A Further Investigation of Generality and Robustness of Adaptive kNN-MT model

Suofeiya Man
Faculty of Arts
McGill University
suofeiya.man@mail.mcgill.ca

Siyuan Sun
Faculty of Science
McGill University
siyuan.sun@mail.mcgill.ca

Ye Yuan
Faculty of Science
McGill University
ye.yuan3@mail.mcgill.ca

Abstract

Adaptive kNN-MT, recently proposed by Zheng et al. (2021), successfully combines pre-trained neural machine translation (NMT) model with token-level kNN retrieval, as well as a light-weight Meta-k Network to improve the translation accuracy. In this paper, we first verify the reproducibility of Adaptive kNN-MT, and then further extend the experiments not conducted in the original paper to evaluate the two significant properties of Adaptive kNN-MT model concluded by Zheng et al. (2021): generality and robustness. Through the experiments, we have shown that Adaptive kNN-MT outperforms Vanilla kNN-MT in all four domains. Furthermore, different from the original paper, generality is sensitive to the choice of k. For k=4 in our case, the generality of the adaptive kNN model is not desirable. Moreover, we verified that Adaptive kNN-MT model is more robust than the Vanilla model.

1 Introduction

Machine translation is an automatic translation process that computer programs are used for translating one language to another language for natural language processing. The model reproduced in this project, namely, Adaptive kNN-MT is a combination of a pre-trained neural machine translation and a KNN classifier over a datastore of cached context representations and corresponding target tokens. The model can achieve generality and robustness at the same time. The main objective of the project is to verify the following two claims.

Claim 1: Adaptive kNN-MT outperforms Vanilla kNN-MT in all four domains (IT/Koran/Medical/Law).

Claim 2: the two properties of Adaptive kNN-MT model proposed in the original paper: generality and robustness. This is motivated by the fact that only a portion of results are provided by the paper. We highly doubt that the results not shown are questionable. Thus, we exhaustively tested all combinations of domains.

2 Related work

Nowadays, neural machine translation is widely used in machine translation, and it is a single neural network that consists of encoder and decoder networks, also trained end-to-end to map from the original sentence to its translation. Recent research has combined word and phrase retrieval with neural machine translation to gain some benefits from the word-based and phrase-based methods. For example, Zhang et al. (2018) offered a strategy for recalling and adding previously encountered translation samples into the NMT decoding process. They collected n-grams that are both in the retrieved target sentences as translation pieces, using a search engine to find the sentence pairings whose source sides are comparable to the input phrase. Based on the similarity between the input sentence and the retrieved source sentences, pseudo-probabilities for each retrieved sentence were computed and used for weighting the retrieved translation components. The proposed method KNN-NMT model has various advantages over the above method, including the fact that the external datastore only needs to be constructed once, and KNN-NMT uses the neural context representations to scale retrieval to orders-of-magnitude larger datastores.

Gu et al. (2018) further analyzed the attention-based machine translation (NMT) model by allowing it to access the complete training set of sentence pairs after it has been trained. The method proposed by this study includes two stages, the retrieval stage and the translation stage. At the retrieval stage, a black-box search engine is used to recover a small fraction of sentence pairings from

a training dataset. Then the similarity score was used to re-rank the training corpus. At the translation stage, a novel extension of the attention-based neural machine translation model was developed to merge a current source phrase with a collection of retrieved translation pairs. The approach used in the study showed a good performance of the retrieved training sentence pairs and achieved critical improvement on the baseline of attention-based neural machine translation. However, unlike their approach, our work focuses on the combination of a pre-trained neural machine translation and a KNN classifier over a datastore of cached context representations and corresponding target tokens instead of attention-based machine translation.
Khandelwal et al. (2020) proposed a k-nearest-neighbour machine translation (KNN-MT) model, which uses representations from a neural translation model for similarity search to predict tokens with the nearest neighbour classifier over a large datastore of cached samples. the kNN-MT model uses a domain-specific datastore to adopt a single model to several domains, improving performance by an average of BLEU over the zero-shot transfer and obtaining new state-of-the-art results — all without having to train on these domains. This is similar to our approach since the model is based on the KNN-MT model. However, the model proposed in this study is a combination of a pre-trained neural machine translation and a KNN classifier over a datastore of cached context representations and corresponding target tokens.

3 Computational requirements

For this project, we used the virtual machine instance and Colab from Google Cloud Platform. The GPU we used for our virtual machine is NVIDIA Tesla T4 graphics card and CPU with 4 cores. The operating system we used is Cent OS, which supports running shell scripts. Due to the limited computation resources, we were not able to do the experiments for larger values (>4) of k.

4 Method

4.1 Models

The adaptive kNN-MT was compared with the baseline model (Vanilla kNN-MT) to carry out the experiment.

