Faster and More Robust Low-Resource Nearest Neighbor Machine Translation

Shuo Sun[1,2], Hongxu Hou[1,2(✉)], Zongheng Yang[1,2], and Yisong Wang[1,2]

[1] College of Computer Science, Inner Mongolia University National and Local Joint Engineering Research Center of Intelligent Information, Hohhot, China
cshhx@imu.edu.cn

[2] Processing Technology for Mongolian, Inner Mongolia Key Laboratory of Mongolian Information Processing Technology,Hohhot, China

Abstract. Transformer-based neural machine translation (NMT) models have achieved performance close to human-level on some languages, but still suffer from poor interpretability and scalability of the models. Many advanced studies enhance the model's translation ability by building external memory modules and utilizing retrieval operations, however, it suffers from poor robustness and low decoding efficiency while improving the model performance, especially for low-resource translation tasks. In this paper, we propose a confidence-based gating mechanism to optimize the decoding efficiency by building a sub-network to determine the confidence of the model's own translation capability and then decide whether the current translation needs to be retrieved from the memory module. By reducing the number of retrievals to improve the model's translation speed without degrading the translation quality as much as possible. In addition, we use a nonparametric dynamic Monte Carlo-based algorithm to fuse retrieval probabilities and model predictions to improve the generalization and robustness of the model. Extensive experiments on different datasets demonstrate the effectiveness of our method.

Keywords: Memory Module · Gate Mechanism · Monte Carlo · Low-Resource Translation

1 Introduction

Neural machine translation (NMT) [5,14] has achieved levels comparable to human translation on multiple large-scale datasets. However, the neural network's neuron parameters have an upper limit on the "memory" of the corpus, and it has poor interpretability of the machine translation model for the learned knowledge. Moreover, when encountering new "knowledge", the model requires large-scale parameter updates and the scalability of the model is limited, especially obvious for low-resource tasks. The recently proposed kNN-MT and its variants [7,15,19,20] combine the traditional NMT model with a token-level memory retrieval module. These methods decouple the memory ability of the

F. Liu et al. (Eds.): NLPCC 2023, LNAI 14303, pp. 404–415, 2023.
https://doi.org/10.1007/978-3-031-44696-2_32

model from the model parameters by storing the training data in the memory module, realizing it to directly access the domain-specific datastore to improve translation accuracy without fine-tuning the entire model, gentle to cope with the discrepancy across domain distributions and improve the generality of the trained models.

Previous works usually use simple linear interpolation to fuse external knowledge guidance and NMT prediction, and use a hyperparameter to control the fusion ratio to obtain the final probability distribution. However, using the same fusion ratio for all sentences may bring some problems, while it is proved through our experiments that the model translation results are quite sensitive to the selection of hyperparameter, which affects the robustness and stability of the model. Furthermore, although kNN-MT and its related models greatly improve the model performance, there is a huge drawback in the practical application, that is, the slow decoding efficiency of the model. The main reason for this phenomenon is that the memory module capacity is quite large, the similarity calculation of high-dimensional vectors is required in finding similar sentences, and the whole memory module must be searched for each retrieval probability during decoding.

This paper aims to improve the performance of low-resource machine translation model by solving the above problems. For the former, in the process of retrieval and fusion of external memory module, we abandon the traditional linear interpolation and adopt non-parametric dynamic fusion method based on Monte Carlo, which improves the robustness and generalization of the model. For the latter, we optimize the translation speed by reducing the retrieval frequency. Specifically, a sub-network is used to judge the confidence of the model's prediction results, and retrieval is performed only with the low confidence of the model's prediction results, and the decoding efficiency is improved by filtering some unnecessary retrieval operations. Extensive experiments on low-resource translation task CCMT2019 and medium-high resource task CCMT2022 Mongolian-Chinese demonstrate the effectiveness of our method.

2 Background and Related Work

2.1 Memory-Augmented NMT

Mark [2] first applies memory-augmented neural network to machine translation. He combines word correspondences from statistical machine translation in the form of "memory" to the decoder to increase the probability of occurrence of rare words, which is particularly effective on small data sets. Akiko [4] enhances the model's translation capability by constructing a sentence-level memory bank. Zhang [18] constructs a fragment-level memory bank that allows the model to obtain more information from it and collect n-gram translation fragments from the target side with higher similarity and alignment scores.

