Learning Decoupled Retrieval Representation for Nearest Neighbour Neural Machine Translation

Qiang Wang[1,2], **Rongxiang Weng**[3,4], **Ming Chen**[2*]
[1]Zhejiang University, Hangzhou, China
[2]RoyalFlush AI Research Insistute, Hangzhou, China
[3]School of Computer Science and Technology, Soochow University, Suzhou, China
[4]miHoYo AI, Shanghai, China
{wangqiangenu, wengrongxiang}@gmail.com, chenming@myhexin.com

Abstract

K-Nearest Neighbor Neural Machine Translation (*k*NN-MT) successfully incorporates external corpus by retrieving word-level representations at test time. Generally, *k*NN-MT borrows the off-the-shelf context representation in the translation task, e.g., the output of the last decoder layer, as the query vector of the retrieval task. In this work, we highlight that coupling the representations of these two tasks is sub-optimal for fine-grained retrieval. To alleviate it, we leverage supervised contrastive learning to learn the distinctive retrieval representation derived from the original context representation. We also propose a fast and effective approach to constructing hard negative samples. Experimental results on five domains show that our approach improves the retrieval accuracy and BLEU score compared to vanilla *k*NN-MT.

1 Introduction

Conventional neural machine translation (NMT) cannot dynamically incorporate external corpus at inference once finishing training (Bahdanau et al., 2015; Vaswani et al., 2017), resulting in bad performance when facing unseen domains, even if feeding millions or billions of sentence pairs for training (Koehn and Knowles, 2017). To address this problem, researchers developed retrieval-enhanced NMT (RENMT) to flexibly incorporate external translation knowledge. Early RENMTs leverage a search engine to find the similar bitext to improve the translation performance (Zhang et al., 2018; Cao and Xiong, 2018; Gu et al., 2018; Xia et al., 2019). However, the results of sentence-level retrieval with high similarity are generally sparse in practical applications, while noises in low similarity retrieval could lead to severe performance degradation (Cao and Xiong, 2018).

*k*NN-MT proposed by Khandelwal et al. (2021) effectively alleviates the sparse problem by introducing the word-level k-nearest neighbor mechanism. Instead of storing the discrete word sequence, *k*NN-MT uses a pre-trained NMT model to force decoding the external corpus and remembers the word-level continuous context representation, e.g., the output of the last decoder layer. During inference, *k*NN-MT assumes that the same target words have similar contextual representations and weights word selection through retrieving current context representation from the memorized datastore. However, we point out that it is sub-optimal to directly use the off-the-shelf context representation in the translation task because this vector is not specific to fine-grained retrieval.

In this work, we attempt to decouple the context representation by learning an independent retrieval representation. To this end, we leverage supervised contrastive learning with multiple positive and negative samples to learn a good retrieval representation (called CLKNN). We also propose a fast and effective method to construct hard negative samples. Experimental results on five domains show that our approach outperforms the vanilla *k*NN-MT in terms of BLEU and retrieval accuracy.

2 Background

Vanilla NMT Given a source sentence $\boldsymbol{x} = \{x_1, x_2, \ldots, x_{|\boldsymbol{x}|}\}$ and a target prefix $\boldsymbol{y}_{<t} = \{y_1, y_2, \ldots, y_{t-1}\}$, the vanilla NMT predicts the next target word y_t by:

$$p_c(y_t|\boldsymbol{x}, \boldsymbol{y}_{<t}) \propto \exp\Big(q(\boldsymbol{h}_t)\Big) \tag{1}$$

where $\boldsymbol{h}_t = f_\theta(\boldsymbol{x}, \boldsymbol{y}_{<t}) \in \mathcal{R}^d$ is the context vector at step t with respect to $\boldsymbol{x}$ and $\boldsymbol{y}_{<t}$; $f_\theta(\cdot)$ can be arbitrary encoder-decoder network with parameters θ, such as Transformer (Vaswani et al., 2017); $q(\cdot)$ linearly projects $\boldsymbol{h}_t$ to target vocabulary size.

kNN-MT *k*NN-MT hypothesizes that the same target words have similar representations. To

*Corresponding author.

