Low Resource Retrieval Augmented Adaptive Neural Machine Translation

Harsha Vardhan * **Anurag Beniwal** **Narayanan Sadagopan** **Swair Shah***
Amazon
vvl@andrew.cmu.edu, {beanurag,sdgpn,shahswai}@amazon.com

Abstract

We propose KNN-Kmeans MT, a sample efficient algorithm that improves retrieval based augmentation performance in low resource settings by adding an additional K-means filtering layer after the KNN step. KNN-Kmeans MT like its predecessor retrieval augmented machine translation approaches (Khandelwal et al. [2020]) doesn't require any additional training and outperforms the existing methods in low resource settings. The additional K-means step makes the model more robust to noise. We benchmark our proposed approach on EMEA and JTRC-Acquis dataset and see 0.2 points improvement in BLEU score on an average in low resource settings. More importantly, the trend of improvement from high to low resource setting is consistently obvious across both the datasets. We conjecture that the observed improvement is a consequence of eliminating bad neighbors as their retrieval databases are small and retrieving a fixed number of neighbors leads to adding noise to the model. The simplicity of the approach makes it a promising direction in opening up the use of retrieval augmentation in low resource setting.

1 Introduction

Neural machine translation (NMT) models have found wide-scale production use in services like Amazon Translate, Google Translate, Bing Translate etc. Invariably these models perform poorly in certain domains (e.g. medical texts, legal documents) or they make certain translation errors due to lack of training data for specific use cases Chu and Wang [2018]. Traditionally these models have been improved by re-training on more data. Recently retrieval based augmentation has shown good results in improving the performance of NMT models Khandelwal et al. [2020], Wang et al. [2021], Zheng et al. [2021]. The retrieval augmentation approach obviates the need for retraining to improve model performance. This performance boost is achieved by allowing the model to look up a large database of source-translation context and next token pairs. This is a viable approach in certain use cases where one can find a large database of source-translation pairs to construct the lookup database and the cost of doing a lookup in a large at every token is not a concern. In other use cases, e.g. where one needs to fix certain kind of translation errors, or in a domain with small amount of data, it is difficult to find large set of source-translation pairs to construct the database. This raises an interesting question about the sample complexity of the retrieval-augmented NMT models. How can we improve the efficacy of the NMT models via retrieval augmentation in a low resource setting? In case of retrieval augmented NMT models, low resource setting means restricting the size of the database. KNN-NMT Khandelwal et al. [2020] uses a nearest neighbor classifier at token-level over a large set of cached examples combined with neural machine trasnlation model to produce the final corrected output. Especially, KNN-MT Khandelwal et al. [2020] and the follow-up works as Zheng et al. [2021], Meng et al. [2021], the trend is to put the entire dataset into the database for search. When we restrict the size of the database the retrieved nearest neighbors are prone to

* Equal contribution

36th Conference on Neural Information Processing Systems (NeurIPS 2022).

noise as there are usually only a few relevant examples in the database that can be used to guide the output distribution of the NMT model in the correct way. The idea we explore in this paper aims at reducing this noise by the use of the classic k-means clustering. The clusters help us filter out the less relevant neighbors for any given context. We call the algorithm KNN-Kmeans MT. We verify this experimentally and show better performance than the original KNN-MT in the low resource setting. As retrieval augmented NMT methods involve the hyperparameter λ which determines how much weight is given to the retrieved neighbors as compared to the output of the NMT model the results are sensitive to this parameter. We show that for the best value λ, KNN-Kmeans MT out performs KNN-MT in the low resource setting. To the best of our knowledge we are the first one to explore the low resource setting for retrieval augmented machine translation models.

2 Related Works

Retrieval augmented models have seen a recent surge in interest in the machine learning community Borgeaud et al. [2022] Guu et al. [2020]. Khandelwal et al. introduced the first such approach for machine translation called KNN-MT in Khandelwal et al. [2020]. This approach was upto 3 times slower than the standard NMT model inference due to a nearest neighbor search for every token of translation Khandelwal et al. [2020]. Two main lines of inquiry to improve upon that work involved - making this approach faster He et al. [2021] and making it more accurate by tuning the parameters λ and k Zheng et al. [2021], Wang et al. [2021]. As the research in Khandelwal et al. [2020] motivated the retrieval augmentation for domain adaptation with a large dataset, the literature following in their footsteps mainly focused on this high resource setting. Low resource machine translation is by itself a very active area of research Haddow et al. [2022]. In the current work we bring together these two research directions by exploring the low resource setting in retrieval augmented NMT.