Khandelwal proposes kNN-MT [7] builds a token-level memory module on the basis of the traditional NMT, which stores the contextual representation of the training data and the key-value pairs of target words, so that the matching degree of memory library retrieval is higher. Figure 1 illustrates of how to employ the kNN algorithm to retrieve from the memory module. The key idea is to query the corresponding word of neighboring sentences similar to the current sentence in the external memory module when translating the current word to obtain reference and guidance from the module, and then use a simple linear interpolation to probabilistically fuse with the translation results of NMT to obtain the final translation results:

$$p\left(y_t \mid x, \hat{y}_{1:i-1}\right) = \lambda p_{NMT}\left(y_t \mid y_{<t}, x\right) + (1-\lambda)\, p_{Mem}\left(y_t \mid y_{<t}\right) \tag{1}$$

After that, many variant models have been proposed, such as Adaptive kNN-MT [19], which trains a meta-k network by artificially constructing features to generate the nearest neighbor hyper-parameter k. Fast kNN-MT [12] introduces efficient hierarchical retrieval to improve the slow translation speed. Moreover, many researchers apply this idea to other natural language processing fields, such as question answering tasks and dialogue systems.

2.2 Decoding Efficiency Optimization

During the development process of memory-augmented NMT, the decoding efficiency of these models remains slowly even though vector retrieval tools like Faiss [6] are available. We summarize three mainstream decoding optimization algorithms in recent years:

1. Dimensionality reduction algorithms such as PCA and SVD are used to reduce the high-dimensional vectors of the memory module. These algorithms are simple to operate, but the disadvantage is that some of the high-dimensional position information will be lost during the dimensionality reduction process, which has a certain negative impact on the translation performance [15].
2. Reducing the memory module capacity by merging key-value pairs [11] or clustering high-dimensional vectors and discarding redundant entries [15], thereby narrowing the scope of retrieval and improving decoding efficiency. Experiments show that both methods can greatly reduce the capacity of memory module, but the disadvantage is that the performance of the model decreases significantly.
3. Narrowing the retrieval frequency by saving a certain amount of retrieval history [16] or adjusting the retrieval granularity [10]. The former imitates the caching technology in computer architecture, while the latter draws on space-for-time operation in algorithm design to reduce the number of retrievals by retrieving more tokens at once, and uses heuristic rules to decide which retrieval processes need to be discarded.

3 Methodology

The overall architecture of the model is shown in Fig. 1, and this section describes the methodology of this paper specifically.

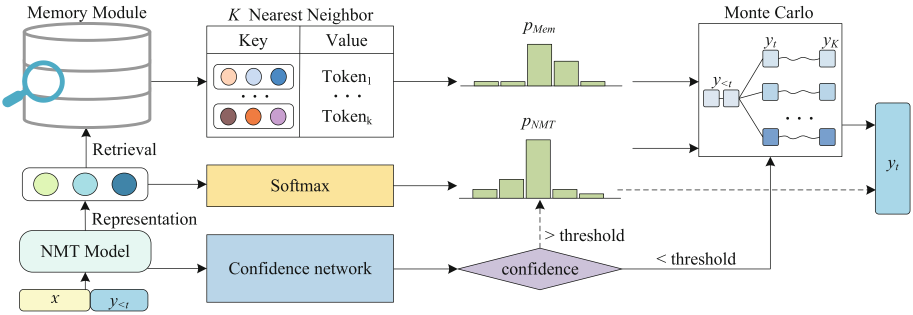

Fig. 1. The Illustration of the proposed method. The confidence network generates a confidence estimate c_t at each step of decoding, and outputs the model prediction directly if c_t is larger than the set threshold c, otherwise, constructs a retrieval probability p_{Mem} to represent the "guidance" of the memory module to the model by retrieving similar contexts from it. Then, dynamically fuses p_{Mem} and p_{NMT} based on Monte Carlo algorithm to obtain the final prediction y_t.

