Improving Neural Machine Translation Using Retrieval, Data Augmentation, and Shared Embeddings: A Project Report

Yifei Zhang[1] * **Jiahao Jiang**[1] * **Tianwei Li**[2] *
{2213218, 2211103, 2212915}@mail.nankai.edu.cn
[1]College of Computer Science, Nankai University
[2]College of Cyber Science, Nankai University

Regular Submission To NKUCS course "Natural Language Processing"

Abstract

Neural machine translation models (NMT models) achieve state-of-the-art performances in numerous translation tasks, and have been widely studied in the field of machine translation. As an effort to improve NMT models, the academia has introduced a series of methods widely employed in the field of Natural Language Processing (NLP) and significantly improved model translation quality. Despite their success in achieving improvements of NMTs, previous researches mostly focus on employing one method at a time, instead of applying multiple existing methods all at once.

In this project, we explore data augmentation, retrieval, and distillation, the three major technical routes to improve NMT models. We discover how these methods affect model performance by conducting both empirical and theoretical analysis. After reproducing the results for 5 previous papers, we attempted to incorporate retrieval and data augmentation for a single NMT model. Furthermore, we utilize shared embeddings, a setting we discovered while reproducing distillation, in our base NMT. Experiment results demonstrated that our approaches to improve prior art are successful, with an improvement of at most +2.7 in BLEU score.

The repository of our project can be accessed via this link: https://github.com/FeliceRivarez/NKU-NLP-course-project

1 Introduction

Remarkable advances have been observed in machine translation (MT) due to the rapid development of deep learning models and techniques. Currently, Neural Machine Translation (NMT) models have achieved state-of-the-art performances in almost every aspect of machine translation. In addition to new model architectures and novel deep learning frameworks, various technical routes to enhance the performance of existing NMT models have also been proposed. These generic methods have been proven to be valuable for their great scalability, significant improvement in performances, and distinctive insights for future NMT models.

Though dozens of generic approaches to enhance NMT models exist, most prior researches focus on improving a single existing method (e.g. data augmentation), and apply such improved method to NMT models independently. Intuitively, since various different methods can all achieve better performance, NMT model translation quality should be greatly improved when multiple methods are employed simultaneously. However, how multiple methods would interact with each other and affect model performance remains rather unexplored.

This phenomenon calls for a systematic evaluation on existing technical routes, as an effort to further improve NMT models and systematize prior art. In this project, we plan to explore three major technical routes that have already been applied to NMT models, and analyze their effects. Furthermore, we seek to integrate these three technical routes into the same NMT model at the same time, aiming to provide insights on how multiple independent methods would affect model performance when applied all at once.

1.1 Motivations

During the conducting of our surveys in NLP prior to this project, we find that though rapid progress have been made in all aspects of MT, different research fields are rather isolated. Specifically, when it comes to improving NMT models with different techniques, most prior work focus on improving a single existing method, and apply such improved method to NMT models. While these contribute

[*] Yifei Zhang reproduced papers involving kNN-MT, conducted exploratory experiments for both method integration and shared embeddings, set up the GPU server and drafted the paper in part. Jiahao Jiang reproduced papers involving data augmentation, preprocessed the data, assisted in exploratory experiments and drafted the paper in part. Tianwei Li attempted in reproducing distillation, conducted some related experiments, and drafted the paper in part.

to the rapid progress of NMT models, an intuitive and yet rather overlooked approach is to combine these state-of-the-art methods into a single NMT model. Situations above give rise to the following key research questions (**RQ**s) that remain to be answered:

- **RQ1**: How to correctly integrate multiple existing methods that enhances the performance of NMT models?
- **RQ2**: How would the integrated methods have impact on, or interact with each other, since these methods are mutually independent and intrinsically different?
- **RQ3**: When multiple existing methods are applied to a single NMT model simultaneously, how would the NMT model perform, compared to those that apply only a single method?

In this project, we plan to look into two intrinsically different technical routes (i.e. data augmentation, retrieval) that have been proven to be effective when applied independently, as an effort to explore their effect and differences. Furthermore, we seek to integrate those methods into a single NMT model, and evaluate model performance through extensive experiments. In addition, we also explored distillation and made an attempt to reproduce the result. Though this was not made possible due to the limitation of our machine, we explored shared embeddings and the impact of such configuration.

1.2 Challenges

At first glance, it seemed an easy job to integrate multiple existing technical routes into a NMT model all at the same time. However, after conducting a brief survey on existing methods and their original implementation (i.e. open-sourced code), we find that this is not often the case. In fact, multiple challenges arise when we look into how to implement and evaluate the performance of different methods.