Proceedings of the 29th International Conference on Computational Linguistics, pages 5142–5147
October 12–17, 2022.

dynamically incorporate external sentence pairs $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})\}_{i=1}^{|\mathcal{D}|}$, kNN-MT extends Eq. 1 by interpolating a retrieval-based probability p_r:

$$p_{knn} = (1 - \lambda) \times p_c + \lambda \times p_r \tag{2}$$

where λ is the interpolation coefficient as a hyper-parameter.

Specifically, kNN-MT first uses a pre-trained NMT model to force decoding each sentence pair $(\boldsymbol{x}^{(i)}, \boldsymbol{y}^{(i)})$ to build a key-value datastore $\mathcal{H}$:

$$\mathcal{H} = \bigcup_{i=1}^{|\mathcal{D}|} \bigcup_{t=1}^{|\boldsymbol{y}^{(i)}|} \left\{ (\boldsymbol{h}_t^{(i)}, y_t^{(i)}) \right\} \tag{3}$$

The key is the word-level context representation $\boldsymbol{h}_t^{(i)}$ and the value is the gold target word $y_t^{(i)}$. Then, given $\mathcal{H}$ and predicted target prefix $\hat{\boldsymbol{y}}_{<t}$ at test time, kNN-MT models $p_r(\hat{y}_t|\boldsymbol{x}, \hat{\boldsymbol{y}}_{<t})$ by measuring the distance between query $\hat{\boldsymbol{h}}_t = f_\theta(\boldsymbol{x}, \hat{\boldsymbol{y}}_{<t})$ and its k-nearest representations $\{(\tilde{\boldsymbol{h}}_i, \tilde{v}_i)\}_{i=1}^{k}$ in $\mathcal{H}$:

$$p_r(\hat{y}_t|\boldsymbol{x}, \hat{\boldsymbol{y}}_{<t}) \propto \sum_{i=1}^{k} \mathbb{1}_{\hat{y}_t = \tilde{v}_i} \exp\Big(\frac{-d(\tilde{\boldsymbol{h}}_i, \hat{\boldsymbol{h}}_t)}{T}\Big), \tag{4}$$

where $d(\cdot)$ is L_2 distance; T is temperature hyper-parameter; $\mathbb{1}$ is the indicator function.

3 Approach

Motivation According to Eq. 1-4, we can see that the context representation $\boldsymbol{h}$ simultaneously plays two roles in kNN-MT: (1) the semantic vector for p_c; (2) the retrieval vector for p_r. We note that coupling the same $\boldsymbol{h}$ in the two scenes is sub-optimal. Recall that $\boldsymbol{h}$ in the translation model is generally learned through cross-entropy loss, which only pays attention to the gold target token and ignores others.[1] However, a good retrieval vector should be able to distinguish between different tokens, especially those owning similar representations. Therefore, we attempt to derive a new retrieval vector $\boldsymbol{z}$ from $\boldsymbol{h}$ for better retrieval performance.

Retrieval representation adapter We use a simple feedforward network as an adapter to transform the original representation $\boldsymbol{h}$ to desired retrieval representation $\boldsymbol{z}$:

$$\boldsymbol{z} = \text{FFN}(\boldsymbol{h}) = \text{ReLU}(\boldsymbol{h}\boldsymbol{W}_1 + \boldsymbol{b}_1)\boldsymbol{W}_2 + \boldsymbol{b}_2, \tag{5}$$

