Efficient Nearest Neighbor Emotion Classification with BERT-whitening

Wenbiao Yin and **Lin Shang***
State Key Laboratory for Novel Software Technology
Department of Computer Science and Technology
Nanjing University, Nanjing, China
wenbiaoyin@smail.nju.edu.cn, shanglin@nju.edu.cn

Abstract

Retrieval-based methods have been proven effective in many NLP tasks. Previous methods use representations from the pre-trained model for similarity search directly. However, the sentence representations from the pre-trained model like BERT perform poorly in retrieving semantically similar sentences, resulting in poor performance of the retrieval-based methods. In this paper, we propose KNN-EC, a simple and efficient non-parametric emotion classification (EC) method using nearest neighbor retrieval. We use BERT-whitening to get better sentence semantics, ensuring that nearest neighbor retrieval works. Meanwhile, BERT-whitening can also reduce memory storage of datastore and accelerate retrieval speed, solving the efficiency problem of the previous methods. KNN-EC average improves the pre-trained model by 1.17 F1-macro on two emotion classification datasets.

1 Introduction

Retrieval-based methods enhance model performance by extracting textual information related to the input from large training data. The model interpolates the original model distribution and retrieved results' distribution as the final distribution for output. Retrieval-based methods can often improve the performance as the model is exposed to related knowledge not present in the input(Wang et al., 2022). Retrieval-based methods have been successfully applied to many tasks such as language modeling(Khandelwal et al., 2019; Guu et al., 2020), machine translation(Gu et al., 2018; Bapna and Firat, 2019; Khandelwal et al., 2020; Zheng et al., 2021), question answering(Chen et al., 2017; Lee et al., 2019), and computer vision(Devlin et al., 2015; Gur et al., 2021).

Retrieval-based methods are expressive, adaptable, and interpretable. Here, we propose KNN-EC, a simple and efficient non-parametric emotion classification (EC) method using nearest neighbor retrieval. KNN-EC can be added to any pre-trained emotion classification model without further training. Pre-trained models such as BERT (Devlin et al., 2019), RoBERTa (Liu et al., 2019), XLNet (Yang et al., 2019) have been widely proven effective for emotion classification, and they obtain state-of-the-art results on many emotion classification tasks.

However, the sentence representations from the pre-trained model like BERT perform poorly in retrieving semantically similar sentences(Reimers and Gurevych, 2019). Previous works found that pre-trained models always induced a non-smooth anisotropic semantic space of sentences, which harmed its performance of semantic similarity. To address this issue, Gao et al. (2018) designed a novel way to mitigate the degeneration problem by regularizing the word embedding matrix. Li et al. (2020) proposed to transform the BERT sentence embedding distribution into a smooth and isotropic Gaussian distribution through normalizing flow, an invertible function parameterized by the neural network. Because the above methods require additional parameters added to the neural network, this is not suitable to be added to KNN-EC to enhance the nearest neighbor retrieval.

Here, we use BERT-whitening(Su et al., 2021), which utilizes a simple linear transformation to enhance the isotropy of sentence representations and achieve competitive results. The details of BERT-whitening are described in Section 2.2. We conduct experiments on two emotion classification datasets. KNN-EC can achieve 1.21-1.65/0.28-1.44 F1-macro scores improvements over pre-trained models on GoEmotions/ISEAR. Meanwhile, by adding the BERT-whitening, our approach can further boost the model performance, optimize memory storage and accelerate retrieval speed.

The main contributions of this paper are summa-

*Corresponding author

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 4738–4745
December 7-11, 2022