First, it is difficult to choose the proper baseline. Since different methods evolve and iterate overtime, gaining better overall performance, a number of variants exist within a single existing method. For example, the retrieval method has multiple variants, from kNN-based retrieval to chunk-based retrieval, variants of a particular method are far from similar. Hence, as we need to choose a specific variant for different methods, the choice of baseline is not a straightforward task.

Second, the reproducing of different methods with respect of their reported performance is challenging, as well as the integration of multiple methods. Since most research focus on application of a single method, their implementation did not consider future integration with other methods, resulting in the low scalability in their source code. Worse still, different implementation of various methods are based on different NMT models, yet we have to re-implement all methods on a single NMT model. It is not possible to guarantee that performance would persist when the original implementation is transferred to a different model, or even an entirely different framework (e.g. from models adopted from Huggingface to fairseq framework).

In order to address the issues stated above, in this project, we seek not only to evaluate how NMT models perform when different method are applied simultaneously, but also how different variants of a single method perform. In addition, we plan to migrate available methods into a single chosen framework (e.g. Huggingface transformer model), as an effort to conduct a fair experimental analysis.

2 Contributions

We have explored different approaches to mitigate the challenges above, and the contribution of this project are as follows:

- **Successfully reproducing at least 4 papers.** We successfully reproduced 4 recent research papers, with 1 in data augmentation and 3 in kNN-MT. Through our experiment results and comparison with the base NMT model (vanilla transformer), we examined the correctness of our implementations.
- **Improving prior art.** We successfully integrated different variations of kNN retrieval with data augmentation, and conducted various experiments. Results indicate that our integration successfully improved prior art. Furthermore, we utilize shared embeddings, and looked into the impacts of this configuration.

3 Background and related work

In this experiment, we successfully replicate the four papers related to machine translation and integrate and optimize their methods. Here, the project addresses and brief summaries of four papers are presented, with further elaboration on the core content in the subsequent subsections.

KNN-Box: K-Nearest Neighbor-Augmented Neural Machine Translation

Paper/Tool repository link: https://github.com/NJUNLP/knn-box

Paper/Tool overview: This paper proposes a toolbox that enables training and evaluating multiple previous kNN-MT papers. We conducted our experiments using the vanilla kNN-MT(Khandelwal et al.), robust kNN-MT(Jiang et al., 2022), and adaptive kNN-MT(Zheng et al., 2021), which are proposed by 3 distinct previous papers.

SSMBA: Self-Supervised Memory-Based Attention for Neural Machine Translation

Paper repository link: https://github.com/nng555/ssmba

Paper overview: This paper introduces a self-supervised memory-enhanced attention mechanism (SSMBA) to improve the precision of the attention mechanism, thereby optimizing the performance of Neural Machine Translation.

Selective Distillation: A Selective Knowledge Distillation Approach for Neural Machine Translation

Paper repository link: https://github.com/LeslieOverfitting/selective_distillation

Paper overview: This paper presents a selective distillation method, which distills knowledge that significantly contributes to translation quality, thereby enhancing the performance of the student model.

Our Project: Multi-Strategy Model Replication Repository

Project link: https://github.com/FeliceRivarez/NKU-NLP-course-project

Project overview: This is a project we conducted in the NLP course at Nankai University, involving multiple tasks in the field of Natural Language Processing (NLP). The project showcases our methods and implementations in various NLP tasks.

3.1 Data augmentation

Models that perform well on a training domain often fail to generalize to out-of-domain (OOD) examples. Data augmentation is a common method used to prevent overfitting and improve OOD generalization. However, in natural language, it is difficult to generate new examples that stay on the underlying data manifold. SSMBA was be introduced, a data augmentation method for generating synthetic training examples by using a pair of corruption and reconstruction functions to move randomly on a data manifold. (Ng et al., 2020)

When the underlying data manifold exhibits easy-to-characterize properties, as in natural images, simple transformations such as translation and rotation can quickly generate local training examples. However, in domains such as natural language, it is much more difficult to find a set of invariances that preserves meaning or semantics.

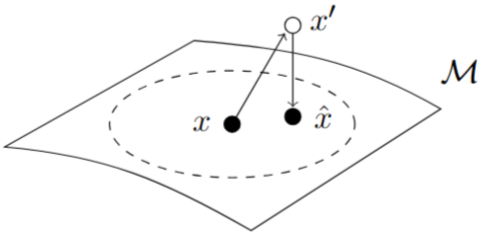


Figure 1: SSMBA moves along the data manifold M by using a corruption function to perturb an example x off the data manifold, then using a reconstruction function to project it back on.

Self-Supervised Manifold-Based Data Augmentation (SSMBA): A data augmentation method designed to generate synthetic examples in domains where the data manifold is difficult to heuristically represent. The approach involves randomly perturbing examples on the data manifold using a corruption function, then projecting them back onto the manifold using a reconstruction function (Figure 1). This ensures that the new examples lie within the manifold neighborhood of the original examples. SSMBA is applicable to any supervised task, requires no task-specific knowledge, and does not rely on class-specific or dataset-specific fine-tuning.

