR}}(c|s,t')$ represents the output probability of c when s concatenated with t' is input to the trained NFR model, while $p_{\text{kNN}}(c|s,t')$ represents the output probability via kNN-MT methods. They are calculated as below:

$$p_*(c|s,t') = \prod_{c^{(l)} \in c} p_*(c^{(l)}|s,t',c^{(<l)})$$

where $p_*(c^{(l)}|s,t',c^{(<l)})$ represents the output probability at the l-th step of decoding, $c^{(<l)}$ represents the token sequence already output at the l-th step, and $c^{(l)}$ represents the token output by the decoder at the l-th step.

6 Experiment

6.1 Datasets

In this paper, we use English-Japanese (en-ja) dataset of Asian Scientific Paper Excerpt Corpus (Nakazawa et al., 2016) (ASPEC), which is extracted from scientific papers, and English-French (en-fr) dataset of EU bookshop Corpus (Skadiņš et al., 2014) (EUbookshop), which is compiled from publications sourced from diverse European institutions available on the EU bookshop website. As for the EUbookshop dataset, it is downloaded from the OPUS website[3], and we randomly sampled 1,000,000 sentence pairs for training, 2,000 sentences for validation, and 2,000 sentences for testing. In our experiments, we solely employ the training data as the TM_{para}. Additionally, we utilize the target language sentences of all the data, excluding the validation and test sets, as the TM_{mono}. Table 2 shows the detailed size of these datasets.

	ASPEC English→Japanese	EUbookshop English→French
training	100,000	1,000,000
validation	1,790	2,000
test	1,812	2,000
TM_{para}	100,000 (En-Jp)	1,000,000 (En-Fr)
TM_{mono}	2,000,000 (Jp)	8,421,120 (Fr)

Table 2: Number of sentences in training, validation and test datasets (According to the experiments conducted by Morishita et al. (2022) and the research findings of Neubig (2014), we perform experiments using a subset of 2 million sentences which exhibits higher alignment scores from original ASPEC training dataset.)

During the preprocessing stage of the data, we utilize MeCab[4] and Moses[5] to tokenize Japanese sentences and English&French sentences, respectively. Furthermore, the preprocessing module of fairseq[6] (Ott et al., 2019) in version 0.10.1 is used to split the tokens into sub-words via byte pair encoding (BPE) (Sennrich et al., 2016).

6.2 Experiment Setting

For the reranking process, we use the NFR model trained with 2-best similar translations[7] for generating 32 candidates, a number determined by validation data, as the evaluation metrics for the reranking process tend to converge with this candidate count.

We employ the Vanilla Transformer (Vaswani et al., 2017) from fairseq 0.10.1, which consists of 6 layers for both the encoder and decoder with 512 hidden dimensions, 2048 dimensions in the feedforward layers and 8 multi-heads. In addition, we utilize a warm-up of 6,000 steps for ASPEC, and 8,000 steps for EUbookshop, while train 30 epochs with a batch size of 96 sentences.

[3] https://opus.nlpl.eu/

[4] https://github.com/neologd/mecab-ipadic-neologd

[5] https://www.statmt.org/moses/

[6] https://github.com/facebookresearch/fairseq

[7] According to the description in Section B, since the models trained with the 2-best similar translations demonstrate higher performance in most cases, we choose it as the foundation for subsequent editing and reranking steps.

(a) Result of ASPEC

Retrieval	Edit	sacreBLEU$_{base}$			COMET$_{base}$			Average time for inference per sentence (sec.)
		w/o Reranking	w/Reranking		w/o Reranking	w/Reranking		
			$Q^{(\text{Proposed})}$	Oracle		$Q^{(\text{Proposed})}$	Oracle	
	w/NMT	24.51	-	-	0.9206	-	-	0.0145
Transformer	w/vanilla-kNN	24.83	-	-	0.9228†	-	-	0.0175
w/o Retrieval	w/robust-kNN	25.42†	-	-	0.9236†	-	-	0.0192
	w/NFR	24.54	26.59	31.37	0.9218	0.9395	0.9444	1.0190
Transformer	w/vanilla-kNN	24.66	26.95†	31.60†	0.9231†	0.9402	0.9451	1.0880
w/ed	w/robust-kNN	25.23†	27.41†	31.86†	0.9250†	0.9407†	0.9455†	1.1443
	w/NFR	25.01	27.41	33.42	0.9240	0.9465	0.9527	0.9470
Transformer	w/vanilla-kNN	25.44	27.59	33.58	0.9251†	0.9465	0.9530	1.0019
w/LaBSE	w/robust-kNN	25.71†	28.24†	33.98†	0.9267†	0.9480†	0.9537†	1.0519