3.1 Monte Carlo Non-parametric Fusion

Vanilla kNN-MT [7] uses simple linear interpolation to fuse p_{NMT} and p_{Mem}, however, due to the long tail effect of the dataset, some sentences have many similar sentences and some sentences have few. It may cause insufficient information for some sentences and noise for others by applying the same fusion ratio to all sentences. To solve this problem, this paper proposes a non-parametric dynamic fusion method based on Monte Carlo algorithm, which abandons the fixed fusion of linear interpolation and alleviates the problem that the fixed fusion ratio cannot adapt to all fusion scenarios. Our method mainly applies to the inference stage, the prediction retrieved from the memory module and the prediction of NMT placed in a large sample collection. According to Y < T uses the Monte Carlo algorithm to simulate the entire sentence, and the prediction of the sentence with the highest BLEU is selected as the current word and output.

Specifically, we use the generator parameters θ of an already trained Conditional Sequence Generative Adversarial Nets [17] and apply the Monte Carlo search under the policy gradient of G^θ to sample the unknown tokens.

$$\left\{y^1_{1:T_1}, ..., y^N_{1:T_N}\right\} = MC^{G^\theta}\left(\left(y_{1:t-1}, x\right), N\right) \tag{2}$$

where T_i represents the length of the sentence sampled by the i'th Monte Carlo search. $y_{1:t-1}$ is the previously generated tokens and $y^N_{t:T_N}$ is sampled based on the policy G^θ. We calculate the BLEU of N sentences and take the current word y_t as the final prediction, which simulates sentence has the highest BLEU.

3.2 Gating Mechanism Based on Confidence Estimation

To enhance the decoding efficiency without affecting the model's translation quality, this paper proposes a decoding efficiency optimization algorithm based on the idea of reducing the retrieval frequency with a confidence-based gating mechanism. The model output is used directly without retrieval from the memory module if the confidence level is higher, and vice versa with retrieval to assist the model in generating words with higher confidence.

Inspired by DeVries [3] and Lu's [9] study, we interpret the confidence as how many prompts the NMT model needs to make a correct prediction. During training, the model can use groud-truth to generate complex translations, but each prompt comes at the cost of a certain penalty. We encourage the model to translate independently in most cases to avoid penalties, but when the model's own capabilities are insufficient to generate tokens with high confidence, the reference's help is available to ensure that the loss function is reduced. Therefore, this paper utilizes a confidence network (Fig. 3) to learn the word-level confidence, which takes the hidden variable of the decoder as the input and outputs a single scalar between 0 and 1 as the current generated word's confidence, and takes the confidence estimate as the threshold indicator for the gating mechanism, c_t closer to 1 indicates the model is confident that it can translate correctly, otherwise output c_t closer to 0 for more prompts:

$$c_t = \sigma(W'h_t + b') \tag{3}$$

where W' and b' are trainable parameters. $\sigma(\cdot)$ is the sigmoid function. To supply the model "prompts" during training, we employ c_t as an interpolation ratio to weight fusion the one-hot encoding of ground-truth y_t with the model prediction to adjust the original prediction probability, and the translation loss is calculated using the adjusted prediction probabilities:

$$p'_t = c_t \cdot p_t + (1 - c_t) \cdot y_t \tag{4}$$

$$\mathcal{L}_{NMT} = \textstyle\sum_{t=1}^{T} - y_t log(p'_t) \tag{5}$$

Furthermore, we add a penalty in the loss function to prevent the model from minimizing the loss by setting $c_t \rightarrow 0$. The final loss is the weighted sum of the translation loss and the confidence loss. Since the model is "fragile" during early training stage and cannot provide prompts in the initial training stage, the value of λ is dynamically controlled using the training step, and λ_0 and β_0 control the initial value and the declining speed of λ.:

$$\mathcal{L}_{Conf} = \textstyle\sum_{t=1}^{T} - log(c_t) \tag{6}$$

$$\mathcal{L} = \mathcal{L}_{NMT} + \lambda\mathcal{L}_{Conf} \qquad \lambda(s) = \lambda_0 * e^{\frac{-s}{\beta_0}} \tag{7}$$

Gating mechanism is a psychological concept, which refers to the mechanism of screening and filtering input information in people's memory and cognitive systems. The main purpose of the gating mechanism proposed in this paper is to filter some unnecessary retrievals of the model, so as to reduce the retrieval times and improve the decoding efficiency. The confidence network conducts synchronous training with the NMT model. During the decoding process, the confidence network generates a confidence estimate $c_t(c_t \subseteq [0, 1])$ at each step to determine whether the current retrieval operation needs to be performed, it's output directly if c_t is larger than the set threshold. We set $\lambda_0 = 30$, $\beta_0 = 45000$ and the threshold $c = 0.9$ in the settings of the confidence network.