In the SSMBA (Self-Supervised Sequence-to-Sequence Masked-Based Augmentation) process, the MLM mechanism is used to generate new sentences, with the following steps (Figure 2):

1. Random Masking: For the input sentence, a number of tokens are randomly selected for masking using the MLM mechanism.

2. Model Prediction: A pre-trained BERT model is used to predict the original tokens at the masked positions. The model outputs a proba-

bility distribution representing the likelihood of each candidate word at the masked position.

3. Generate New Sentence: Based on the probability distribution output by the model, the token with the highest probability can be selected to replace the masked token, thereby generating a new sentence. Alternatively, tokens can be sampled from the probability distribution to increase the diversity of the generated sentence.

4. Label Handling: The class label of the newly generated sentence can either remain the same as the original sentence or be re-predicted using a trained classifier, yielding a new class label.

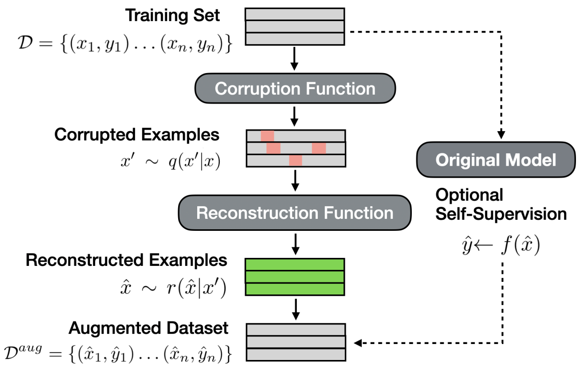

Training Set
$\mathcal{D} = \{(x_1, y_1) \ldots (x_n, y_n)\}$
Corruption Function
Corrupted Examples
$x' \sim q(x'|x)$
Original Model
Reconstruction Function
Optional
Self-Supervision
$\hat{y} \leftarrow f(\hat{x})$
Reconstructed Examples
$\hat{x} \sim r(\hat{x}|x')$
Augmented Dataset
$\mathcal{D}^{aug} = \{(\hat{x}_1, \hat{y}_1) \ldots (\hat{x}_n, \hat{y}_n)\}$

Figure 2: SSMBA generates synthetic examples by corrupting then reconstructing the original training inputs. To form the augmented dataset, corresponding outputs are preserved from the original data or generated from a supervised model f trained on the original data.

In this experiment, we manually extracted the label and context input files, as they are unnecessary in machine translation. We modified the input interface in the original code repository to use translation pairs in the form of **sentence_en \t sentence_de \n**, which helps improve data I/O speed. Following the pseudocode(Algorihtm 1) provided in the paper, we used DistilRoBERTa (Sanh et al., 2019), with 82M parameters for sentence reconstruction. In the end, 200,000 translation pairs were augmented, and the data augmentation reproduction was completed in 3 hours on a single NVIDIA RTX 4090 GPU server.

Algorithm 1 SSMBA

```
Require: perturbation function $q$
         reconstruction function $r$
Input: Dataset $D = \{(x_1, y_1) \ldots (x_n, y_n)\}$
       number of augmented examples $m$
function SSMBA($D$, $m$)
    train a model $f$ on $D$
    for $(x_i, y_i) \in D$ do
        for $j \in 1 \ldots m$ do
            1.sample perturbed
            $x'_{ij} \sim q(x'|x_i)$
            2.sample reconstructed
            $\hat{x}_{ij} \sim r(\hat{x}|x'_{ij})$
            3.generate $\hat{y}_{ij} \leftarrow f(\hat{x}_{ij})$
            or preserve the original $y_i$
        end for
    end for
    let $D^{aug} = \{(\hat{x}_{ij}, \hat{y}_{ij})\}_{i=1 \ldots n, j=1 \ldots m}$
    augment $D' \leftarrow D \cup D^{aug}$
    return $D'$
end function
```

3.2 Retrieval

Retrieval augmented machine translation utilize a pre-built datastore to reinforce the translation process on demand. Specifically, a datastore is created by leveraging ground-truth derived from accurate translation examples, and the model performs queries to the datastore when it needs to translate a sentence, with the hope that a similar or helpful translation example exist in the datastore. In this way, the machine translation model has access to the ground-truth during the translation process, which can possibly provide reference and assistance to the model, enhancing the quality of the translation. Note that since the creation of a datastore is none-parametric and usually requires no additional model training, retrieval augmentation is a none-parametric, highly scalable technical route that is widely adapted in the field of machine translation.

kNN Machine Translation (kNN-MT) is among the most promising retrieval augmentation methods. At ICLR'21, Khandelwal et al.(Khandelwal et al.) proposed k-Nearest-Neighbor retrieval augmented machine translation, which is the first to propose kNN-MT. They used high-dimensional vectors as the representation for the translation examples, and store these representations in the datastore. When the trained MT model translates a sentence on demand, the sentence is also transformed into a high-dimensional vector, and is queried in the datastore to find relevant translation example. By leveraging the retrieved examples and the model's output probability, a new probability is calculated and hence achieves retrieval augmentation.