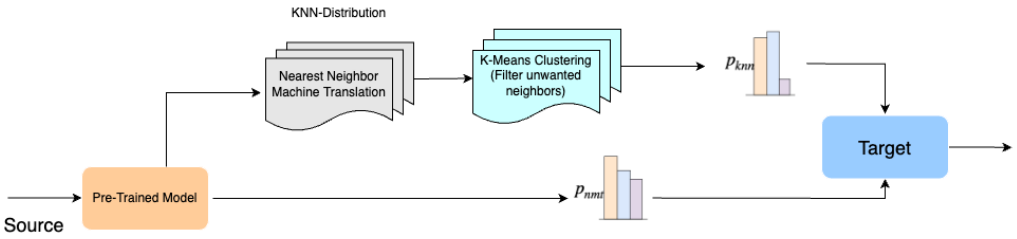

Figure 1: KNN-Kmeans MT pipeline

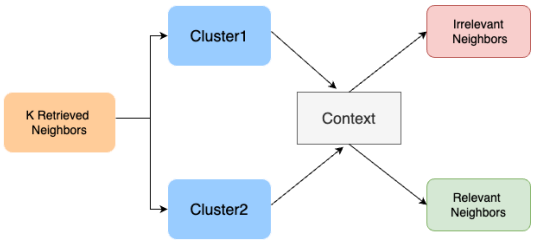

Figure 2: K-Means clustering filter

3 Proposed Approach

The KNN-MT work Khandelwal et al. [2020] showed that increasing the database size improves the performance. This is because the query has a higher chance of fetching relevant nearest neighbors. On contrary, nearest neighbor queries on smaller databases end up fetching noisy data points. In this work we show how to improve retrieval performance in low resource settings. We propose KNN-KMeans MT. We perform unsupervised (K-Means) clustering of the retrieved nearest neighbors

to determine the relevance of the fetched nearest neighbors. The database lookup is followed by a weighting stage where the cluster whose center is the closest to the query vector is deemed more relevant than the other clusters. These neighbors falling outside of the relevant cluster are down weighted which influences the final soft-max distribution obtained from KNN-retrieval.

The database for a parallel text collection of source and target sentences $(\mathcal{S}, \mathcal{T})$ is given as follows:

$$(\mathcal{K}, \mathcal{V}) = \{(f(s, t_{1:i-1}), t_i), \forall t_i \in t \mid (s, t) \in (\mathcal{S}, \mathcal{T})\} \tag{1}$$

where $\mathcal{K}$ is the query stored as key and $\mathcal{V}$ is the corresponding next token. At test time, given a source x, the model outputs a distribution over the vocabulary $p_{MT}(y_i \mid x, \hat{y}_{1:i-1})$ for the target y_i where $\hat{y}$ are the generated tokens. The representation $f(x, \hat{y}_{1:i-1})$ is used to query the database for the k nearest neighbors $\mathcal{N}$ according to L^2 distance d.

Before the retrieved set is converted into a softmax distribution, we perform K-means clustering of the retrieved nearest neighbors to determine the relevance of the fetched nearest neighbors. The database lookup is followed by a weighting stage where "relevant" the cluster whose center is closest to the query vector is deemed more relevant than the other cluster. These "irrelevant" neighbors falling outside of the relevant cluster are down weighted by w and relevant neighbors are up weighted by w. This influences the final soft-max distribution obtained from KNN-retrieval for a given set of relevant neighbors $\mathcal{R}$ distances as $\forall d_j \in \mathcal{R}, d_j = d_j / w$ and $\forall d_i \notin \mathcal{R}, d_i = d_i * w$. Then retrieved set is used to form a softmax distribution over the vocabulary with temperature T as shown in Equation 2. We aggregate over multiple occurrences of the same word. The k nearest neighbors distribution is given as follows:

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

The model and kNN distributions are interpolated with a tuned parameter λ, resulting in the final $k\text{NN} - \text{MT}$ distribution:

$$p(y_i \mid x, \hat{y}_{1:i-1}) = \lambda p_{\text{kNN}}(y_i \mid x, \hat{y}_{1:i-1}) + (1 - \lambda) p_{\text{MT}}(y_i \mid x, \hat{y}_{1:i-1}) \tag{3}$$

The improvements in NMT results show that filtering the noisy neighbors with an adaptive K helps the NMT predictions. KNN-Kmeans MT performance compared to KNN-NMT for smaller databases is crucial as we tend to retrieve irrelevant neighbors when the data is very scarce. The number of retrieved neighbors is a hyperparameter. We use 2 clusters as part of the KMeans clustering, the relevant and the irrelevant clusters. A good illustration of our approach is shown in Figure1 and Figure2

4 Evaluation

4.1 Evaluation paradigm

The experiments are setup as an online paradigm to emulate real-world settings. During the inference process, there are three steps - Translation, Correction and Adaptation. For translation, we use the pre-trained model to generate the target translation $\hat{y}$ for a given source sentence x. The human annotator corrects the translation $\hat{y}$ to generate y. In adaptation phase, we encode x and y with the pre-trained model using teacher forcing. We update the data-stores used for K-Nearest Neighbor search using these representations. The corrected translation for the current sentence serves as a source of correction for similar sentences that follow.

4.2 Datasets and Metrics

For the experiments we use two datasets that are commonly used to evaluate machine translation models similar to previous works Wang et al. [2021], Khandelwal et al. [2020]. The first dataset we use is European Medicines Agency (EMEA) Tiedemann dataset which consists of documents related to medical products. The other dataset is JRC-Acquis dataset Ralf et al. which consists of European Union laws data related to European states. Both the datasets are tokenized using Moses toolkit and

Byte-Pair encoded. The datasets are divided into buckets based on the number of tokens and length of sentences which helps us evaluate the models in low-resource vs high-resource settings. There are five buckets based on the length of sentences - 0-50, 50-100, 100-200, 200- 500 and 500-1000. We consider buckets with less than 500 sentences as low resource. The statistics of the the datasets with respect to each bucket are described in Tables 1 and 2 All the models are evaluated on SacreBLEU metric with case-sensitive detokenized BLEU. The translations of the documents in a given bucket are concatenated and the BLEU scores are reported at corpus-level for each bucket in a given dataset.

Data Statistics for EMEA dataset					
Statistics	0-50	50-100	100-200	200-500	500-1000
Number of documents	22	14	7	4	5
Avg. number of Sentences	38.4	73	157.9	392.8	759.2
Avg. number of tokens	1174.7	1938.9	3466.1	9334.5	22725.6

Table 1: Statistics for EMEA dataset

Data Statistics for JRC Acquis dataset					
Statistics	0-50	50-100	100-200	200-500	500-1000
Number of documents	22	14	7	4	5
Avg. number of Sentences	38.1	73.1	158.5	373.8	734.8
Avg. number of tokens	1347.1	2477.7	5345.4	12518.2	26409.2

Table 2: Statistics for JRC Acquis dataset

4.3 Baselines

We compare our proposed methods with state-of-the-art neural machine translation methods.

1. Pre-Trained - Evaluating the pre-trained model without any adaptation quantifies the domain shift beween trained dataset and evaluation dataset. Comparing with this helps us understand the adaptation achieved by other models.
2. Online-Tuning - The pre-trained models are re-trained on human corrected translations to correct mistakes. These methods may not perform well in low-resource setting as language models require a lot of corrected data.
3. KNN-NMT Khandelwal et al. [2020] - The retrieval augmented method recently proposed based on K Nearest Neighbor search was shown to be very effective for Neural Machine Translation outperforming fine-tuning and other retrieval-augmented methods. Here the number retrieved neighbors K is 8 for both KNN-NMT and KNN-K means models.

