Dynamic Fusion Nearest Neighbor Machine Translation via Dempster-Shafer Theory

Zongheng Yang[1,2], Hongxu Hou[1,2(✉)], Shuo Sun[1,2], Nier Wu[1,2], Yisong Wang[1,2], Weichen Jian[1,2], and Pengcong Wang[1,2]

[1] College of Computer Science, Inner Mongolia University, Hohhot, China
cshhx@imu.edu.cn
[2] Inner Mongolia Key Laboratory of Mongolian Information Processing Technology, Hohhot, China

Abstract. kNN-MT has been recently proposed, uses a token-level k-nearest neighbor approach to retrieve similar sentences, obtaining knowledge guidance from an external memory module, and then combined with the prediction results of the translation model, which greatly improves the accuracy of machine translation. However, kNN-MT uses simple linear interpolation in the fusion of retrieval probability and translation probability, which can not dynamically adjust the fusion ratio according to the matching degree of the retrieved sentences. Moreover, different fusion ratios need to be explored in different translation scenarios, and the translation effect will be affected when the retrieved sentences have a low matching degree or contain noise. In this paper, we propose an approach via Dempster-Shafer theory (DST) to dynamically fuse different probability distributions to suit different scenarios. We demonstrate that our approach is more significantly improved and more robust than the traditional kNN-MT, and we explore the application of kNN-MT in low-resource translation scenarios for the first time.

Keywords: kNN-MT · Dynamic fusion · Translation

1 Introduction

Over the past few years, with the development of deep learning, neural machine translation has come a long way. In order to further improve the translation accuracy, more and more researches have started to express the training data as some kind of external knowledge rather than as model parameters, which is called non-parametric method. Since this method requires search to obtain external knowledge, it is also called search-based model. The representative methods are as follows: Nearest neighbor language models (kNN-LM) [1], which introduces kNN to the language model for the first time and gains tremendous enhancements; k-nearest-neighbor machine translation (kNN-MT) [2], which extends kNN-LM to

T. Xiao and J. Pino (Eds.): CCMT 2022, CCIS 1671, pp. 82–92, 2022.
https://doi.org/10.1007/978-981-19-7960-6_9

translation model, has made a qualitative leap in bilingual translation, multilingual translation, and especially domain adaptation translation tasks compared with traditional methods; As well as Adaptive kNN-MT implemented by [3] on this basis, a meta-k network is trained by artificially constructing features for generating the number of nearest neighbors k, instead of artificially specifying them; And Fast kNN-MT [4] introduces hierarchical retrieval to improve the retrieval efficiency thus improving the slow translation speed of kNN-MT.

kNN-MT bulids an external memory module on top of the ordinary NMT, storing the context representation of the corresponding sentence as well as the target word. The idea of kNN-MT is to retrieve sentences similar to the current sentence in the memory module when translating the current word, and get reference and guidance from the translation memory by the words corresponding to the similar sentences. Then it is fused with the translation result of NMT to get the final result.

Although kNN-MT has demonstrated its powerful capability in high-resource languages as well as domain adaptation, there are still two problems. On the one hand, kNN-MT has not been studied in low-resource scenarios due to its particular reliance on the representational power of pre-trained translation models and the retrieval effect of similar sentences. On the other hand, in the fusion of NMT with an external memory module, the fusion ratio is controlled by a hyperparameter λ, i.e., how much information the NMT model obtains from the external memory module. However, it poses some problems, due to the long-tail effect of the dataset, some sentences have more similar sentences while some sentences have less similar sentences. Using the same fusion ratio for all data will cause the problem that some sentences do not acquire enough information and some sentences introduce noise. We illustrate this with a concrete example in Fig. 1. Moreover, it is experimentally demonstrated that the model translation results are very sensitive to the selection of hyperparameter λ, which affects the robustness of the model.