(b) Result of EUbookshop

Retrieval	Edit	sacreBLEU$_{base}$			COMET$_{base}$			Average time for inference per sentence (sec.)
		w/o Reranking	w/Reranking		w/o Reranking	w/Reranking		
			$Q^{(\text{Proposed})}$	Oracle		$Q^{(\text{Proposed})}$	Oracle	
	w/NMT	26.57	-	-	0.6445	-	-	0.0210
Transformer	w/vanilla-kNN	26.86†	-	-	0.6478†	-	-	0.0278
w/o Retrieval	w/robust-kNN	27.61†	-	-	0.6535†	-	-	0.0311
	w/NFR	26.70	26.84	26.98	0.6424	0.6451	0.6473	1.7973
Transformer	w/vanilla-kNN	26.95	27.06†	27.23†	0.6476†	0.6509†	0.6528†	2.0148
w/ed	w/robust-kNN	27.42†	27.70†	27.86†	0.6533†	0.6568†	0.6588†	2.1205
	w/NFR	28.29	29.33	33.49	0.6519	0.6837	0.7092	1.2153
Transformer	w/vanilla-kNN	28.68†	30.37†	33.93†	0.6520	0.6849†	0.7129†	1.4328
w/LaBSE	w/robust-kNN	29.16†	30.44†	34.04†	0.6573†	0.6875†	0.7129†	1.5385

Table 3: Experiment results of the translation models with reranking methods (sacreBLEU$_{base}$ and COMET$_{base}$ represent the models that achieve the highest scareBLEU score or COMET22 score on the validation data. "w/o Reranking" represents that no reranking has been performed. "Oracle" represents that the reference translation is used to identify the candidate with the highest value for the evaluation metric. † for significant (p<0.05) difference with "w/NMT" or "w/NFR" in each retrieval method in each row. "Average time for inference" represents the average time required for the final output of each method, where, in the case of "w/o Retrieval", it does not include the processes of candidate generation and reranking, while in "w/ ed or LaBSE", these two processes are included.")

For the implementation of kNN-MT methods, we use the kNN-BOX[8] provided by Zhu et al. (2023). All the experiments are conducted on two NVIDIA RTX A6000 GPUs.

6.3 Result

As in Figure 4, "NFR" represents the proportion of noisy similar translations when generating output translations by the NFR model, and we utilize 32 similar translations used for candidate generation to perform an analysis of Noisy Similar Translation (NST). For ASPEC, without considering the kNN-MT methods, it can be observed that the noise percentages of LaBSE is lower than ed. However, for the EUbookshop, due to the inadequacy of 32 similar translations by ed, some source sentences lack concatenated similar translations, resulting in that their output translations are the same as the benchmark translation, causing the low rate MRoN.

After applying the kNN-MT methods for noise reduction, there is a decreasing trend in the MRoN for two retrieval methods. In most cases, regardless of evaluation method, with the edit of Robust kNN-MT, the similar translations exhibit the lowest noise level for both ed and LaBSE.

The result of all the experiments with different retrieval and edit methods is shown in Table 3[9]. We employ sacreBLEU and COMET22 for evaluation. For sacreBLEU, we utilize mteval[10] to perform significance test (p<0.05) by bootstrap method with resampling 1,500 samples 1,000 times, while for COMET22, the comet-compare[11] module is used for the t-test method.