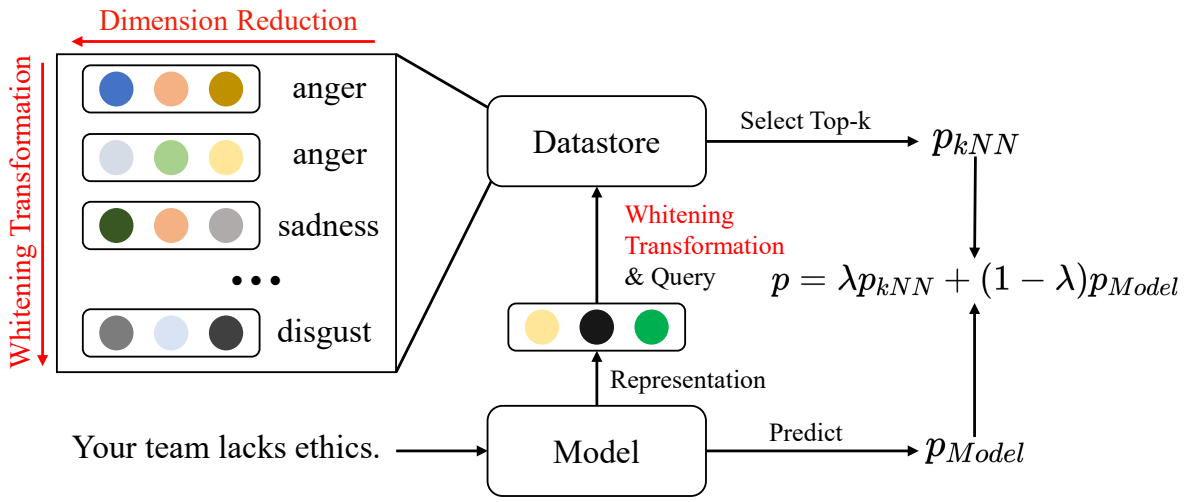

Figure 1: An overview of the proposed KNN-EC. The datastore stores the hidden representations of each sentence in the training data as keys and their corresponding labels as values. We use whitening transformation to enhance the isotropy of sentence representations and dimension reduction to optimize memory storage and accelerate retrieval speed. In inference, we use the whitening transformation on the test sentence's representation to retrieve the k nearest neighbors from the datastore. We interpolate the model and kNN distributions with a hyper-parameter λ as the final distribution.

rized as follows:

- We propose KNN-EC, a non-parametric emotion classification method using nearest neighbor retrieval. KNN-EC is plug-and-play for any pre-trained emotion classification model without further training.
- We utilize BERT-whitening to get better sentence semantics, ensuring the nearest neighbor retrieval works. Meanwhile, BERT-whitening can also optimize memory storage and accelerate retrieval speed.
- The results demonstrate that KNN-EC average improves the pre-trained model by 1.17 F1-macro scores on two emotion classification datasets.

2 Background

In this section, we will briefly introduce the background of KNN-MT(Khandelwal et al., 2020) and BERT-whitening(Su et al., 2021).

2.1 KNN-MT

KNN-MT includes two steps: creating a datastore and inferring based on the datastore.

2.1.1 Datastore Creation

The datastore in KNN-MT stores the hidden representations of translation contexts as keys and their corresponding next tokens as values. Given a bilingual sentence pair in the training set $(x, y) \in (\mathcal{X}, \mathcal{Y})$, an MT model translates the t-th target token y_t based on the translation context $(x, y_{<t})$. The key is $f(x, y_{<t})$, where f represents a mapping from input to an intermediate representation of the decoder and the value is y_t. The representations are generated by a single forward pass over the training set $(\mathcal{X}, \mathcal{Y})$ and the datastore is defined as follows:

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

2.1.2 Inference

At test time, the KNN-MT model aims to predict $\hat{y}_t$ given the already generated tokens $\hat{y}_{<t}$ as well as the context representation $f(x, \hat{y}_{<t})$, which is used to query the datastore for the k nearest neighbors according to L2 distance. Denoting the retrieved k nearest neighbors as $N^t = \{(h_i, v_i), i \in \{1, 2, \ldots, k\}\}$, their distribu-

tion over the vocabulary is computed as:

$$p_{\text{kNN}}(y_t \mid x, \hat{y}_{<t}) \propto \sum_{(h_i, v_i) \in N^t} \mathbb{1}_{y_t = v_i} \exp\left(\frac{-d(h_i, f(x, \hat{y}_{<t}))}{T}\right) \quad (2)$$

where $d(.,.)$ is an L2 distance function between the two vectors, and T is the temperature. KNN-MT interpolates the MT model and kNN distributions with a hyper-parameter λ:

$$p\left(y_t \mid x, \hat{y}_{<t}\right) = \lambda p_{\text{kNN}}\left(y_t \mid x, \hat{y}_{<t}\right) + (1-\lambda) p_{\text{MT}}\left(y_t \mid x, \hat{y}_{<t}\right) \quad (3)$$