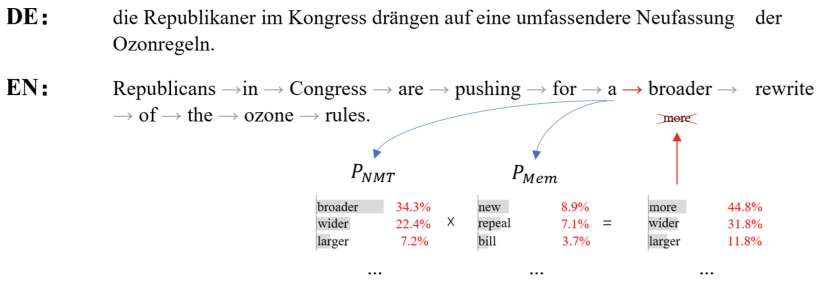

Fig. 1. Example of failure of probability interpolation between p_{NMT} and p_{Mem}, while translating DE-EN.

To solve this problem, we propose a dynamic fusion method via Dempster-Shafer theory, which drops the fixed fusion method with linear interpolation, and

gives different fusion results for different retrieval probabilities and translation probabilities. The problem of high confidence in retrieval probability, but too low fusion ratio, i.e., better prediction of retrieval probability, but biased final translation result due to too low fusion ratio, and vice versa, is alleviated. Moreover, our method improves the robustness of the model to cope with translation in complex scenarios. More importantly, we explore the application of kNN-MT in low-resource translation scenarios for the first time, demonstrating the effectiveness of non-parametric methods in low-resource scenarios. We validate the effectiveness of our methodology for multi-domain datasets, including IT, Medical, Koran, Law, and the CCMT'19 Mongolian-Chinese low-resource dataset. Our method obtains an increase of 0.41-1.89 BLUE, and the robustness of the model is improved.

2 Background

The main approach of kNN-MT involves the building of memory modules and the fusion of external knowledge with the predicted results of the NMT model. In terms of memory module construction, unlike [5] and [6] which construct sentence-level and fragment-level memory datastore, kNN-MT constructs token-level memory datastore. Its advantage is better retrieval and higher matching, but the memory module size is the total number of tokens in the target language, which leads to low retrieval efficiency. In terms of construction method, kNN-MT selects an offline construction method, therefore a pre-trained model with strong knowledge representation capability is required. The memory module is stored as a key-value pair of a context vector and a target token, and is constructed by feeding the training data into the model in a single forward pass. Given a bilingual corpus $(x, y) \in (\mathcal{X}, \mathcal{Y})$ the decoder decodes y_t based on the source language x and the words $y_{<t}$ that have been generated. Assuming that the hidden layer state of the pre-trained model is $f(x, y_{<t})$, the key of the datastore is $f(x, y_{<t})$ and the value is y_t, then the construction process is:

$$(\mathcal{K}, \mathcal{V}) = \{(f(x, y_{<t}), y_t), \forall y_t \in y \mid (x, y) \in (\mathcal{X}, \mathcal{Y})\} \tag{1}$$

Once the memory module is constructed, the similar sentences can be retrieved when decoding, and the token corresponding to the similar sentences can be used to obtain a retrieval probability, i.e., the retrieval probability p_{Mem} given by the memory module through historical data.

$$p_{Mem}(y_i \mid x, \hat{y}_{1:i-1}) \propto \sum_{(k_i, v_i \in \mathcal{N})} \mathbb{1}_{y_i = v_i} exp\left(-\frac{-d(k_j, f(x, \hat{y}_{1:i-1}))}{T}\right) \tag{2}$$

The retrieval probability represent external knowledge guidance, and kNN-MT fuses the external knowledge with the model knowledge by simple linear interpolation to obtain the final probability distribution.

$$p(y_t \mid x, \hat{y}_{1:i-1}) = \lambda p_{NMT}(y_t \mid y_{<t}, x) + (1 - \lambda) p_{Mem}(y_t \mid y_{<t}) \tag{3}$$