As we employ the 32 output translations for NST (Noisy Similar Translation) analysis, which simultaneously serves as the 32 translation candidates for reranking, we combine the results of MRoN and translation accuracy for discussion. The Robust kNN-MT method, which exhibits the least noise in most cases, also achieves the highest translation accuracy. Furthermore, for EUbookshop, when evaluated using COMET22, the MRoN of editing with the Vanilla kNN-MT is lower than editing with the Robust kNN-MT for LaBSE, while the Robust kNN-MT outperforms the Vanilla kNN-MT

[8] https://github.com/NJUNLP/knn-box

[9] Although we omit the detailed results for $Q^{(\text{Tamura})}$ and $Q^{(\text{COMET}_{21})}$ for brevity, $Q^{(\text{Proposed})}$ achieves the highest translation accuracy, where significant improvements over $Q^{(\text{Tamura})}$ and $Q^{(\text{COMET}_{21})}$ are observed.

[10] https://github.com/odashi/mteval

[11] https://unbabel.github.io/COMET/html/running.html

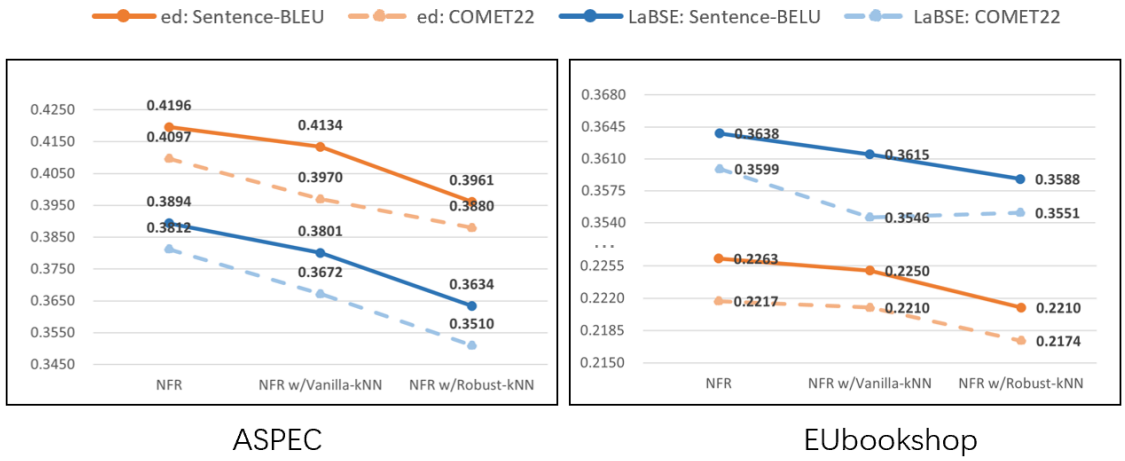

Figure 4: Results of Mean Rate of Noise (MRoN).

in both sacreBLEU and COMET22.

6.4 Speed of Translation System

As in Table 3, the speed of the Retrieve-Edit-Rerank Framework is notably slower than normal NMT model or kNN-MT methods. This disparity arises from the additional processes of retrieval and reranking introduced in the former. The methods using the LaBSE model is slightly faster compared to using edit-distance for retrieval. This is because the LaBSE model utilizes Faiss for retrieval on GPU, whereas ed relies on CPU. Compared to NMT/NFR models, after applying the kNN-MT methods, the overall time consumption increases due to the additional time required for loading datastore and k nearest neighbors searching. Furthermore, the Robust-kNN method, building upon the Vanilla-kNN approach, incorporates two additional networks, resulting in even longer inference times.

6.5 Domain Analysis of Noisy Similar Translation

In this section, we aim to investigate whether the translation accuracy damaged by NST is correlated with differences in the domain of the source sentence by human evaluation. We conduct separate random samplings of 100 NST and non-NST (nNST) samples[12] from both the ASPEC and EUbookshop datasets. The result is shown in Table 4. After employing a chi-squared test to discern whether there is a relationship between NST and the domain, the p-values for the ASPEC and EUbookshop dataset are determined to be 0.7773 and 0.7760. Therefore, it cannot be considered that the domain is a significant factor causing similar translations to be classified as NST[13].