At ACL'21, Zheng et al.(Zheng et al., 2021) introduced adaptive kNN-MT as an effort to reduce noise extracted during the kNN retrieval process. Essentially, examples retrieved by the kNN method affect translation quality in a probabilistic way, determined by various hyper-parameters. Zheng et al. designed an adaptive approach to support more accurate and reliable retrieval of results. They utilize a meta-k network to adaptively choose the number of retrieved examples for each translation task on demand.

At EMNLP'22, Jiang et al. (Jiang et al., 2022) proposed a more robust kNN-MT. They observed that vanilla kNN-MT can potentially retrieve noisy results and hinder the effectiveness of the kNN-MT method. As an effort to reduce noise extracted by the kNN retrieval process, Jiang et al. leverage the prediction confidence of NMT model to enhance the robustness of kNN-MT, achieving an increment observed in the experiment results.

3.3 Distillation

In the field of Neural Machine Translation (NMT), Knowledge Distillation (KD) has been widely applied to enhance the performance of models by transferring knowledge from a teacher model to a student model. Previous research has focused on sequence-level and word-level KD, where the student model learns to mimic the outputs of the teacher model. However, these studies rarely discuss the different impacts and connections among the samples that serve as the medium for transferring knowledge.

Wang et al. proposed a novel analytical protocol to analyze the impacts of different samples by partitioning them into two halves based on specific criteria, such as sentence length or word cross-entropy, and studying the performance gap. Extensive experiments reveal that different samples have a substantial margin in transferring knowledge, and some samples may even hurt the performance of KD. Therefore, a more sophisticated selective strategy is necessary for KD methods.

The contributions of this work include:

- Proposing a novel protocol for analyzing the property of suitable medium samples for transferring teacher's knowledge.

- Conducting extensive analyses and finding that some of the teacher's knowledge will hurt the overall effect of knowledge distillation.

- Proposing two selective strategies: batch-level selection and global-level selection. Experimental results validate the effectiveness of these methods.

Neural Machine Translation. Given a source sentence $\boldsymbol{x} = (x_1, ..., x_n)$, and its corresponding ground-truth translation sentence $\boldsymbol{y} = (y_1^*, ..., y_m^*)$, an NMT model minimizes the word negative log-likelihood loss at each position by computing cross-entropy. For the j-th word in the target sentence, the loss can be formulated as:

$$\mathcal{L}_{ce} = -\sum_{k=1}^{|V|} \mathbb{1}\{y_j^* = k\} \log p(y_j = k|\boldsymbol{y}_{<j}, \boldsymbol{x}; \theta), \tag{1}$$

where $|V|$ is the size of target vocabulary, $\mathbb{1}$ is the indicator function, and $p(\cdot|\cdot)$ denotes conditional probability with model parameterized by θ.

Word-level Knowledge Distillation.In knowledge distillation, student model S gets extra supervision signal by matching its own outputs to the probability outputs of teacher model T. Specifically, word-level knowledge distillation defines the Kullback–Leibler distance between the output distributions of student and teacher. After removing constants, the objective is formulated as:

$$\mathcal{L}_{kd} = -\sum_{k=1}^{|V|} q(y_j = k|\boldsymbol{y}_{<j}, \boldsymbol{x}; \theta_T) \times \log p(y_j = k|\boldsymbol{y}_{<j}, \boldsymbol{x}; \theta_S), \tag{2}$$

where $q(\cdot|\cdot)$ is the conditional probability of teacher model. θ_S and θ_T is the parameter set of student model and teacher model, respectively.

And then, the overall training procedure is minimizing the summation of two objectives:

$$\mathcal{L} = \mathcal{L}_{ce} + \alpha\mathcal{L}_{kd}, \tag{3}$$

where α is a weight to balance two losses.