Model 1: Vanilla kNN-MT
It is proposed by Aharoni and Goldberg (2020), the model successfully combines a pre-trained neural machine translation (NMT) model with token-level KNN retrieval to improve the accuracy. The limitation is that it retrieves the same number of neighbours for each token and may include noises and cause prediction errors.

Model 2: Adaptive kNN-MT
This variant of base kNN-MT proposed by Khandelwal et al. (2020) can dynamically determine the choice of k for each target token by a light-weight Meta-k Network, based on which Adaptive kNN-MET leverages in-domain datastore better. This model shows how to use cached contextual information in inference in a simple but effective way. Meta-k Network helps to effectively filter out the noises when retrieving the neighbours and prevent performance degradation on the Vanilla KNN-MT baseline. In the original paper, it outperforms the Vanilla kNN-MT in all-domains and is stated to be more robust in out-of-domain experiments.

4.2 Implementation details of kNN-MT

In each of kNN-MT model, there are two main steps: datastore-creation step and prediction step.

4.2.1 Datastore creation

Datastore consists of a set of key-value pairs. It is constructed through a single forward pass over the training set (X,Y). Specifically, given a bilingual sentence pair from the training set:

$$(x, y) \in (X, Y)$$

The pre-trained autoregressive NMT decoder translates the t-th target token y_t based on the translation context: $x, y_{<t}$. Datastore is used to query for k nearest neighbours with respect to the distance l_2 given the context representation:

$$f(x, \hat{y}_{<t})$$

4.2.2 Prediction

During inference, at each decoding step t, given previously generated tokens $\hat{y}_{<t}$ and the context representation

$$f(x, \hat{y}_{<t})$$

The kNN-MT model assigns $\hat{y}_t$ to the most probable result based on the neighbours retrieved from the datastore. Denote the retrieved neighbours

$$N_t = \{(h_i, v_i) \in \{1, 2, ..., k\}\}$$

Their distribution is computed as

$$p_{kNN}(y_t|x, \hat{y}_{<t}) \propto$$
$$\sum_{h_i,v_i} I_{y_t=v_t} exp(\frac{-d(h_i, f(x, \hat{y}_{<t})}{T})$$

Where T is the temperature and $d_{(.,.)}$ indicates l_2 distance.The final probability is calculated as the interpolation of two distributions with a hyper-parameter λ:

$$p(y_t|x, \hat{y}_{<t}) =$$
$$\lambda p_{kNN}(y_t|x, \hat{y}_{<t}) + (1 - \lambda)p_{NMT}(y_t|x, \hat{y}_{<t})$$

4.3 Dataset and evaluation metrics

The model utilizes one multi-domain dataset as the pre-trained baseline and the other 4 datasets with different domains of IT, Medical, Koran, and Law from the OPUS corpus (Tiedemann, 2012). The raw data is then preprocessed with toolkits (Moses-Smt, 2017) for tokenization and the bpe-codes (Ng et al., 2019) provided by pre-trained model. SacreBLEU score is used for model evaluation.

4.4 Hyperparameter tuning

We used the fairseq toolkit (Ott et al., 2019) and faiss (Johnson et al., 2019) to replicate the kNN-MT model. For kNN-MT, we tune the hyper-parameter λ and report the best scores for each domain. The hidden size of the two-layer FFN in Meta-k Network was set to 32. We directly use the dev set (about 2k sents) to train the Meta-k Network for about 5k steps. We used Adam (Kingma and Ba, 2015) to optimize our model, the learning rate is set to 3e-4, and batch size is set to 32 sentences.
To verify the first claim, we reran the code and compared the BLEU score of the 2 main models: Vanilla kNN-MT and Adaptive kNN-MT.
To verify the second claim, we set K=4 in all settings. To check the generality, we utilized the Meta-k Network trained on each of these 4 domains to evaluate the other three domains. As for robustness, we used one domain, denoted as domain A, and its datastore to tune hyperparameters of Vanilla kNN-MT and to train Meta-k Network of Adaptive kNN-MT, then applied the test set of other domains, denoted as domain B, with A's datastore.

5 Results

5.1 Vanilla kNN-MT vs. Adaptive kNN-MT

This section refers to Table 2.

5.2 Generality

To demonstrate the generality of our method, we directly utilized the Meta-k Network that was trained on one domain and applied to the test sets of all domains.
This section refers to Table 3.

5.3 Robustness

This section refers to Table 4.

6 Discussion and conclusion

In this project, we reproduced the Adaptive kNN-MT model and compared it with the Vanilla kNN-MT model to investigate two claims in our selected paper. Through our experiments, we reached some conclusions in the original paper. We found that Adaptive kNN-MT can indeed improve the performance of Vanilla kNN-MT model. But for k=4, the generality of the adaptive model was not desirable, which is contradicted with the claim in the original paper. Additionally, the robustness of the Adaptive kNN-MT model is better than that of the Vanilla kNN-MT model.