4 Experiment

4.1 Datasets, Baselines and Configurations

This paper mainly improves on the Mongolian-Chinese translation task. The experiment's corpus comes from CCMT2019 and CCMT2022 to explore the model performance in low-resource and medium-high-resource scenarios respectively. Table 1 shows the specific size of two corpus. According to previous research and experimental verification on this translation task, we use the preprocessing operations of ULM and word segmentation+ULM for Mongolian and Chinese respectively.

We compare our method against the traditional Transformer-base [14] and some classical or leading memory-augmented NMT baselines including: MANN [2], TM-augmented [1], kNN-MT [7], Adaptive kNN-MT [19], Fast kNN-MT [12]. Due to the characteristics of Chinese, different segmentation methods may cause huge differences in BLEU scores, so we use SacreBLEU [13] to evaluate the results. We adopt Adam optimizer [8] and set 2000 warm-up steps. All the above baselines and our method are based on fairseq[1] implementation.

Table 1. The information table of experimental corpus.

Corpus		CCMT2019			CCMT2022		
		train	valid	test	train	valid	test
Mongolian	sentence	247,829	1,000	1,000	962,986	10,000	10,000
	token	7,024,958	52,966	11,516	17,945,237	220,585	218,743
	unk	0.0%	0.0189%	0.0347%	0.0%	0.0372%	0.0261%
Chinese	sentence	247,829	1,000	1,000	962,986	10,000	10,000
	token	4,733,603	32,807	9,462	13,507,680	154,431	153,875
	unk	0.0%	0.0183%	0.0%	0.0%	0.0246%	0.0227%

4.2 Main Results

Table 2 shows the comparative experimental results of our method and different baselines. MANN [2] adds a memory module on the basis of RNN, but the effect is still far behind the Transformer. TM-augmented [1] uses monolingual corpus to build translation memory and augments the NMT model with a learnable cross-lingual memory retriever, which performs better on low-resource datasets because the large-scale monolingual corpus can compensate for the model's own under training in low-resource scenarios. kNN-MT [7] constructs a token-level

[1] https://github.com/pytorch/fairseq.

Table 2. Comparison experiments of different memory enhancement models.

Models	CCMT2019		CCMT2022	
	valid	test	valid	test
Transformer [14]	27.85	36.56	34.69	36.81
MANN [2]	25.30	34.92	32.42	34.26
TM-augmented [1]	31.21	**43.84**	35.74	36.51
*k*NN-MT [7]	31.37	42.06	36.02	37.58
Adaptive *k*NN-MT [19]	32.52	42.68	36.49	37.81
Fast *k*NN-MT [12]	30.11	41.24	35.09	36.24
Ours	**34.09**	43.78	**37.26**	**39.05**

memory module to guide model generation by retrieval during decoding, but the optimal choice of k is different when using different data stores, leading to poor robustness and generalizability of the method. Adaptive kNN-MT [19] trains a meta-k network by artificially constructing features to generate the nearest neighbor hyper-parameter k. It performs well in various translation tasks. Fast kNN-MT [12] introduces hierarchical retrieval to improve decoding efficiency, but has certain damage to performance. Our method utilizes a Monte Carlo non-parametric dynamic fusion method to further improves the model robustness. Meanwhile, we introduce a confidence-based gating mechanism to accelerate the decoding, so our method obtains consistent improvement in all scenarios.