Partition of Different Parts. Researchers proposed a new analytical protocol that divides samples into two parts based on specific criteria to study the impact of different samples on Knowledge Distillation (KD). These criteria include three different perspectives: data attributes, student models, and teacher models. Specifically:

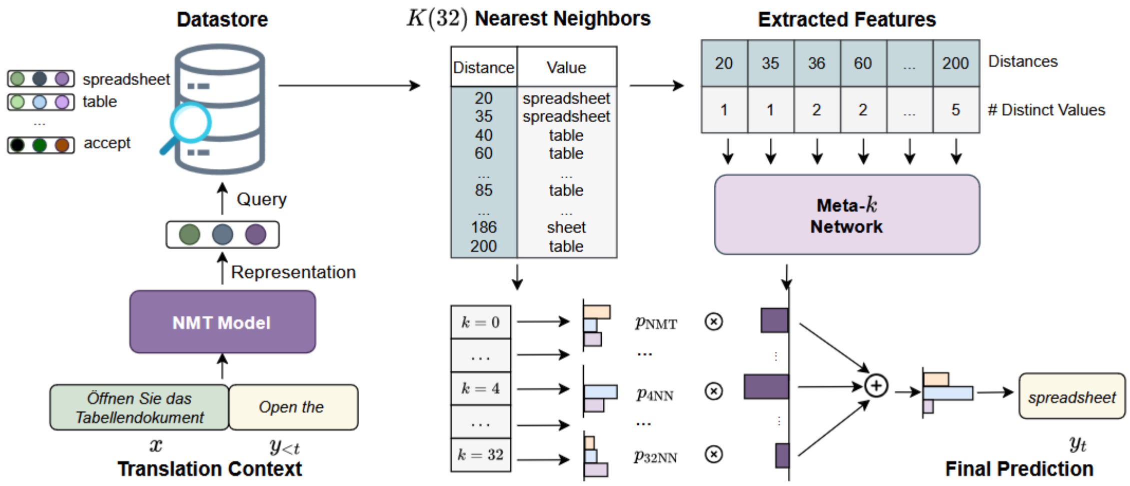


Figure 3: Adaptive kNN-MT.

- **Data Attributes**: Sentence length and word frequency are chosen as criteria because longer sentences and rare words are more difficult to translate and may contain more teacher knowledge.

- **Student Models**: Word Cross-Entropy (Word CE), Sentence Cross-Entropy (Sentence CE), and word embedding norms are used as criteria to assess which samples are considered more difficult by the student models.

- **Teacher Models**: The prediction probability of true labels (Pgolden) and the entropy of the prediction distribution are used as criteria to assess the teacher model's prediction confidence.

Criteria	BLEU		
	$\mathcal{S}_{High}$	$\mathcal{S}_{Low}$	Δ
Baseline	27.29		-
Distill-All	28.14		-
Distill-Half(Random)	28.18		-
Data Property			
Sentence Length	27.81	27.59	+0.22
Word Frequency	28.35	27.99	+0.36*
Student Model			
Embedding Norm	27.90	27.73	+0.17
Word CE	**28.42**	27.78	**+0.64***
Sentence CE	28.29	27.84	+0.45*
Teacher Model			
Teacher P_{golden}	27.97	28.00	-0.03
Entropy	27.62	27.92	-0.30

Table 1: BLEU score (%) of different criteria in WMT'14 En-De. Δ denotes the difference of BLEU score (%) between $\mathcal{S}_{High}$ and $\mathcal{S}_{Low}$. '*': significantly ($p < 0.05$) difference between the $\mathcal{S}_{High}$ and $\mathcal{S}_{Low}$.

Table 1 shows the results under different parts. The researchers also added a Transformer baseline, Distill-All (distillation using all words) and Distill-Hal (distillation with 50 percent of the words randomly selected) for performance comparison.

Selective Knowledge Distillation for NM. Researchers have proposed two simple yet effective strategies to select suitable samples for knowledge distillation (KD), namely Batch-level Selection (BLS) and Global-level Selection (GLS). These strategies aim to address the issue that some samples may negatively affect overall performance during the knowledge distillation process.

1. **B**atch-level Selection (BLS): Within the current batch, words are sorted according to word cross-entropy (Word CE), and the top r percent of words with the highest cross-entropy are selected for distillation. This method reflects the cross-entropy distribution of the current batch but may be influenced by the composition of words within the batch.

2. **G**lobal-level Selection (GLS): To better represent the global cross-entropy distribution of the model, researchers use an advanced first-in, first-out (FIFO) global queue to cache cross-entropy values from multiple steps. This approach reduces the fluctuation in cross-

entropy distribution caused by batch-level selection.

In the original paper, experimental results show that these selective strategies can significantly improve the performance of neural machine translation (NMT) models. On two large-scale machine translation tasks, WMT'14 English-German and WMT'19 Chinese-English, these methods achieved improvements of +1.28 and +0.89 BLEU points, respectively, compared to the Transformer baseline.

3.4 Shared embeddings

During our experiments on distillation, we find that a shared embeddings can be applied as a setting to neural machine translation models. Shared embeddings means that the source and the target language share the same embedding, which we believe is beneficial for kNN-MT and data augmentation, since kNN retrieval process requires a joint dictionary shared by both the source and the target language. In our exploratory experiments, we will integrate this setting with other methods to examine our intuition.