2.2 BERT-whitening

BERT-whitening uses the whitening operation in traditional machine learning to enhance the isotropy of sentence representations and reduce the dimension of the sentence representation. BERT-whitening includes two steps: whitening transformation and dimension reduction.

2.2.1 Whitening Transformation

BERT-whitening transforms the mean value of the sentence vectors into 0 and the covariance matrix into the identity matrix. Given a set of row vectors $\{e_i\}_{i=1}^N$, BERT-whitening transforms them into $\{\tilde{\boldsymbol{e}}_i\}_{i=1}^N$ as:

$$\tilde{\boldsymbol{e}}_i = (\boldsymbol{e}_i - \boldsymbol{\mu})\, \boldsymbol{W} \quad (4)$$

where the mean value $\mu = \frac{1}{N}\sum_{i=1}^N e_i$, and W is the linear transformation that is solved later. The original covariance matrix $\sum$ is denoted as:

$$\Sigma = \frac{1}{N}\sum_{i=1}^{N} (e_i - \mu)^T (e_i - \mu) \quad (5)$$

The transformed covariance matrix $\widetilde{\Sigma} = W^T \Sigma W$ and $\widetilde{\Sigma} = I$, therefore,

$$\boldsymbol{\Sigma} = \left(\boldsymbol{W}^\top\right)^{-1} \boldsymbol{W}^{-1} = \left(\boldsymbol{W}^{-1}\right)^\top \boldsymbol{W}^{-1} \quad (6)$$

Σ satisfies the following form of SVD(Golub and Reinsch, 1971) decomposition:

$$\Sigma = U \Lambda U^T \quad (7)$$

where U is an orthogonal matrix, Λ is a diagonal matrix and the diagonal elements are all positive. Therefore, let $W^{-1} = \sqrt{\Lambda} U^T$, we can obtain the solution:

$$W = U\sqrt{\Lambda^{-1}} \quad (8)$$

2.2.2 Dimension Reduction

The elements in the diagonal matrix Λ deriving from SVD have been sorted in descending order. We only need to retain the first n columns of W to achieve a dimension reduction effect, which is equivalent to PCA(Abdi and Williams, 2010). Here, n is an empirical hyper-parameter, the details of algorithm implementation of BERT-whitening are shown in Algorithm 1.

Algorithm 1 Whitening-n Workflow

Input: The original embeddings $\{e_i\}_{i=1}^N$ and reserved dimension n

Output: Transformed embeddings $\{\widetilde{e}_i\}_{i=1}^N$

compute μ and $\sum$ of $\{e_i\}_{i=1}^N$
compute $U, \Lambda, U^T = \text{SVD}(\Sigma)$
compute $W = \left(U\sqrt{\Lambda^{-1}}\right)[:, :n]$
for $i = 1, 2, ..., N$ **do**
    $\widetilde{e}_i = (e_i - \mu)\, W$

3 Nearest Neighbor Emotion Detection

In this section, we will introduce the details of our proposed method, KNN-EC. We build the datastore as KNN-MT. Before that, we do whitening transformation and dimension reduction on the high-dimensional vector representations obtained from the model to get a better similarity search. In inference, the KNN-EC uses the nearest neighbor retrieval mechanism to enhance the pre-trained model without additional training. The overview of the proposed KNN-EC is shown in Figure 1.

3.1 Datastore Creation

Given a training set $(x, y) \in (\mathcal{X}, \mathcal{Y})$, a neural emotion classification model transforms the input sentence x into a high-dimensional vector representation $e = f(x)$, where f represents a mapping from input to an intermediate representation of the neural emotion classification model.