4.3 Ablation Study

To verify the effect of different components on the model performance, this paper conducts ablation experiments based on Transformer, and the experimental results are shown in Table 3. It is clear that memory module plays a critical role, in the CCMT2019 low-resource Mongolian-Chinese translation, there is a maximum of 5 BLEU improvements, while in the CCMT2022 high-resource scenarios, there is an average of less than 2 BLEU improvements, indicating that the improvement of the memory module to the model is affected by the model's own capabilities, the stronger the model capability, the smaller the additional achievements of the memory module on the model. Since the test set of CCMT2019 is mostly simple and short sentences, while the validation set has more long difficult sentences. Therefore, the improvement rate on the valid set is not as large as that on the test set, which also reflects the effectiveness of our method in complex translation scenarios to a certain extent. Line 4 represents the utilize of Monte Carlo non-parametric fusion on ordinary kNN-MT, which also has some improvement, indicating the effectiveness of this algorithm. The introduction of the confidence network is also shown to be benefit of improving performance (Line 3), the reason is that it can calibrate the confidence estimates of the model itself during training, mitigating the confidence bias in the testing phase due to exposure bias. Moreover, it also has a cumulative effect on translation results when combined with Monte Carlo fusion (Line 5).

Table 3. The results of ablation study, "○" means utilize this method and "×" means not. MM, MC and CE represent Memory Module, Monte Carlo and Confidence Estimation respectively.

ID	Method			CCMT2019		CCMT2022	
	MM	MC	CE	valid	test	valid	test
1	×	×	×	27.85	36.56	34.69	36.81
2	○	×	×	31.19	42.29	36.24	37.69
3	×	×	○	28.01	36.74	34.83	36.94
4	○	○	×	33.71	42.96	36.74	38.52
5	○	○	○	**34.09**	**43.78**	**37.26**	**39.05**

4.4 Effect of Memory Module Capacity and Threshold *c*

Fig. 2 shows the effect of memory module capacity. It can be seen that the translation quality improves with the increase of the external memory module size, but for memory modules containing tens of millions of tokens, the retrieval speed slows down with the increase of memory module size. It also demonstrates that the model is not necessary to be retrained when encountering new training data, and directly storing the data in the memory module can also improve the translation performance. In addition, the external memory module can significantly improve the translation results in low-resource scenarios. For middle-high resource scenarios, there is a very obvious bottleneck in the improvement rate. After reaching this value, the improvement in translation effect brought about by increasing the memory module capacity is far less than the negative impact on slower retrieval speed. Therefore, for middle-high resource scenarios, it is necessary to balance the direct relationship between memory module capacity and translation speed.

To explore whether the model can improve the translation ability for unfamiliar data by modifying the memory module when it encounters new data, we design a test of an extreme scenario and use the model trained on the CCMT2019 dataset to translate the CCMT2022 test set. It can be seen from Table 4 that after adding the test set to the external memory module, the model translation ability for this data has been significantly improved, which proves that the model can be updated by storing unfamiliar data into the external memory module. The translation quality improved significantly after adding a large amount of training data into the memory module, indicating that the performance of "small" models can also be improved by increasing the capacity of the memory module rather than retraining on a large amount of data. We explore the model performance

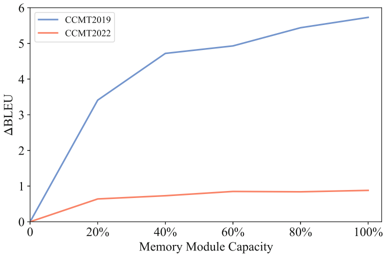

Fig. 2. The effect of memory module capacity on translation quality.

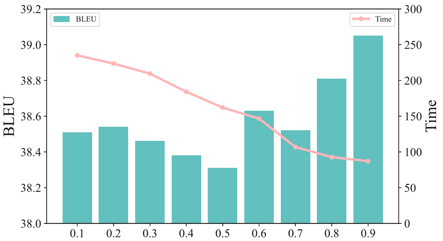

Fig. 3. Effect of different thresholds on BLEU and total translation time.

and decoding time under different threshold c on CCMT2022, and the results are shown in Fig. 3. It can be seen that the model quality does not fluctuate greatly under different threshold settings, but the total decoding time of the model decreases as the threshold keeps increasing. It indicates that our method can't affect the translation quality of the model too much while optimizing the decoding efficiency.