KNN-Kmeans MT - EMEA dataset							
Model	λ	0-50	50-100	100-200	200-500	500-1000	Average
Pre-Trained	No lambda	43.8	43.1	38.3	41.9	40.8	41.6
Online Tuning	No lambda	**44**	43.5	39.6	43.8	**44.7**	43.1
KNN-NMT	Lambda fixed-0.3	43.8	44.3	39.9	**43.9**	44.1	43.2
KNN-KMeans-NMT	Lambda fixed-0.2	43.8	**44.5**	**40.1**	43.6	44.1	**43.22**

Table 3: BLEU score comparisons for EMEA dataset

KNN-Kmeans MT - JRC Acquis dataset							
Model	λ	0-50	50-100	100-200	200-500	500-1000	Average
Pre-Trained	No lambda	54	49.9	41.9	39.9	43.4	45.82
Online Tuning	No lambda	54.4	50.9	43.8	42.8	47.5	47.88
KNN-NMT	Lambda fixed - 0.3	56.3	52.4	45.8	**43.8**	**47.9**	**49.24**
KNN-K Means-NMT	Lambda fixed-0.2	**56.4**	**52.6**	**45.9**	43.5	47.1	49.1

Table 4: BLEU score comparisons for JRC Acquis dataset

5 Results

We compare the BLEU score Papineni et al. [2002] for KNN-Kmeans MT with other baselines desribed in the section 4.3 EMEA and JRC-Acquis datasets. As observed in the Tables 3 and 4, the KNN-Kmeans MT algorithm shows improved performance compared to KNN-NMT on both the datasets as we decrease the size of the database. For larger databases the chances of finding more relevant neighbors are higher and hence the vanilla KNN-NMT is expected to perform better due to the availability of more relevant neighbors. The clustering to down-weigh irrelevant neighbors improves the results in case of smaller databases. The analysis across varying size of databases provides insight about their performance in low-resource vs high-resource scenarios, specifically the trend we see in the columns with smaller database sizes in Table 3 and Table 4. Empirical results for this hypothesis serve of as proof of concepts that filtering unwanted neighbors based on error types improves NMT performance.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George Bm Van Den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. Improving language models by retrieving from trillions of tokens. In *International Conference on Machine Learning*, pages 2206–2240. PMLR, 2022.

Chenhui Chu and Rui Wang. A survey of domain adaptation for neural machine translation. In *Proceedings of the 27th International Conference on Computational Linguistics*, pages 1304–1319, Santa Fe, New Mexico, USA, August 2018. Association for Computational Linguistics. URL `https://aclanthology.org/C18-1111`.

Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Mingwei Chang. Retrieval augmented language model pre-training. In *International Conference on Machine Learning*, pages 3929–3938. PMLR, 2020.

Barry Haddow, Rachel Bawden, Antonio Valerio Miceli Barone, Jindřich Helcl, and Alexandra Birch. Survey of low-resource machine translation. *Computational Linguistics*, 48(3):673–732, 2022.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. Efficient nearest neighbor language models. *arXiv preprint arXiv:2109.04212*, 2021.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Nearest neighbor machine translation, 2020. URL `https://arxiv.org/abs/2010.00710`.

Yuxian Meng, Xiaoya Li, Xiayu Zheng, Fei Wu, Xiaofei Sun, Tianwei Zhang, and Jiwei Li. Fast nearest neighbor machine translation. *arXiv preprint arXiv:2105.14528*, 2021.

Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318, 2002.

Steinberger Ralf, Bruno Pouliquen, Anna Widiger, Camelia Ignat, Tomaž Erjavec, Dan Tufis, and Dániel Varga. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. URL https://joint-research-centre.ec.europa.eu/language-technology-resources/jrc-acquis_en#download-the-jrc-acquis-corpus.

J. Tiedemann. The jrc-acquis: A multilingual aligned parallel corpus with 20+ languages. URL https://opus.nlpl.eu/EMEA.php.

Dongqi Wang, Haoran Wei, Zhirui Zhang, Shujian Huang, Jun Xie, and Jiajun Chen. Non-parametric online learning from human feedback for neural machine translation, 2021. URL https://arxiv.org/abs/2109.11136.

Xin Zheng, Zhirui Zhang, Junliang Guo, Shujian Huang, Boxing Chen, Weihua Luo, and Jiajun Chen. Adaptive nearest neighbor machine translation, 2021. URL https://arxiv.org/abs/2105.13022.