Then, we get the high-dimensional vector representation $\{e_i\}_{i=1}^N$ of all training inputs $\{x_i\}_{i=1}^N$. We use BERT-whitening to enhance the isotropy of sentence representations and reduce the dimension of the sentence representation, the details of algorithm implementation of BERT-whitening are shown in Algorithm 1.

$$\{\widetilde{e}_i\}_{i=1}^N = \textbf{Whitening-n}(\{e_i\}_{i=1}^N) \quad (9)$$

GoEmotions	Anger	Disgust	Fear	Joy	Sadness	Surprise	Total
Train	3878	498	515	12920	2121	3553	23485
Test	520	76	77	1603	259	449	2984

Table 1: The statistics of GoEmotions.

ISEAR	Anger	Disgust	Fear	Joy	Sadness	Shame	Guilt	Total
Train	863	853	861	873	865	857	840	6012
Test	216	213	215	219	217	214	210	1504

Table 2: The statistics of ISEAR.

where n is an empirical hyper-parameter to reduce the sentence representation's dimension and enhance the search results in the datastore.

Our datastore is constructed offline and consists of a set of key-value pairs. The key is $\widetilde{e}_i$, and the value is the corresponding ground truth y_i. The datastore is defined as follows:

$$(\mathcal{K}, \mathcal{V}) = \bigcup_{(x,y)\in(\mathcal{X},\mathcal{Y})} \{(\widetilde{e}_i, y_i)\} \tag{10}$$

3.2 Inference

At test time, given a sentence x, the pre-trained emotion classification model outputs a emotion distribution $p_{Model}(y|x)$ for target y. The model also outputs the vector representation e of x. We do whitening transformation for e by μ and W calculated in Equation 9 as follows:

$$\widetilde{e} = (e - \mu)W \tag{11}$$

$\widetilde{e}$ is used to query the datastore for the k nearest neighbors $\mathcal{M}$ according to L2 distance, $d(.,.)$, which is the same as Equation 2.

$$p_{\text{kNN}}(y \mid x) \propto \sum_{(k_i,v_i)\in\mathcal{M}} \mathbb{1}_{y=v_i} \exp\left(\frac{-d(k_i, \widetilde{e})}{T}\right) \tag{12}$$

where T is the temperature. We interpolate the pre-trained emotion classification model and kNN distributions with a hyper-parameter λ:

$$\begin{aligned} p(y \mid x) = &\lambda p_{\text{kNN}}(y \mid x) \\ &+ (1 - \lambda) p_{\text{Model}}(y \mid x) \end{aligned} \tag{13}$$

4 Experiment

4.1 Datasets and Evaluation Metrics

We evaluate our model[1] on two datasets: GoEmotions and ISEAR.

GoEmotions(Demszky et al., 2020) was published by Google for fine-grained emotions classification. GoEmotions is the largest manually annotated dataset of 58k English Reddit comments, labeled for 27 emotion categories or Neutral. Here, we filter the sentences with Neutral labels or multiple labels. We adopt Ekman-style(Ekman, 1992) grouping into six coarse categories(joy, anger, fear, sadness, disgust, and surprise). The statistics of GoEmotions are shown in Table 1.

ISEAR(Scherer and Wallbott, 1994)(International Survey on Emotion Antecedents and Reactions) collected personal reports on emotional events written by 3000 people from different cultural backgrounds. ISEAR contains 7k sentences that are labeled into seven categories(anger, disgust, fear, guilt, joy, sadness, and shame). The statistics of ISEAR are shown in Table 2.

Here, we use F1-macro(Opitz and Burst, 2019) to measure all results.

4.2 Implementation Details

Considering the generalization, we conduct experiments on the popular pre-trained models, such as BERT, RoBERTa, and XLNet. We fine-tune the pre-trained models from Transformers library[2] and adopt faiss[3] to implement our model. For our method, the temperature T is set to 2. We set the batch size of this model as 64 and use the AdamW optimizer and the learning rate warm-up where the learning rate is 2e-5.

4.3 Overall Results

Experimental results are shown in Table 3. The results in the table are the average of 5 repeated experiments. According to the results, several observations can be noted.