(a) ASPEC

	In domain	Out of domain	Total
NST	54	46	100
nNST	52	48	100
Total	100	100	200

(b) EUbookshop

	In domain	Out of domain	Total
NST	55	45	100
nNST	57	43	100
Total	100	100	200

Table 4: Domain Analysis of NST and non-NST (nNST)

7 Conclusion

This paper introduces the kNN-MT method in the Edit stage of the Retrieve-Edit-Rerank framework, effectively addressing the issue caused by noisy similar translations, and enhancing the accuracy in reranking phase. Additionally, a novel reranking function is proposed, which surpasses previous research with higher precision.

[12]We require the sentence-BLEU of each NST sample to be less than that of the benchmark translation minus 10, while that of each nNST sample to be more than that of the benchmark translation plus 10.

[13]We observe that even in cases where nNST not belonging to the same domain as the source sentence can lead to improvement of translation accuracy. This may be attributed to a higher degree of cross-lingual alignment between source sentence and nNST, albeit lacking domain vocabulary in these alignments. Similarly, for NST, we notice that translation accuracy can decrease even when the source sentence is within the same domain. We suggest that this could be due to a situation where although domain-specific terms align, the alignment of common vocabulary is relatively weaker, resulting in an overall reduction in translation precision.

References

Bram Bulte and Arda Tezcan. 2019. Neural fuzzy repair: Integrating fuzzy matches into neural machine translation. In Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics, pages 1800–1809, Florence, Italy. Association for Computational Linguistics.

Deng Cai, Yan Wang, Huayang Li, Wai Lam, and Lemao Liu. 2021. Neural machine translation with monolingual translation memory. In Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers), pages 7307–7318, Online. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. BERT: pre-training of deep bidirectional transformers for language understanding. CoRR, abs/1810.04805.

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

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

Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In Proceedings of the Sixth Conference on Machine Translation, pages 733–774, Online. Association for Computational Linguistics.

Nabil Hossain, Marjan Ghazvininejad, and Luke Zettlemoyer. 2020. Simple and effective retrieve-edit-rerank text generation. In Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, pages 2532–2538, Online. 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, pages 5468–5477, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. IEEE Transactions on Big Data, 7(3):535–547.

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

Vladimir Levenshtein. 1966. inary codes capable of correcting deletions, insertions, and reversals. In Soviet Physics Doklady, pages 10(8):707–710.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach.

Zhuoyuan Mao and Tetsuji Nakagawa. 2023. LEALLA: Learning lightweight language-agnostic sentence embeddings with knowledge distillation. In Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics, pages 1886–1894, Dubrovnik, Croatia. Association for Computational Linguistics.

Makoto Morishita, Katsuki Chousa, Jun Suzuki, and Masaaki Nagata. 2022. JParaCrawl v3.0: A large-scale English-Japanese parallel corpus. In Proceedings of the Thirteenth Language Resources and Evaluation Conference, pages 6704–6710, Marseille, France. European Language Resources Association.

Toshiaki Nakazawa, Manabu Yaguchi, Kiyotaka Uchimoto, Masao Utiyama, Eiichiro Sumita, Sadao Kurohashi, and Hitoshi Isahara. 2016. ASPEC: Asian scientific paper excerpt corpus. In Proceedings of the Tenth International Conference on Language Resources and Evaluation (LREC'16), pages 2204–2208, Portorož, Slovenia. European Language Resources Association (ELRA).

Graham Neubig. 2014. Forest-to-string SMT for Asian language translation: NAIST at WAT 2014. In Proceedings of the 1st Workshop on Asian Translation (WAT2014), pages 20–25, Tokyo, Japan. Workshop on Asian Translation.

Myle Ott, Sergey Edunov, Alexei Baevski, Angela Fan, Sam Gross, Nathan Ng, David Grangier, and Michael Auli. 2019. fairseq: A fast, extensible toolkit for sequence modeling. In Proceedings of NAACL-HLT 2019: Demonstrations.