4 Experimental setup

4.1 GPU Server Configuration

The experimental environment for this study is configured as shown in the table below, with the core environment relying on a single NVIDIA RTX 4090 GPU to train the multi-strategy fusion model, as show in table2.

4.2 datasets selection

English-to-German (en-de) translation is one of the most classical tasks in the field of machine translation, involving the language pair of English and German. The characteristics of this language pair include structural differences, variations in lexical richness, and diversity across domains. In this experiment, we focus on the classical task of English-to-German translation. Research in machine translation relies heavily on large-scale parallel corpora, which consist of aligned text pairs in different languages. Commonly used machine translation datasets include WMT (Workshop on Statistical Machine Translation), IWSLT (International Workshop on Spoken Language Translation), Europarl, and OPUS, among others.

In order to ensure the proper execution of the experiments, it is essential to thoroughly investigate the data scales of existing mainstream datasets and translate them into equivalent computational resource requirements. Taking the WMT series datasets as an example, the data scale of the WMT datasets for each year is summarized in Table 3. Considering the computational environment, we account for the time costs associated with both the data augmentation process and the model training process. Ultimately, we select the first 200,000 sentence pairs from the WMT 2014 English-German (en-de) dataset as the baseline for this experiment. It is important to note that the 200,000 figure represents the initial dataset size, not the final training data scale, as the SSMBA data augmentation technique effectively doubles the dataset from 200,000 sentence pairs to 400,000 sentence pairs.

4.3 Framework

In order to make fair comparison across different methods, as well as to evaluate our efforts in improving prior art, we choose the Fairseq framework to implement and conduct all experiments. Fairseq is developed as an effort to simplify the model building for NMT models, and it supports various model structures along with optional functions.

5 Reproducing prior art

5.1 Data augmentation

The reproduction of the SSMBA data augmentation method is generally straightforward, but there are a few important considerations. One key aspect is the maximum length limit for translation pairs. Since the BERT model used in this reproduction does not support inputs longer than 512 tokens, translation pairs with a string length exceeding 512 characters were removed before data augmentation.

Additionally, modifications were made to the original code's data I/O process. In the original code, data is read from text files, which introduces some issues. For instance, translation pairs split by \t sometimes result in anomalies. Moreover, file reading is relatively slow, with high latency, causing the data augmentation program to run at full load on the server while the GPU utilization remains below 10%, which is unacceptable.

To address this, we revised the data I/O method in this reproduction. We adopted the datasets package to directly load the dataset from the Hugging Face platform into the program's memory. Within memory, we extracted the English-to-German (en-de) translation pairs and passed them directly to the SSMBA data augmentation algorithm inter-

Experimental Environment Component	Configuration
Server Rental Platform	AutoDL
Operating System	Ubuntu 20.04
Programming Language	Python 3.10
Libraries	PyTorch, TensorFlow, NumPy, etc.
Machine Configuration	NVIDIA RTX 4090, 24GB GPU Memory

Table 2: Experimental Environment Configuration

Dataset	WMT 2014	WMT 2015	WMT 2016	WMT 2017	WMT 2018	WMT 2019
Training Set	4,508,785	4,522,998	4,548,885	5,906,184	42,271,874	38,690,334
Validation Set	3,000	3,003	2,169	2,999	3,004	2,998
Test Set	3,003	2,169	2,999	3,004	2,998	2,000

Table 3: WMT 2014 - WMT 2019 German-English (de-en) Dataset Sizes

face. This change significantly improved GPU utilization, increasing the training speed by approximately three times compared to the original approach.

It is not feasible to include the augmented dataset of 200,000 pairs, ‘augmented_train_sentence_pairs_en_de_200000.txt‘, in the paper. Therefore, the dataset is provided via a Baidu Cloud link: Baidu Cloud. The extraction code is: vstj. If downloading the data seems cumbersome, you can refer to the sample data augmentation shown below, which provides a more intuitive overview.

During the experiment, we observed that the SSMBA data augmentation method, based on BERT, demonstrates the ability to recognize context and infer related sentences. This capability enables SSMBA to generate more in-domain and out-of-domain data, enhancing the robustness and diversity of the dataset while preserving the original context.

For example, when the original data is "Resumption of the session," SSMBA can automatically associate it with related matters such as parliamentary bills, creatively generating translation pairs like "It seems absolutely disgraceful that we pass judgment and do not adhere with it ourselves" and "Why are no-good areas not enforced?" (Figure 4)As proven, the augmented data indeed leads to a 4.5% performance improvement for our model.