3 Method

In this section, we mainly introduce our proposed method, and our method is mainly applied in the inference stage of the model. We discard linear interpolation and use DST (Dempster-Shafer theory) in the fusion process of p_{NMT} and p_{Mem}, and our method is shown in Fig. 2. Since p_{Mem} only generates probabilities for a few relevant words of the similar neighbors in the actual calculation process, and the probabilities of other irrelevant words are all 0, resulting in a very hard distribution of p_{Mem}, and more 0 probabilities will have a very significant impact on the DST results, so we use label smoothing for p_{Mem} to make the distribution of p_{Mem} smoother.

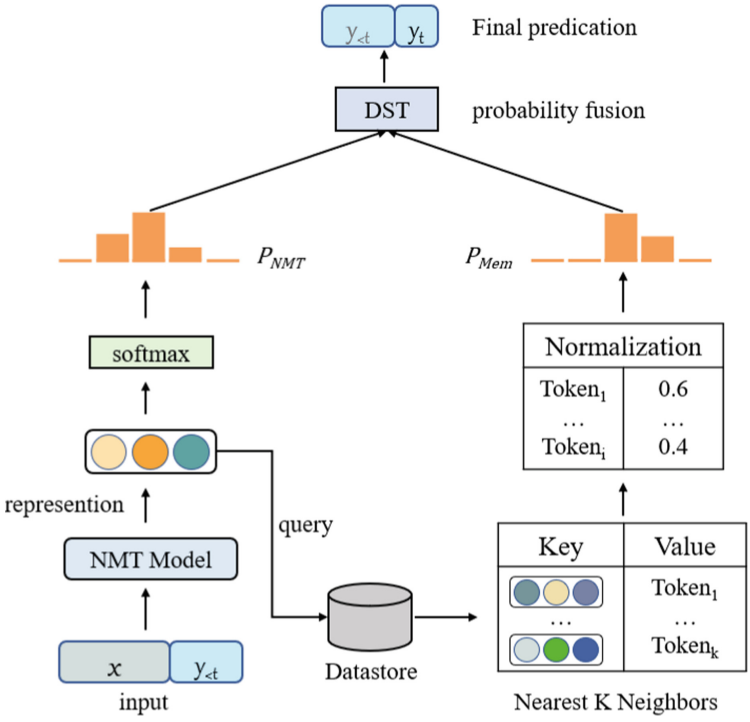

Fig. 2. Schematic diagram of our approach, the retrieval process occurs at the decoder, where similar sentences are retrieved in the memory module based on the context vector. The retrieval probability is obtained by normalizing the target token and then dynamically fused with the translation probability using the DST algorithm.

3.1 Dempster-Shafer Theory

Dempster-Shafer theory [7] is a generalization of probability theory and a very effective method for data fusion. DST extends the basic event space in probabil-

ity theory to power sets of basic elements by replacing a single probability value of a basic element with a probability range. DST is based on the mathematical theory proposed by Demster and Schaeffer, and is a more general formulation of Bayesian theory. DST proposes a framework that can be used to represent incomplete knowledge and update credibility. If a set is defined as $\Theta = \{\theta_1, \theta_2, ..., \theta_N\}$ and all elements in the set are independent and mutually exclusive, Θ is called the frame of discernment framework. Under this premise, the DST combination rules are provided.

Let m_1 and m_2 be the two probability assignment functions on the same discernment framework. The corresponding focal elements are $A_i\,(i = 1, 2, ..., k)$ and $B_j\,(j = 1, 2, ..., l)$, respectively, and the new probability assignment (BPA) functions after the combination is denoted by m. Then the DST combination rule can be expressed as the following form:

$$m(A) = m_1(A) \oplus m_2(A) \begin{cases} m(\phi) = 0 \\ \frac{1}{1-k} \sum\limits_{A_i \cap B_j = A} m_1(A_i) m_2(B_j) \end{cases} \tag{4}$$

Dempster-Shafer theory has been widely used to deal with problems with uncertainty or imprecision. Because it can integrate different algorithms based on its basic probability assignment framework to improve the reliability of the results. In this paper, we use evidence theory to execute data fusion for p_{NMT} and p_{Mem}, where m_1 in Eq. 4 is p_{NMT} and m_2 is p_{Mem}.