Table 4. The effect of updating memory module on translation quality.

Model	Memory Module	BLEU
CCMT2019	–	28.36
	CCMT2022 test set	29.71
	CCMT2022 train set	32.68

4.5 Decoding Efficiency Verification in Different Dimensions

This paper measures the decoding efficiency from three dimensions on the test set of CCMT2022, namely the total translation time, the number of sentences translated per second, and the number of tokens translated per second. Experimental results are shown in Fig. 4. The decoding efficiency of the proposed method is about 2 times that of the original method when the retrieval nearest neighbor number k is small. With the increase of k, the improvement range of the proposed method becomes smaller and smaller. However, since the optimal k of the experimental model is less than 24, the decoding efficiency of this method is better than that of the traditional method in general. Moreover, this method does not affect the model quality while improving the decoding efficiency.

4.6 Domain Adaptation and Robustness Analysis

To verify the effectiveness of our approach in domain adaption, we according to Adaptive kNN-MT conduct experiments in four domains including, IT (I),

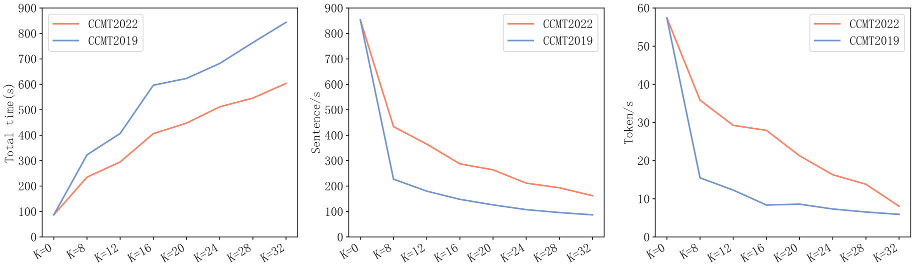

Fig. 4. Decoding efficiency comparison chart.

Medical (M), Koran (K) and Laws (L) in German-English. The main results are shown in Table 5, where the hyperparameter k are 8, 4, 8 and 4, respectively, and this paper's approach has obtained consistency improvement in all domains. In IT→Medical (I→M) setting, we use the IT domain hyperparameters and the memory module to translate the medical test set. The kNN-MT encounters drastic performance degradation due to the retrieved "neighbors" are highly noisy. In contrast, Adaptive kNN-MT can filter out noises and therefore prevent performance degradation as much as possible. The performance of this paper is further improved by Monte Carlo nonparametric fusion and gating mechanism compared to Adaptive kNN-MT.

Table 5. Our method on domain adaptive experiments and robustness evaluation.

Model	IT	Medical	Koran	Laws	I→M	M→I
Transformer	32.05	36.25	14.38	41.78	36.25	32.05
kNN-MT	36.68	51.27	17.55	57.55	15.81	12.31
Adaptive kNN-MT	39.22	51.84	18.25	58.46	24.62	20.14
Ours	39.43	52.07	18.46	58.72	24.93	20.48

5 Conclusion

In this paper, we propose a non-parametric method based on Monte Carlo to dynamically integrate memory module's prediction and NMT prediction, which improves model performance and robustness in various scenarios. In view of the slow retrieval speed in kNN-MT, this paper proposes a gating mechanism based on confidence estimation to filter the unnecessary retrieval behavior of the model, so as to improve the decoding efficiency. Our method is effective in low resource scenarios, but marginal utility appears in high resource scenarios. Therefore, future work will further study the optimization and promotion in high resource tasks.