After we produce an augmented dataset, we train the base-NMT model on the augmented dataset to observe the effects of data augmentation and to analyze its impact on translation quality. The results are shown in tab. 4, it can be seen that after applying augmentation to the original dataset, the transformer-based NMT model achieves better performance. Though the reported result for both base-NMT and SSMBA augmented NMT are lower than those in the original work, this is mainly due to the fact that we limited the training set size and significantly reduced training time. In this light, we focus on the increment data augmentation achieved, rather than the exact BLEU score.

NMTs	BLEU
base-NMT	15.3
SSMBA augmented NMT	17.62

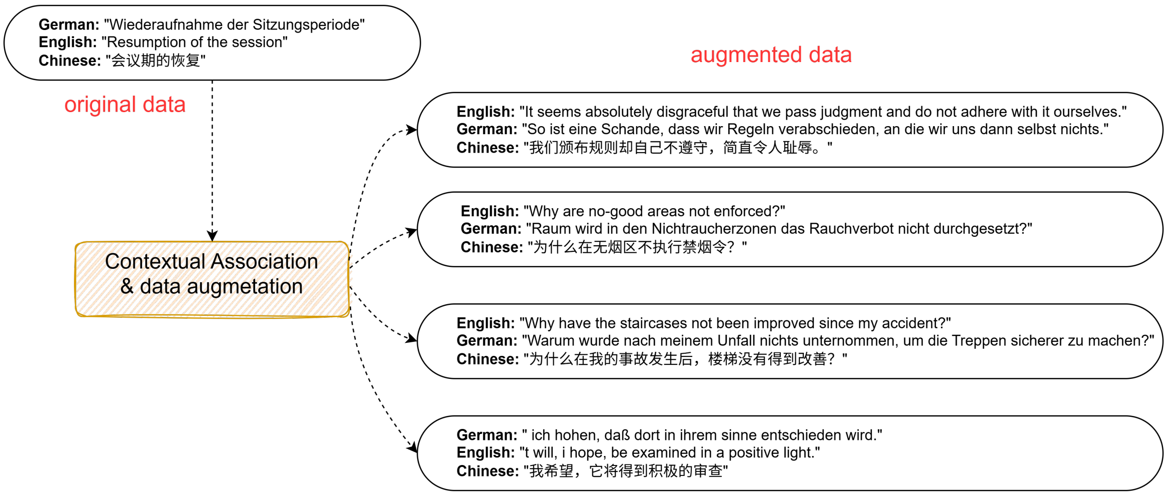
Table 4: Results for base-NMT and NMT trained on SSMBA augmented dataset.

NMTs	BLEU
base-NMT (Transformer)	15.3
vanilla kNN-MT	15.66
robust kNN-MT	15.86

Table 5: Results for base-NMT, vanilla kNN-MT, and robust kNN-MT.

5.2 kNN retrieval

For kNN-MT retrieval augmentation method, we implement and conduct experiments for base-MT (Transformer based MT model), vanilla-kNN-MT, Adaptive-kNN-MT, and Robust-kNN-MT. Note



German: "Wiederaufnahme der Sitzungsperiode"
English: "Resumption of the session"
Chinese: "会议期的恢复"

English: "It seems absolutely disgraceful that we pass judgment and do not adhere with it ourselves."
German: "So ist eine Schande, dass wir Regeln verabschieden, an die wir uns dann selbst nichts."
Chinese: "我们颁布规则却自己不遵守，简直令人耻辱。"

English: "Why are no-good areas not enforced?"
German: "Raum wird in den Nichtraucherzonen das Rauchverbot nicht durchgesetzt?"
Chinese: "为什么在无烟区不执行禁烟令？"

English: "Why have the staircases not been improved since my accident?"
German: "Warum wurde nach meinem Unfall nichts unternommen, um die Treppen sicherer zu machen?"
Chinese: "为什么在我的事故发生后，楼梯没有得到改善？"

German: " ich hohen, daß dort in ihrem sinne entschieden wird."
English: "t will, i hope, be examined in a positive light."
Chinese: "我希望，它将得到积极的审查"

Figure 4: SSMBA data augmentation will relate to the context and infer more related sentences.

that we only conducted experiments on adaptive-kNN-MT with data augmentation, hence the results will be presented in the corresponding section.

As shown in tab. 5, both vanilla kNN-MT and robust kNN-MT achieve higher BLEU score compared with the transformer-based base NMT. Though the figures are lower than those reported in the original papers, we mainly focus on the relative performance, since we greatly reduced training data amount and traning epochs. From the results, we can infer that our implementation and reproduction of kNN-MT is correct.

5.3 Distillation & shared-embedding

During our experiments trying to reproduce distillation, we found that distillation requires much more memory and is computationally expensive. We only trained the teacher model, failed to reproduce the training and testing of the student model.