3.2 Label Smoothing

Label Smoothing [8] is a widely used regularization technique in machine translation. LS penalizes the high confidence in the hard target to introduce noise to the label and change the hard target into a soft target. The idea of label smoothing is simple: the token corresponding to the ground truth should not have exclusive access to all probabilities; other tokens should have a chance to be used as ground truth. In parameter estimation of complex models, it is often necessary to assign some probabilities to unseen or low-frequency events to ensure the better generalization ability of the model. For the specific implementation, label smoothing uses an additional distribution q which is a uniform distribution over the vocabulary V, i.e., $q_k = \frac{1}{v}$, where q_k denotes the kth dimension of the distribution. The distribution of final result is then redefined as a linear interpolation of y_j and q:

$$y_j^{ls} = (1 - \alpha) \cdot y_j + \alpha \cdot q \tag{5}$$

Here, α denotes a coefficient to control the importance of the distribution q, and y_j^{ls} denotes the learning target after using label smoothing. The schematic diagram is shown in Fig. 3.

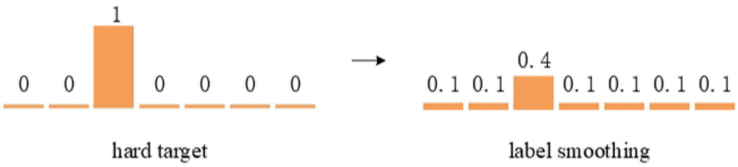

Fig. 3. Targets with Label Smoothing when $\alpha = 0.1$.

Label smoothing can also be seen as an adaptation of the loss function with the introduction of additional prior knowledge (i.e., the part related to q). But this prior knowledge is not fused with the original loss function by means of linear interpolation.

The process of generating the final probability can be summarized by the following procedure, where the LS denotes a label smoothing, DST denotes Dempster-Shafer theory, p_{Mem} denotes the retrieval probability obtained from the memory module, and p_{NMT} denotes the translation probability of the NMT model.

$$p\left(y_t \mid y_{<t}\right) = DST\left(p_{NMT}, LS\left(p_{Mem}\right)\right) \tag{6}$$

4 Experiment

We validate the effectiveness of our method in two translation scenarios: (1) domain adaptation. (2) Mongolian-Chinese low resource language.

4.1 Experimental Setup

Data. We use the following datasets for training and evaluation:

MULTI-DOMAINS: We use the multi-domains dataset [9], re-split by [10] for the domain adaptation experiments. It includes German-English parallel data for train/valid/test sets in four domains: **Medical**, **Law**, **IT** and **Koran**. The sentence statistics of MULTI-DOMAINS datasets are illustrated in Table 1.

Table 1. Statistics of dataset in different domains.

	Train	Valid	Test
IT	222,927	2,000	2,000
Medical	248,009	2,000	2,000
Koran	17,982	2,000	2,000
Laws	467,309	2,000	2,000

Low-resource: We use the CCMT'19 Mongolian-Chinese dataset to evaluate the performance of our method in low-resource scenarios. The bilingual parallel

corpus comes from a comprehensive field, including daily conversations, government documents, government work reports, laws and regulations, etc. The sentence statistics of Mongolian-Chinese dataset are illustrated in Table 2.

Table 2. Statistics of dataset in Mongolian-Chinese.