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 G. C. de Souza, Duarte Alves, Chrysoula Zerva, Ana C Farinha, Taisiya Glushkova,

Alon Lavie, Luisa 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), pages 578–585, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Ricardo Rei, Ana C Farinha, Chrysoula Zerva, Daan van Stigt, Craig Stewart, Pedro Ramos, Taisiya Glushkova, André F. T. Martins, and Alon Lavie. 2021. Are references really needed? unbabel-IST 2021 submission for the metrics shared task. In Proceedings of the Sixth Conference on Machine Translation, pages 1030–1040, Online. Association for Computational Linguistics.

Ricardo Rei, Craig Stewart, Ana C Farinha, and Alon Lavie. 2020. Unbabel's participation in the WMT20 metrics shared task. In Proceedings of the Fifth Conference on Machine Translation, pages 911–920, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. 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 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. In Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 4512–4525, Online. Association for Computational Linguistics.

Nils Reimers and Iryna Gurevych. 2021. The curse of dense low-dimensional information retrieval for large index sizes. 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 605–611, Online. Association for Computational Linguistics.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of BERT: smaller, faster, cheaper and lighter. CoRR, abs/1910.01108.

Florian Schroff, Dmitry Kalenichenko, and James Philbin. 2015. FaceNet: A unified embedding for face recognition and clustering. In 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR). IEEE.

Rico Sennrich, Barry Haddow, and Alexandra Birch. 2016. Neural machine translation of rare words with subword units. In Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), pages 1715–1725, Berlin, Germany. Association for Computational Linguistics.

Raivis Skadiņš, Jörg Tiedemann, Roberts Rozis, and Daiga Deksne. 2014. Billions of parallel words for free: Building and using the EU bookshop corpus. In Proceedings of the Ninth International Conference on Language Resources and Evaluation (LREC'14), pages 1850–1855, Reykjavik, Iceland. European Language Resources Association (ELRA).

Takuya Tamura, Xiaotian Wang, Takehito Utsuro, and Masaaki Nagata. 2023. Target language monolingual translation memory based nmt by cross-lingual retrieval of similar translations and reranking. In Proceedings of Machine Translation Summit XIX: Research Track, pages 313–323.

Tom Vanallemeersch and Vincent Vandeghinste. 2015. Assessing linguistically aware fuzzy matching in translation memories. In Proceedings of the 18th Annual Conference of the European Association for Machine Translation, pages 153–160, Antalya, Turkey.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In Advances in Neural Information Processing Systems, volume 30. Curran Associates, Inc.

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, Qianfeng Zhao, Yunzhe Lv, Shujian Huang, Siheng Zhao, Sizhe Liu, and Jiajun Chen. 2023. knn-box: A unified framework for nearest neighbor generation.

A Process of kNN-MT methods

A.1 Vanilla kNN-MT

Datastore Creation In this paper, all the employed kNN-MT methods(Khandelwal et al., 2021; Zheng et al., 2021; Jiang et al., 2022) follow the same process for constructing the datastore, composed of key-value (k, v) pairs generated from the training data of the NMT model.

Specifically, for each training source sentence $s \in \mathcal{S}$, it is input into the trained NMT model and output t in the target language, as the decoder predicts the l-th token $t^{(l)}$ based on the previously generated tokens $(s, t^{(<l)})$. The value is recorded as ground truth l-th token $t^{(l)}$ into the datastore, while its corresponding predicted hidden representation $f(t^{(l)}|s, t^{(<l)})$ is stored as a key. For a parallel

training dataset $(s,t) \in (\mathcal{S},\mathcal{T})$, the datastore is created as below:

$$(\mathcal{K},\mathcal{V}) = \bigcup_{(s,t)\in(\mathcal{S},\mathcal{T})} \{(f(t^{(l)}|s,t^{(<l)}),t^{(l)}),\forall t^{(l)} \in t \,\Big|\, (s,t)\in(\mathcal{S},\mathcal{T})\}$$

Prediction During the prediction of Vanilla kNN-MT, given a source sentence x, the NMT model can output a target sentence y with l-th token $y^{(l)}$ in every generation step. Moreover, the generated hidden representation $f(y^{(l)}|x,y^{(<l)})$ in each step is used to query the datastore for the k nearest neighbors $\mathcal{N}^{(l)}$ according to squared-L^2 distance d.

$$\mathcal{N}^{(l)} = \{(h_i,v_i), i = 1,2,\ldots,k\}$$

where v_i represents the i-th token retrieved from the datastore and h_i denotes its corresponding key vector.