References

1. Cai, D., Wang, Y., Li, H., Lam, W., Liu, L.: 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, ACL/IJCNLP 2021, (Volume 1: Long Papers), Virtual Event, 1–6 August 2021, pp. 7307–7318. Association for Computational Linguistics (2021)
2. Collier, M., Beel, J.: Memory-augmented neural networks for machine translation. In: Proceedings of Machine Translation Summit XVII Volume 1: Research Track, MTSummit 2019, Dublin, Ireland, 19–23 August 2019, pp. 172–181. European Association for Machine Translation (2019)
3. DeVries, T., Taylor, G.W.: Learning confidence for out-of-distribution detection in neural networks. CoRR abs/1802.04865
4. Eriguchi, A., Rarrick, S., Matsushita, H.: Combining translation memory with neural machine translation. In: Proceedings of the 6th Workshop on Asian Translation, WAT@EMNLP-IJCNLP 2019, Hong Kong, China, 4 November 2019, pp. 123–130. Association for Computational Linguistics (2019)
5. Hassan, H., et al.: Achieving human parity on automatic Chinese to English news translation. CoRR abs/1803.05567
6. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**(3), 535–547
7. Khandelwal, U., Fan, A., Jurafsky, D., Zettlemoyer, L., Lewis, M.: Nearest neighbor machine translation. In: 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, 3–7 May 2021. OpenReview.net
8. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: 3rd International Conference on Learning Representations, ICLR 2015 (2015)
9. Lu, Y., Zeng, J., Zhang, J., Wu, S., Li, M.: Learning confidence for transformer-based neural machine translation. In: Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, 22–27 May 2022, pp. 2353–2364. Association for Computational Linguistics (2022)
10. Martins, P.H., Marinho, Z., Martins, A.F.T.: Chunk-based nearest neighbor machine translation. In: Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, EMNLP 2022, Abu Dhabi, United Arab Emirates, 7–11 December 2022, pp. 4228–4245. Association for Computational Linguistics (2022)
11. Martins, P.H., Marinho, Z., Martins, A.F.T.: Efficient machine translation domain adaptation. CoRR abs/2204.12608. https://doi.org/10.48550/arXiv.2204.12608
12. Meng, Y., et al.: Fast nearest neighbor machine translation. In: Findings of the Association for Computational Linguistics: ACL 2022, Dublin, Ireland, 22–27 May 2022, pp. 555–565. Association for Computational Linguistics (2022)
13. Post, M.: A call for clarity in reporting BLEU scores. In: Proceedings of the Third Conference on Machine Translation: Research Papers, WMT 2018, pp. 186–191 (2018)
14. Vaswani, A., et al.: Attention is all you need. In: Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems (2017)
15. Wang, D., Fan, K., Chen, B., Xiong, D.: Efficient cluster-based k-nearest-neighbor machine translation. In: Proceedings of the 60th Annual Meeting of the Association

for Computational Linguistics (Volume 1: Long Papers), ACL 2022, Dublin, Ireland, 22–27 May 2022, pp. 2175–2187. Association for Computational Linguistics (2022)

16. Wang, D., Wei, H., Zhang, Z., Huang, S., Xie, J., Chen, J.: Non-parametric online learning from human feedback for neural machine translation. In: Thirty-Sixth AAAI Conference on Artificial Intelligence, AAAI 2022, Thirty-Fourth Conference on Innovative Applications of Artificial Intelligence, IAAI 2022, The Twelveth Symposium on Educational Advances in Artificial Intelligence, EAAI 2022 Virtual Event, February 22–1 March 2022, pp. 11431–11439. AAAI Press (2022)
17. Yang, Z., Chen, W., Wang, F., Xu, B.: Improving neural machine translation with conditional sequence generative adversarial nets. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, 1–6 June 2018, vol. 1 (Long Papers), pp. 1346–1355. Association for Computational Linguistics (2018)
18. Zhang, J., Utiyama, M., Sumita, E., Neubig, G., Nakamura, S.: Guiding neural machine translation with retrieved translation pieces. In: Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2018, New Orleans, Louisiana, USA, 1–6 June 2018, vol. 1 (Long Papers), pp. 1325–1335. Association for Computational Linguistics (2018)
19. Zheng, X., et al.: 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, ACL/IJCNLP 2021, (Volume 2: Short Papers), Virtual Event, 1–6 August 2021, pp. 368–374. Association for Computational Linguistics (2021)
20. Zheng, X., et al.: Non-parametric unsupervised domain adaptation for neural machine translation. In: Findings of the Association for Computational Linguistics: EMNLP 2021, Virtual Event/Punta Cana, Dominican Republic, 16–20 November 2021, pp. 4234–4241. Association for Computational Linguistics (2021)