	Train	Valid	Test
Mo-Zh	247,829	1,000	1,000

Models. For the domain adaptation experiments, we use the WMT'19 German-English news translation task winner [11], available via the FAIRSEQ library [12]. It is a Transformer encoder-decoder model [13] with 6 layers, 1,024 dimensional representations, 8,192 dimensional feedforward layers and 8 attention heads. Apart from WMT'19 training data, this model is trained on over 10 billion tokens of back translation data and fine-tuned on newstest test sets from years prior to 2018.

For low-resource translation, we train a Mongolian-Chinese translation model based transformer. The corpus is subworded using subword-nmt[1] [14], using a Adam optimizer [15] with a warmup step of 10,000, epoch of 30 and setting early stop. Other settings are kept the same as transformer-base.

Our experiments are based on the fairseq[2] sequence modeling toolkit to train NMT models, using the faiss[3] [16] toolkit for external memory module construction and high-speed retrieval. We implement our approach on the open source code of adaptive-knn-mt[4], which implements the original kNN-MT based on fairseq and has a good code structure.

4.2 Result and Analysis

For the domain adaptive task, the main results are shown in Table 3. Consistency improvement is obtained for all four domains of our method. The BLEU scores are improved by 1.89, 0.51, 0.48, and 0.55 compared to kNN-MT. The minimum improvement is in the Koran domain and the highest is in the IT domain.

For the low-resource task, the experimental results are shown in Table 4, and it can be found that kNN-MT can also obtain a huge improvement on the translation result in the low-resource domain, and our method is also improved compared with kNN-MT.

Analysis. Compared with kNN-MT our method is more flexible in the probabilistic fusion stage, which is reflected in the results to obtain a consistent improvement of BLEU. The biggest improvement in the domain adaptive experiments is in the IT domain, and by analyzing the translation results we speculate

[1] https://github.com/rsennrich/subword-nmt.
[2] https://github.com/pytorch/fairseq.
[3] https://github.com/facebookresearch/faiss.
[4] https://github.com/zhengxxn/adaptive-knn-mt.

Table 3. BLEU scores of Base NMT model, kNN-MT and our method on domain adaptive experiments with hyperparameters k of 8, 4, 8 and 4, respectively. The linear interpolation ratios α for kNN-MT are 0.7, 0.8, 0.7, and 0.7.

Model	IT	Medical	Koran	Laws
Base-NMT	32.05	36.25	14.38	41.78
kNN-MT	36.68	51.27	17.55	57.55
Ours	**38.57**	**51.78**	**18.03**	**58.1**

Table 4. BLEU scores of Base NMT model, kNN-MT and our method on Mongolian-Chinese low-resource experiments with hyperparameter $k = 4$.

Model	Valid	Test
Base-NMT	27.85	36.56
kNN-MT	31.19	42.29
Ours	**33.64**	**42.77**

that it may be due to the presence of more low-frequency special nouns in the IT domain. kNN-MT introduces noise in the retrieval process, while our method performs better in the translation of low-frequency words.

In the low-resource scenario since the test sets of Mongolian-Chinese are mostly simple short sentences, while the valid sets have more long and difficult sentences. Therefore, the improvement of our method on the test sets is not as large as that on the valid sets, which also reflects the effectiveness of our method in complex translation scenarios to some extent. Since DST can produce different results according to different probabilities and expose more information after using label smoothing for p_{Mem}, it increases the generalization and robustness of the model.

4.3 Robustness

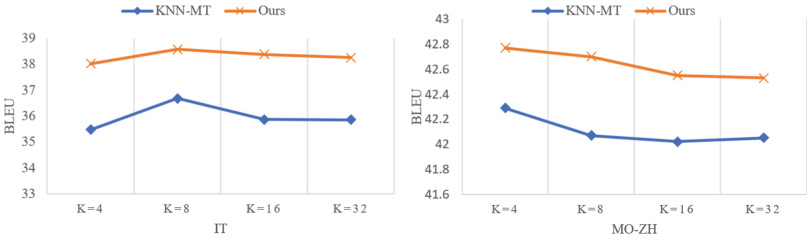

Fig. 4. Robustness experiments of kNN-MT and our method at different hyperparameters k.