where $\boldsymbol{W}_1 \in \mathcal{R}^{d \times d_f}$, $\boldsymbol{W}_2 \in \mathcal{R}^{d_f \times d_o}$, $\boldsymbol{b}_1 \in \mathcal{R}^{d_f}$, and $\boldsymbol{b}_2 \in \mathcal{R}^{d_o}$ are learnable parameters; d_f and d_o are the intermediate hidden size and output size of the adapter, respectively. When $d_o < d$, the adapter network can be regarded as a dimension reducer. As FFN is very lightweight compared to the calculation of $\boldsymbol{h}$, there is almost no latency in converting $\boldsymbol{h}$ to $\boldsymbol{z}$.For convenience, in the following description, we redefine $\boldsymbol{h}_i$ as the key of i-th key-value pair in the original datastore $\mathcal{H}$, and the corresponding value is denoted by Y_i when there is no ambiguity. In this way, the new datastore $\mathcal{Z}$ can be denoted as $\mathcal{Z} = \{(\boldsymbol{z}_i, Y_i)|i = 1, \ldots, |\mathcal{H}|\}$, where $\boldsymbol{z}_i = \text{FFN}(\boldsymbol{h}_i)$.

Supervised contrastive learning In machine translation field, contrastive learning has been applied in multilingual translation (Pan et al., 2021; Wei et al., 2021), cross-modal translation (Ye et al., 2022), and learning robust representation for low-frequency word (Zhang et al., 2021) etc. In this work, we use supervised contrastive learning (Grill et al., 2020) with multiple positive and negative samples to learn the desired retrieval representation $\boldsymbol{z}$. Here, we regard the unique token v in the target vocabulary V as a natural supervision signal. We aim to make $\boldsymbol{z}$ more distinguishable, for example, pulling $\boldsymbol{z}$ of the same words together and pushing $\boldsymbol{z}$ of different words apart. Specifically, we first divide $\mathcal{Z}$ into $|V|$ clusters according to the token class label. E.g., $C_v = \{\boldsymbol{z}_i|i = 1, \ldots, |\mathcal{Z}|, Y_i = v\}$, where C_v is the context representation cluster of token v. Thus, given any context representation $\boldsymbol{z} \in \mathcal{Z}$ and its token label v, we can construct M positive samples $\boldsymbol{z}^+ = \{\boldsymbol{z}_1^+, \ldots, \boldsymbol{z}_i^+, \ldots, \boldsymbol{z}_M^+\}$, where $\boldsymbol{z}_i^+$ is uniformly sampled from its owned cluster C_v and $\boldsymbol{z}_i^+ \neq \boldsymbol{z}$.[2] Likely, we further construct N negative samples $\boldsymbol{z}^- = \{\boldsymbol{z}_1^-, \ldots, \boldsymbol{z}_i^-, \ldots, \boldsymbol{z}_N^-\}$, where $\boldsymbol{z}_i^- \in \backslash C_v$, $\backslash C_v$ denotes other clusters except C_v. In the next part, we will describe how to build $\boldsymbol{z}^-$. Finally, given the anchor vector $\boldsymbol{z}$, its multiple positive samples $\boldsymbol{z}^+$ and multiple negative samples $\boldsymbol{z}^-$, we learn the adapter network through the following contrastive learning loss:

$$-\log \frac{\sum\limits_{1 \le i \le M} \exp(s(\boldsymbol{z}, \boldsymbol{z}_i^+))}{\sum\limits_{1 \le i \le M} \exp(s(\boldsymbol{z}, \boldsymbol{z}_i^+)) + \sum\limits_{1 \le j \le N} \exp(s(\boldsymbol{z}, \boldsymbol{z}_j^-))}, \tag{6}$$

where $s(\cdot)$ is the score function implemented as cosine similarity with temperature T': $s(\boldsymbol{a}, \boldsymbol{b}) =$

[1] In practice, we often use its label-smooth variant, which evenly assigns a small probability mass to all non-gold labels without distinction.

[2] We use sampling with replacement when $|C_v| < M$.

$\frac{1}{T'} \times \frac{\boldsymbol{a}^T \boldsymbol{b}}{\|\boldsymbol{a}\| \cdot \|\boldsymbol{b}\|}$. Note that T' is the temperature in training, which is different from the inference temperature T in Eq. 4.