Finally, the distribution by Vanilla kNN-MT over the vocabulary is calculated as follows:

$$p_{\mathrm{kNN}}(y^{(l)}|x,y^{(<l)}) \propto \sum_{(h_i,v_i)\in\mathcal{N}^{(l)}} \mathbb{1}_{y^{(l)}=v_i} \exp\left(\frac{-d(h_i,f(y^{(l)}|x,y^{(<l)}))}{T}\right) \quad (1)$$

where T indicates the temperature, and d represents squared-L^2 distance. The final output probability consisting of $p_{\mathrm{NMT}}(y^{(l)}|x,y^{(<l)})$ and $p_{\mathrm{kNN}}(y^{(l)}|x,y^{(<l)})$ is computed as below:

$$p(y^{(l)}|x,y^{(<l)}) = \lambda p_{\mathrm{kNN}} + (1-\lambda)p_{\mathrm{NMT}}$$

where p_{NMT} indicates the output probability by the NMT model and λ is the interpolation.

A.2 Robust kNN-MT

As for Robust kNN-MT(Jiang et al., 2022), which exhibits superior robustness in handling scenarios with considerable noise presenting in the datastore, and achieves the state-of-art accuracy among all kNN-MT methods, comprises two newly trained networks: the **D**istribution **C**alibration (DC) network and the **W**eight **P**rediction (WP) network. The DC network leverages the output distribution of the NMT model as ground truth to calibrate the kNN-MT distribution, and the WP network works on estimating the weight $\lambda^{(l)}$ for combining these two distributions to form the final output. The distribution by Robust kNN-MT is computed as follows:

$$p_{\mathrm{kNN}}(y^{(l)}|x,y^{(<l)}) \propto \sum_{(h_i,v_i)\in\mathcal{N}^{(l)}} \mathbb{1}_{y^{(l)}=v_i} \exp\left(\frac{-d_i}{T^{(l)}} + c_i^{(l)}\right)$$

where d_i indicates the squared-L^2 distance between h_i and $f(y^{(l)}|x,y^{(<l)}))$, $T^{(l)}$ and $c_i^{(l)}$ are the output parameters from the DC network for each step of token generation. The final output distribution by Robust kNN-MT is computed as follows:

$$p(y^{(l)}|x,y^{(<l)}) = \lambda^{(l)} p_{\mathrm{kNN}} + (1-\lambda^{(l)})p_{\mathrm{NMT}}$$

where $\lambda^{(l)}$ is also calculated for each step of generation by the WP network.

All the hyperparameters of kNN-MT methods are fine-tuned on the validation data, and we include all of them in Table 7.

B Detailed Experiment Settings and Results

We employ three retrieval methods to search for n-best similar translations for each source sentence, and subsequently train the NFR model, while "w/o Retrieval" represents the NMT model trained without similar translations. Specifically, for instance, when training with the 2-best similar translations, the input format is defined as below:

- **2-best format:**
  $\mathrm{input}\,0 : s\langle\mathrm{sep}\rangle$
  $\mathrm{input}\,1 : s\langle\mathrm{sep}\rangle t'_1$
  $\mathrm{input}\,2 : s\langle\mathrm{sep}\rangle t'_2$

where t'_i represents the similar translation ranked at the top-i in terms of similarity, and $\mathrm{input}\,0$ represents the scenario that the source sentence is not concatenated with any similar translation, which makes the implementation of generating a benchmark translation in Section 4.2 for the judgment of noisy similar translations become reasonable.