To verify the robustness of our method, we test the accuracy of translation under different hyperparameters k. The experimental results are shown in Fig. 4. We find that the BLEU scores of kNN-MT fluctuate more in the case of not optimal k values, indicating that the performance of kNN-MT is more sensitive to the noise brought by k. And the performance of our method is also affected during the increase of k, but with less fluctuation. It indicates that a relatively good performance can be maintained at different noise intensities.

We also evaluate the robustness of our method in the domain-mismatch setting, where we consider a scenario that the user inputs an out-of-domain sentence (e.g. Medical domain) to a domain-specific translation system (e.g. IT domain) to evaluate the robustness of different methods. Specifically, in IT→Medical setting, We use hyperparameters and datastore in the IT domain, and then use Medical test set to test the model with IT datastore. As shown in Table 5, the retrieved results are highly noisy so that the kNN-MT encounters drastic performance degradation. In contrast, our method could filter out some noise and therefore prevent performance degradation as much as possible.

Table 5. Robustness Evaluation, where the test sets are from Medical/IT domains and the datastore are from IT/Medical domains respectively.

Model	IT→Medical	Medical→IT
Base-NMT	36.25	32.05
kNN-MT	15.81	12.31
Ours	**24.1**	**19.56**

4.4 Case Study

	IT	Mo-Zh
Source	Weitere Suchergebnisse anzeigen	ᠪᠤᠴᠤᠬᠤ ᠬᠠᠢᠯᠲᠠᠨ ᠤ <unk>
Reference	Show the next search results	不要 让 机遇 从 我们 身边 溜走 。
Base-NMT	View more search results	不要 让 机遇 从 <unk> 溜走 。
kNN-MT	Show & Location Find additional matches	不要 让 机遇 从 <unk> 溜走 。
Ours	Show more search results	不要 让 机遇 从 我们 身边 溜走 。

Fig. 5. Translation examples of different systems in IT domain and Mongolian-Chinese.

As shown in Fig. 5, we show examples of translations in the IT domain and Mongolian-Chinese. We can observe that kNN-MT can produce mistranslations in some cases, and our method can generate translations with more fidelity and

fluency in this case. Moreover, our method can alleviate the $\langle unk \rangle$ problem to a certain extent. In the Mongolian-Chinese example, both the Base NMT model and kNN-MT can not translate correctly when the corpus contains $\langle unk \rangle$, which also shows that our method is more robust and higher error tolerance.

5 Conclusion

In this paper we propose dynamic fusion of kNN-MT. By using Dempster-Shafer theory instead of fixed linear interpolation to dynamically fuse the two probability distributions from NMT model and memory modules. Through experiments in domain adaptation, we verify that our method has some improvement on kNN-MT and validate that our method is more robust. In addition, we explore the possibility of applying kNN-MT in low-resource scenarios for the first time. In the future, we will deeply explore the application of non-parametric methods in low-resource scenarios.

References

1. Khandelwal, U., Levy, O., Jurafsky, D., Zettlemoyer, L., Lewis, M.: Generalization through memorization: nearest neighbor language models. In: 8th International Conference on Learning Representations, ICLR 2020, Addis Ababa, Ethiopia, 26–30 April 2020. OpenReview.net (2020). https://doi.org/10.48550/arXiv.1911.00172
2. 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 (2021). https://doi.org/10.48550/arXiv.2010.00710
3. Zheng, X., et al.: Adaptive nearest neighbor machine translation. In: Zong, C., Xia, F., Li, W., Navigli, R. (eds.) 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). https://doi.org/10.18653/v1/2021.acl-short.47
4. Meng, Y., et al.: Fast nearest neighbor machine translation. CoRR abs/2105.14528 (2021). https://arxiv.org/abs/2105.14528
5. Eriguchi, A., Rarrick, S., Matsushita, H.: Combining translation memory with neural machine translation. In: Nakazawa, T., et al. (eds.) 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). https://doi.org/10.18653/v1/D19-5214
6. Zhang, J., Utiyama, M., Sumita, E., Neubig, G., Nakamura, S.: Guiding neural machine translation with retrieved translation pieces. In: Walker, M.A., Ji, H., Stent, A. (eds.) 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). https://doi.org/10.18653/v1/n18-1120