Fast hard negative sample The key for Eq. 6 is the construction of negative samples $\boldsymbol{z}^-$. A trivial solution is randomly sampling from the entire space of $\backslash C_v$. However, this negative sample may be too easy to provide the effective learning signal (Robinson et al., 2020). On the contrary, an extreme method for hard negative samples is to traverse $\backslash C_v$ to find the most similar negative samples for the anchor. The problem is that $|\backslash C_v|$ is close to $|\mathcal{Z}|$, with a scale of millions or more, resulting in enormous computational complexity. To solve it, we propose a fast and cheap approach to constructing hard negative samples. Specifically, we first collect the cluster centre $\bar{C}_v = \frac{1}{|C_v|} \sum_{i=1}^{|\mathcal{Z}|} \mathbb{1}_{Y_i=v} \boldsymbol{z}_i$. We calculate the nearest K (K>=N) cluster centers w.r.t the anchor and randomly sample N clusters to make the source of the negative sample diverse. Then we randomly sample one point from the corresponding cluster as a negative sample. As the anchor vector only involves querying $|C|$ cluster centers and $|C| << |\mathcal{Z}|$, our approach runs faster than the exact global search.

Inference After training, we use the well-trained FFN to rebuild the retrieval datastore $\mathcal{H}$ into $\mathcal{Z}$. To further reduce calculation cost at test time, we introduce PCA to reduce the dimension of the retrieval vector. We also add normalization after PCA to guarantee the numerical stability of the input to the inner product. Another difference with Eq. 4 is that we use the inner product instead of the L2 distance as distance metrics. The reason is that using consistent distance metrics in training and inference improves performance in primitive experiments. Concretely, we modify the original kNN-MT in Eq. 4 as:

$$p_r(\hat{\boldsymbol{y}}_t|\boldsymbol{x}, \hat{\boldsymbol{y}}_{<t}) \propto \sum_{i=1}^{k} \mathbb{1}_{\hat{y}_t=\tilde{v}_i} \exp\Big(\frac{g(\tilde{\boldsymbol{z}}_i) \otimes g(\hat{\boldsymbol{z}}_t)}{T}\Big), \tag{7}$$

where $g(x) = \text{Norm}(\text{PCA}(x))$, $\otimes$ denotes inner product operation, $\tilde{\boldsymbol{z}}_i$ is the i-th nearest neighbor in $\mathcal{Z}$ for the current retrieval representation $\hat{\boldsymbol{z}}_t$. As a bonus, since the numeric range of the normalized inner product is $[0, 1]$, which can be seen as the confidence in retrieving.[3] We leverage this nature

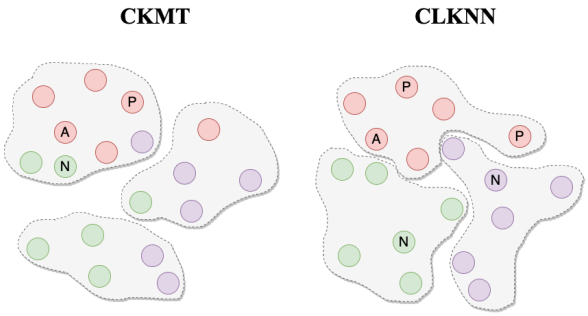


Figure 1: Illustration the differences between CKMT and CLKNN in constructing positive and negative samples. Different colors indicate different tokens. A/P/N means anchor, positive sample and negative sample, respectively.

to modify the interpolation coefficient λ in Eq. 2 to be aware of retrieval confidence:

$$\lambda^* = \lambda \times \frac{\sum_{i=1}^{k} g(\tilde{\boldsymbol{z}}_i) \otimes g(\hat{\boldsymbol{z}}_t)}{k}. \tag{8}$$

λ^* can be considered a simple adaptive coefficiency like Zheng et al. (2021); Jiang et al. (2021); Wang et al. (2022), but does not require training.

Discussion The closest work with us is CKMT (Wang et al., 2022). As illustrated in Figure 1, there are two major differences compared with CKMT: (1) CLKNN uses multiple positive and negative samples, while CKMT only considers a single positive and negative sample, limiting the exploration of representation space. (2) CKMT requires to partition clusters through cost-expensive clustering in full-scale datastore, while CLKNN predefines clusters based on vocabulary labels and only involves calculating cluster centers. In practice, we spent about 6 hours on the CPU to complete the cluster operation in CKMT, while CLKNN only takes about 3 minutes.