6.1 Vanilla kNN-MT vs. Adaptive kNN-MT

Table 2 supports the first claim, which is that Adaptive kNN-MT outperforms Vanilla kNN-MT in all four domains (IT/Koran/Medical/Law).

6.2 Generality

As shown in Table 3, compared to the corresponding results provided in the original paper, generality cannot be proved. Specifically, the "in-domain" performance (bold) remains mostly identical, whereas the out-of-domain performance drastically decreases with a large variance. This is because we only tested a subset of k values due to computation constraints: the original paper chooses k=32, while we used k=4. We conclude that generality correlates with the choice of the upper bound K. Because the hyper-parameter k is the upper bound in our experiment, it is reasonable to say that the larger value of k means more choices for our model. Consequently, we would achieve more general models. Moreover, generality implies sacrificing computation. The training time for using k=1 to k=4 is about five hours for each domains. It will cost more time if we use larger value of k.

Corpus	Words	Sentences	W/S
Law (Acquis)	18,128,173	715,372	25.3
Medical (EMEA)	14,301,472	1,104,752	12.9
IT	3,041,677	337,817	9.0
Koran (Tanzil)	9,848,539	480,421	20.5
Subtitles	114,371,754	13,873,398	8.2

Table 1: The corpus

k	Domain	IT Base NMT:38.35		Medical Base NMT:39.99		Koran Base NMT:16.26		Law Base NMT:45.48	
	Model	Vanilla	Adap	Vanilla	Adap	Vanilla	Adap	Vanilla	Adap
	1	39.48	41.90	51.39	51.00	16.21	18.27	58.53	58.48
	2	43.48	46.28	53.90	55.73	17.83	19.15	60.84	61.57
	4	44.70	47.38	54.31	55.22	19.35	18.77	61.36	62.74

Table 2: Vanilla kNN-MT vs. Adaptive kNN-MT (measured with BLEU score).

Adaptive kNN-MT	Test in IT	Test in Medical	Test in Koran	Test in Law
Training in IT	**47.38**	37.10	14.85	40.63
Training in Medical	30.24	**55.22**	11.96	37.74
Training in Koran	31.19	31.32	**18.77**	38.22
Training in Law	33.52	35.64	14.21	**62.74**

Table 3: All combinations of domains to verify generality (measured with BLEU score).

Training Domain->Test Domain	Basel NMT	Vanilla kNN-MT	Adaptive kNN-MT
IT ->Law	45.48	20.84	40.63
Law ->IT	38.35	15.58	33.52
IT ->Koran	16.26	7.14	14.85
Koran ->IT	38.35	2.26	31.19
IT ->Medical	39.99	23.41	37.10
Medical ->IT	38.35	13.59	30.24
Law ->Koran	16.26	8.18	14.21
Koran ->Law	45.48	1.27	38.22
Law ->Medical	39.99	21.57	35.64
Medical ->Law	45.48	22.41	37.74
Medical ->Koran	16.26	5.56	11.96
Koran ->Medical	39.99	1.70	31.32

Table 4: All combinations of domains to verify robustness (measured with BLEU score)

6.3 Robustness

As shown in Table 4, we compared the performance of the Vanilla model and the adaptive kNN-MT model. The adaptive kNN-MT model is obviously more robust than the Vanilla model. For each combination we chose for testing, the BLEU score of the adaptive kNN-MT model is almost two times of the BLEU score of the Vanilla Model. This part of the experiment verified the claim about robustness in the original paper. The adaptive kNN-MT model significantly enhances the robustness with a smaller variance. In the original paper, Zheng et al. (2021) mentioned that the Vanilla model is sensitive to the choice of the value of k. As shown in the Table 4, some BLEU scores are barely satisfactory, which verified the original claim.

7 Statement of contribution

Suofeiya Man is responsible for paper writeup and proofreading.
Siyuan Sun is responsible for paper writeup and formatting.
Ye Yuan is responsible for paper selection, code running and computer model reproduction.

References

Roee Aharoni and Yoav Goldberg. 2020. Unsupervised domain clusters in pretrained language models. *arXiv preprint arXiv:2004.02105*.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 32.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with gpus. *IEEE Transactions on Big Data*.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. *arXiv preprint arXiv:2010.00710*.

Diederik P Kingma and Jimmy Ba. 2015. Adam: A method for stochastic optimization. in yoshua bengio and yann lecun editors. In *3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings*.

Moses Moses-Smt. 2017. Moses-smt/mosesdecoder: Moses, the machine translation system.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook fair's wmt19 news translation task submission.

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. *arXiv preprint arXiv:1904.01038*.

Tiedemann. 2012. ... the open parallel corpus.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. Guiding neural machine translation with retrieved translation pieces. *arXiv preprint arXiv:1804.02559*.

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. 2021. Adaptive nearest neighbor machine translation. *arXiv preprint arXiv:2105.13022*.