7. Dempster, A.P.: Upper and lower probabilities induced by a multivalued mapping. In: Yager, R.R., Liu, L. (eds.) Classic Works of the Dempster-Shafer Theory of Belief Functions. Studies in Fuzziness and Soft Computing, vol. 219, pp. 57–72. Springer, Heidelberg (2008). https://doi.org/10.1007/978-3-540-44792-4_3
8. Szegedy, C., Vanhoucke, V., Ioffe, S., Shlens, J., Wojna, Z.: Rethinking the inception architecture for computer vision. In: 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, 27–30 June 2016, pp. 2818–2826. IEEE Computer Society (2016). https://doi.org/10.1109/CVPR.2016.308
9. Koehn, P., Knowles, R.: Six challenges for neural machine translation. In: Luong, T., Birch, A., Neubig, G., Finch, A.M. (eds.) Proceedings of the First Workshop on Neural Machine Translation, NMT@ACL 2017, Vancouver, Canada, 4 August 2017, pp. 28–39. Association for Computational Linguistics (2017). https://doi.org/10.18653/v1/w17-3204
10. Aharoni, R., Goldberg, Y.: Unsupervised domain clusters in pretrained language models. In: Jurafsky, D., Chai, J., Schluter, N., Tetreault, J.R. (eds.) Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics, ACL 2020, Online, 5–10 July 2020, pp. 7747–7763. Association for Computational Linguistics (2020). https://doi.org/10.18653/v1/2020.acl-main.692
11. Ng, N., Yee, K., Baevski, A., Ott, M., Auli, M., Edunov, S.: Facebook fair's WMT19 news translation task submission. In: Bojar, O., et al. (eds.) Proceedings of the Fourth Conference on Machine Translation, WMT 2019, Florence, Italy, 1–2 August 2019, - Volume 2: Shared Task Papers, Day 1, pp. 314–319. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/w19-5333
12. Ott, M., et al.: fairseq: a fast, extensible toolkit for sequence modeling. In: Ammar, W., Louis, A., Mostafazadeh, N. (eds.) Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, NAACL-HLT 2019, Minneapolis, MN, USA, 2–7 June 2019, Demonstrations, pp. 48–53. Association for Computational Linguistics (2019). https://doi.org/10.18653/v1/n19-4009
13. Vaswani, A., et al.: Attention is all you need. In: Guyon, I., et al. (eds.) Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, 4–9 December 2017, Long Beach, CA, USA, pp. 5998–6008 (2017). https://proceedings.neurips.cc/paper/2017/hash/3f5ee243547dee91fbd053c1c4a845aa-Abstract.html
14. Sennrich, R., Haddow, B., Birch, A.: Neural machine translation of rare words with subword units. In: Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics, ACL 2016, 7–12 August 2016, Berlin, Germany, Volume 1: Long Papers. The Association for Computer Linguistics (2016). https://doi.org/10.18653/v1/p16-1162
15. Kingma, D.P., Ba, J.: Adam: a method for stochastic optimization. In: Bengio, Y., LeCun, Y. (eds.) 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, 7–9 May 2015, Conference Track Proceedings (2015). http://arxiv.org/abs/1412.6980
16. Johnson, J., Douze, M., Jégou, H.: Billion-scale similarity search with GPUs. IEEE Trans. Big Data **7**(3), 535–547 (2021). https://doi.org/10.1109/TBDATA.2019.2921572