4 Experiments

Setup To fairly compared with previous work (Khandelwal et al., 2021), we use WMT'19 German-English news translation task winner (Ng et al., 2019) as our strong general domain baseline. We use the same German-English multi-domain datasets, consisting of five domains, including `Medical`, `Law`, `IT`, `Koran` and `Subtitles` [4]. Besides, to test the proposed training approach robust in out-domain scenery, we also use a 2M subset of the baseline's training data, including *News*

[3] L2 distance lacks this feature because its numeric range is too broad, e.g., 0~1000 in our observation.

[4] We use the provided 500K sentence pairs version subtitle data rather than full size 12.4M due to memory limitation.

Dataset	Medical	Law	IT	Koran	Subtitle	NC+Euro
Train	248K	467K	222K	52K	500K	2M
Valid	2000	2000	2000	2000	2000	-
Test	2000	2000	2000	2000	2000	-
Datastore	6.9M	19.0M	3.6M	0.5M	6.2M	5M†

Table 1: Statistics of datasets in different domains. †: Due to limited memory, we randomly sampled 5M samples from a total of 65.7M samples in NC+Euro for training.

Method	Medical	Law	IT	Koran	Subtitle	Avg.
Baseline (WMT19 winner, Ng et al. (2019))	39.91	45.71	37.98	16.3	29.21	33.82
*k*NN-MT (Khandelwal et al., 2021)	54.35	61.78	45.82	19.45	**31.73**†	42.63
*k*NN-MT (our implementation)	54.41	61.01	45.20	21.07	29.67	42.27
train by out-domain data						
CLKNN	56.37	61.54	46.50	21.52	30.81	43.35
CLKNN + λ^*	**56.52**	61.63	46.68	21.60	30.86	43.46
train by in-domain data						
CLKNN	55.86	61.92	47.77	21.46	31.02	43.61
CLKNN + λ^*	55.87	**62.01**	**47.84**	**21.81**	31.05	**43.72**

Table 2: The SacreBLEU scores of our proposed CLKNN and the baseline methods in five domains. λ^* denotes using retrieval confidence aware interpolation coefficiency. † denotes the number is not comparable because Khandelwal et al. (2021) use full-size subtitle data than ours. All the CLKNN results are significantly better ($p<0.01$) than our re-implemented *k*NN-MT, measured by paired bootstrap resampling (Koehn, 2004).

Commentary v14 and *Europarl v9*, and randomly sample 5M samples out of 65.7M samples from its datastore. See Table 1 for detailed data statistics.

Implementation details All experiments run on a single NVIDIA 2080 Ti GPU. We use *Faiss* [5] for vector retrieval. For CLKNN, the number of positive samples is M=2, and the number of negative samples is N=32. We sample N negative samples from K=128 nearest clusters. The training batch size is 32. During training, we set $T^{'}$=0.01, while we vary T according to the validation set at test time. The hidden state size d_f and output size d_o of adapter is 4096 and 512, respectively. The output dimension of PCA is 128. We train all models for 500k steps and select the best model on the validation set. We use a beam size of 5 and a length penalty of 1.0 for all experiments for inference. We measure case-sensitive detokenized BLEU by SacreBLEU.

Experimental results Table 2 reports the SacreBLEU scores in five domains. We can see that: (1) CLKNN is robust about training data: using out-domain or in-domain average improves 1+ points than our *k*NN-MT; (2) The gap between in-domain and out-domain is small (about 0.3 points), meaning that our approach does not rely on in-domain data and is more practical than Zheng et al. (2021); Jiang et al. (2021); (3) using proposed λ^* slightly improve the performance across the board. These results show that learning independent retrieval representation is helpful for vanilla *k*NN-MT. Besides, we also compare the inference speed between CLKNN and *k*NN-MT through running five times on IT test set. The results show that CLKNN has a comparable speed (97%±2%) to that of *k*NN-MT because the adapter in CLKNN is very lightweight.