However, in this failed attempt, we found that shared embeddings can also be applied to NMT models. Specifically, as kNN-MT require a joined dictionary for both source and target language, this shared embedding settings can be potentially beneficial for both the retrieval process and the training process. Hence, we switch to exploring the impact of shared embeddings instead of distillation.

The results are shown in tab. 6. It can be seen that NMT with shared embeddings significantly improved translation quality compared with base-NMT. This implies that when joined dictionary is used, shared embeddings can enable the model to learn semantic features of words more accurately.

NMTs	BLEU
base-NMT	15.3
NMT with shared embeddings	17.52

Table 6: Results for base-NMT and NMT with shared embeddings.

6 Improving prior art

6.1 Integrating data augmentation with kNN-MT

The integration of data augmentation with kNN-MT is rather straight forward, once we successfully reproduced all related methods. Specifically, kNN-MT is a none-parametric approach, and only requires a trained NMT model. Data augmentation, on the other hand, affects only the training data, and hence influence the performance of the trained model. Intuitively, the NMT model we train on an augmented dataset can be readily utilized by kNN retrieval, whereas the we only need to re-generate the datastore using the new trained model.

In this project, we seek to explore how data augmentation cooperates with kNN retrieval. Hence, based on SSMBA augmentation method, we employ vanilla kNN-MT, robust kNN-MT and Adaptive kNN-MT on top of the NMT model trained on the augmented dataset. In addition, we build datastore not only on the ground-truth, but also for augmented training data, as an effort to observe the quality of the augmented data. The validation data and test data remain unchanged.

Methods for NMT improvement	BLEU
vanilla kNN-MT	15.3
vanilla data augmentation (SSMBA)	17.62
(ours) vanilla kNN-MT + data augmentation (SSMBA)	18.36
(ours) robust kNN-MT + data augmentation	18.24
(ours) adaptive kNN-MT + data augmentation	18.03
(ours) data augmentation (SSMBA) +shared embeddings	17.66
(ours) shared embeddings+vanilla kNN-MT	18.03

Table 7: Results for exploratory experiments. Except for the first two rows, other figures are the results of our attempt to improve previous methods.

6.2 Using shared embeddings

Shared embeddings is an optional configuration for transformer-based NMT. Though this setting have not been stressed in previous related work, it is of great interest in the context of this project, since we pre-process our data using a joined dictionary for both source and target language, in order to apply kNN retrieval to the base NMT. Using shared embeddings, the model can potentially capture semantic feature of words more accurately, and improve the effectiveness of kNN retrieval method since it changes the high-dimensional representation for sentences in the datastore.

In this project, we investigate how shared embeddings affect model performance, for both kNN retrieval and data augmentation.

6.3 Experiment results

We first explore how NMT models perform when trained on an augmented training set (using SSMBA), then enhanced with a datastore and kNN retrieval. As shown in tab. 7, for vanilla kNN-MT, robust kNN-MT, and adaptive kNN-MT, they all outperform their original implementation after using data augmentation techniques. This indicates that data augmentation can be effectively integrated with kNN retrieval augmentation techniques.

The reason behind this improvement can be that when a good data augmentation method is selected, the generated data can also provide guidance for the model, when stored in the datastore. Additionally, data augmentation itself enhances model performance, after applying kNN retrieval to the model, the NMT model can still access the original training data during translation, hence enabling more accurate translation.

Later, we enable shared embeddings configuration for the transformer-based NMT model, and train the model on augmented data, or build a datastore after the training. The results for shared-embeddings-enabled data augmentation and kNN retrieval are shown in tab. 7, it can be seen that the model performs significantly better after enabling the shared embeddings option. This increment is within our expectation, as shared embeddings improve the base NMT model's performance, possibly due to the fact that in kNN-MT, a joined dictionary is applied for data tokenization.

7 Conclusion

In this project, we first successfully reproduced SSMBA data augmentation, vanilla kNN-MT, robust kNN-MT, and adaptive kNN-MT (with adaptive kNN-MT only experimented with augmented dataset). By comparing the relative results compared with the baseline NMT model, we believe our reproduction is correct.

On top of reproducing prior art, we attempt to improve existing work. We integrated kNN retrieval with data augmentation, and successfully achieved improvement compared with their original implementation. Furthermore, we explored the effect of shared embeddings, which we came across during our failed experiments in reproducing distillation. Experiments demonstrated that in the context of kNN retrieval where a joined dictionary is required, shared embeddings is beneficial for model's translation quality.

References

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.

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

Nathan Ng, Shashank Sharma, and Richard Socher. 2020. Ssmba: A data augmentation method for improving out-of-domain generalization. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*.

Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. 2019. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. In *NeurIPS 2019 Workshop on Energy Efficient Machine Learning and Cognitive Computing*.

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)*. Association for Computational Linguistics.