For the trained NFR model, we use the strategy of concatenating the source sentence in the test data with its top-1 similar translation as the input for the inference step. As depicted in Table 5, in most cases, the approach of training with the 2-best similar translations achieves the highest accuracy. Therefore, we employ the NFR model trained with the 2-best similar translations retrieved via these three methods for the subsequent Edit and Candidate Generation steps.

	n-best Similar Translations		ASPEC En→Jp		EUbookshop En→Fr	
	Training	Inference	scareBLEU_{base}	COMET22_{base}	scareBLEU_{base}	COMET22_{base}
w/o Retrieval	0-best	-	24.51	0.9206	26.57	0.6445
w/ed	1-best	top 1	24.60	0.9234	27.21†	0.6482†
	2-best		24.54	0.9218	26.70	0.6424
	3-best		24.32	0.9222	26.42	0.6401
	4-best		24.50	0.9210	26.28	0.6395
w/LaBSE	1-best	top 1	24.43	0.9237	27.51†	0.6489†
	2-best		25.01†	0.9240†	28.29†	0.6589†
	3-best		24.78	0.9222	27.52†	0.6497†
	4-best		24.74	0.9199	27.38†	0.6477†

Table 5: Experiment results of the training and inference (For different retrieval methods, the translation models are trained by concatenating from zero to up to four similar translations for each source sentence, while 0-best represents the situation for "w/o Retrieval". However, during the inference stage, only the similar translation with the highest similarity is concatenated to the test sentence as the input. († for significant ($p < 0.05$) difference with "w/o Retrieval".))

Retrieval	Edit	ASPEC		EUbookshop	
		$\text{Sentence-BLEU}_{base}$	COMET22_{base}	$\text{Sentence-BLEU}_{base}$	COMET22_{base}
ed	MRoN of NFR	0.4196	0.4097	0.2263	0.2217
	MRoN of NFR w/Vanilla-kNN	0.4134	0.3970	0.2250	0.2210
	MRoN of NFR w/Robust-kNN	<u>0.3961</u>	<u>0.3880</u>	**<u>0.2210</u>**	**<u>0.2174</u>**
LaBSE	MRoN of NFR	0.3894	0.3812	0.3638	0.3599
	MRoN of NFR w/Vanilla-kNN	0.3801	0.3672	0.3615	<u>0.3546</u>
	MRoN of NFR w/Robust-kNN	**<u>0.3634</u>**	**<u>0.3510</u>**	<u>0.3588</u>	0.3551

Table 6: Results of Mean Rate of Noise (MRoN). ("MRoN of NFR" represents the average of Rate of Noise (RoN) among the 32 similar translations retrieved for each test sentence. "MRoN of NFR w/kNN-MT method" represents the average RoN of the 32 output translations obtained from 32 similar translations, which were edited using kNN-MT methods and still judged as noisy similar translations. Sentence-BLEU and COMET22 are metrics utilized as indicators to assess whether a similar translation is considered noise. In the table, the smaller the values are, the smaller number of similar translations are identified as noise. The portion marked with an underline denotes the minimum value for each retrieval method, while the bold text for the minimum value among all the editing methods used.)

Retrieval	Edit	ASPEC					EUbookshop				
		λ	k	T	α_0	β	λ	k	T	α_0	β
Transformer w/o Retrieval	w/NMT			-					-		
	w/vanilla-kNN	0.1	8	10	-	-	0.1	16	10	-	-
	w/robust-kNN	-	16	-	1.0	100	-	16	-	1.0	1,000
Transformer w/ed	w/NFR			-					-		
	w/vanilla-kNN	0.1	8	10	-	-	0.1	16	10	-	-
	w/robust-kNN	-	16	-	1.0	100	-	16	-	1.0	1,000
Transformer w/LaBSE	w/NFR			-					-		
	w/vanilla-kNN	0.1	8	10	-	-	0.1	8	5	-	-
	w/robust-kNN	-	16	-	1.0	100	-	16	-	1.0	1,000

Table 7: The respective hyperparameters of the kNN-MT methods used in the experiments