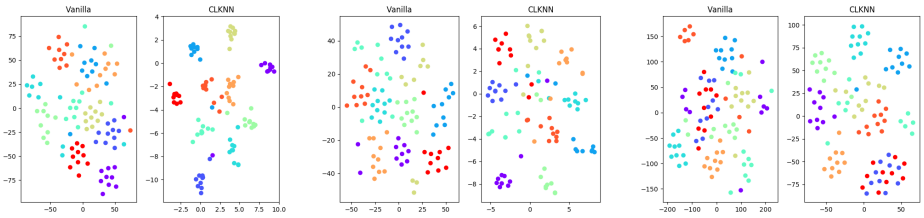
M	N	BLEU	M	N	BLEU
1	1	45.54	2	16	46.37
1	16	45.91	2	32	46.68
1	32	46.13	2	64	46.55
1	64	45.88	4	32	46.29

Table 3: The BLEU scores on IT test set against the number of the positive (M) and negative (N) samples.

5 Analysis

Effect of the number of contrastive samples One of the main differences between Wang et al. (2022) and us is that we use multiple positive and negative samples in our training objective. We vary the number of M and N and report the BLEU scores in Table 3. As we can see, increasing M and N is helpful for our method. However, large M can-

[5] https://github.com/facebookresearch/faiss



(a) High (b) Middle (c) Low

Figure 2: Visualization of retrieval vector on different frequency words by t-SNE. We uniformly sample 10 classes in each category, and each class contains ten random representations. The same color denotes the same class.

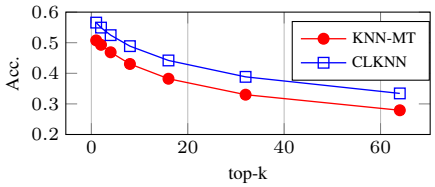


Figure 3: Retrieval accuracy curve against top-k.

not befit more than increasing N. We attribute it to positive samples that are too easy to learn because most are close in embedding space. On the contrary, negative samples from different clusters can provide a stronger learning signal. To further validate the effectiveness of multiple samples, we also conduct experiments on `Medical`. The results are similar to that of `IT`: using M=2, N=32 is 1.64 BLEU points higher than using M=1, N=1 (56.52 vs. 54.88). It indicates that using multiple positive and negative samples is necessary to achieve good performance for contrastive learning.

Retrieval accuracy Intuitively, our approach can learn more accurate retrieval representation than vanilla *k*NN-MT. To validate this hypothesis, we use `IT` validation as the datastore and plot the retrieval accuracy on top-k in Figure 3. We can see that CLKNN has more robust retrieval accuracy than *k*NN-MT no matter how k changes. It indicates that the performance improvement comes from our better retrieval representation.

Visualization We visually present the differences between baseline and CLKNN on embedding space. Specifically, we split three categories according to the word frequency in `IT` training set: `HIGH`(the first 1%), `Middle`(40%-60%) and `LOW`(the last 1%) [6]. We uniformly sample 10 unique words in each category and randomly sample 10 unique vector representations from the training datastore. We use t-SNE to plot these representations, as illustrated in Figure 2. We can see that: (1) high-frequency words' representations are prone to distinguish for both baseline and CLKNN; (2) CLKNN has more close distances in the same vocabulary than baseline; (3) CLKNN has more robust accuracy for low-frequency words.

[6]We filter words whose frequency is less than 10.

6 Conclusion

In this work, we proposed to use supervised contrastive learning to decouple the context representation from vanilla *k*NN-MT. Experimental results on several tasks show that our approach outperforms *k*NN-MT and learns a more accurate retrieval representation.

Acknowledgements

We would like to thank the anonymous reviewers for the helpful comments. We also thank Shuqin Pan for the writing suggestions.

References

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In *3rd International Conference on Learning Representations, ICLR 2015*.

Qian Cao and Deyi Xiong. 2018. Encoding gated translation memory into neural machine translation. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 3042–3047, Brussels, Belgium. Association for Computational Linguistics.

Jean-Bastien Grill, Florian Strub, Florent Altché, Corentin Tallec, Pierre Richemond, Elena Buchatskaya, Carl Doersch, Bernardo Avila Pires, Zhaohan Guo, Mohammad Gheshlaghi Azar, Bilal Piot, koray kavukcuoglu, Remi Munos, and Michal Valko. 2020. Bootstrap your own latent - a new

approach to self-supervised learning. In *Advances in Neural Information Processing Systems*, volume 33, pages 21271–21284. Curran Associates, Inc.

Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor O. K. Li. 2018. Search engine guided neural machine translation. In *Proceedings of the Thirty-Second AAAI Conference on Artificial Intelligence, (AAAI-18), the 30th innovative Applications of Artificial Intelligence (IAAI-18), and the 8th AAAI Symposium on Educational Advances in Artificial Intelligence (EAAI-18), New Orleans, Louisiana, USA, February 2-7, 2018*, pages 5133–5140. AAAI Press.

Qingnan Jiang, Mingxuan Wang, Jun Cao, Shanbo Cheng, Shujian Huang, and Lei Li. 2021. Learning kernel-smoothed machine translation with retrieved examples. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7280–7290, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2021. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Philipp Koehn. 2004. Statistical significance tests for machine translation evaluation. In *Proceedings of the 2004 Conference on Empirical Methods in Natural Language Processing*, pages 388–395, Barcelona, Spain. Association for Computational Linguistics.

Philipp Koehn and Rebecca Knowles. 2017. Six challenges for neural machine translation. In *Proceedings of the First Workshop on Neural Machine Translation*, pages 28–39.

Nathan Ng, Kyra Yee, Alexei Baevski, Myle Ott, Michael Auli, and Sergey Edunov. 2019. Facebook FAIR's WMT19 news translation task submission. In *Proceedings of the Fourth Conference on Machine Translation (Volume 2: Shared Task Papers, Day 1)*, pages 314–319, Florence, Italy. Association for Computational Linguistics.

Xiao Pan, Mingxuan Wang, Liwei Wu, and Lei Li. 2021. Contrastive learning for many-to-many multilingual neural 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 1: Long Papers)*, pages 244–258, Online. Association for Computational Linguistics.

Joshua David Robinson, Ching-Yao Chuang, Suvrit Sra, and Stefanie Jegelka. 2020. Contrastive learning with hard negative samples. In *International Conference on Learning Representations*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, pages 6000–6010.

Dexin Wang, Kai Fan, Boxing Chen, and Deyi Xiong. 2022. Efficient cluster-based k-nearest-neighbor machine translation. *CoRR*, abs/2204.06175.

Xiangpeng Wei, Rongxiang Weng, Yue Hu, Luxi Xing, Heng Yu, and Weihua Luo. 2021. On learning universal representations across languages. In *International Conference on Learning Representations*.

Mengzhou Xia, Guoping Huang, Lemao Liu, and Shuming Shi. 2019. Graph based translation memory for neural machine translation. In *The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019*, pages 7297–7304. AAAI Press.

Rong Ye, Mingxuan Wang, and Lei Li. 2022. Cross-modal contrastive learning for speech translation. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 5099–5113, Seattle, United States. Association for Computational Linguistics.

Jingyi Zhang, Masao Utiyama, Eiichro Sumita, Graham Neubig, and Satoshi Nakamura. 2018. 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, Volume 1 (Long Papers)*, pages 1325–1335, New Orleans, Louisiana. Association for Computational Linguistics.

Tong Zhang, Wei Ye, Baosong Yang, Long Zhang, Xingzhang Ren, Dayiheng Liu, Jinan Sun, Shikun Zhang, Haibo Zhang, and Wen Zhao. 2021. Frequency-aware contrastive learning for neural machine translation. *CoRR*, abs/2112.14484.

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)*, pages 368–374, Online. Association for Computational Linguistics.