[1] https://github.com/WenbiaoYin/KNN-EC

[2] https://github.com/huggingface/transformers

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

Dataset	GoEmotions			ISEAR		
Model	p_{Model}	p_{kNN}	p	p_{Model}	p_{kNN}	p
bert-base	71.13	71.55	72.71$\uparrow_{1.58}$	69.18	69.28	69.46$\uparrow_{0.28}$
bert-large	71.64	72.23	72.85$\uparrow_{1.21}$	70.20	70.47	70.94$\uparrow_{0.74}$
xlnet-base	71.31	71.81	72.77$\uparrow_{1.46}$	68.39	69.66	69.83$\uparrow_{1.44}$
roberta-base	71.86	72.39	73.51$\uparrow_{1.65}$	69.36	70.40	70.68$\uparrow_{1.32}$
roberta-large	73.01	74.10	**74.51**$\uparrow_{1.50}$	73.09	73.12	**73.56**$\uparrow_{0.47}$
μ	71.79	72.42	**73.27**$\uparrow_{1.48}$	70.04	70.59	**70.89**$\uparrow_{0.85}$

Table 3: The F1-macro scores of the pre-trained model(p_{Model}) and k nearest neighbor(p_{kNN}) and the proposed KNN-EC model(p). Underline results indicate the best results of baselines, and our best results are marked bold. The red numbers indicate the magnitude of our model effect improvement. μ indicates the mean value of results among different pre-trained models.

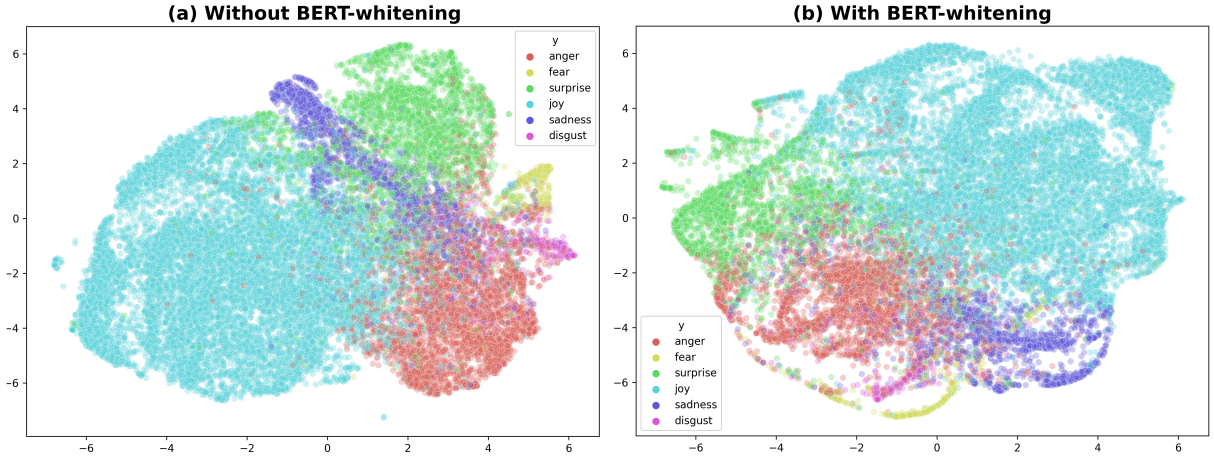

Figure 2: Visualization of sentence representations on GoEmotions. (a) uses sentence representations from the fine-tuned RoBERTa without BERT-whitening, while (b) uses RoBERTa with BERT-whitening.

$\mathbf{p_{kNN}}$ with BERT-whitening is always better than $\mathbf{p_{Model}}$. It is a fascinating phenomenon: why non-parametric method using nearest neighbor retrieval is better than the well-trained linear classifier? Because BERT always induces a non-smooth anisotropic semantic space of sentences, BERT-whitening can boost the isotropy of sentence distribution. It makes p_{kNN} with BERT-whitening is always better than p_{Model}.

KNN-EC average improves the pre-trained models by 1.17 F1-macro on two emotion classification datasets. KNN-EC can achieve 1.21-1.65/0.28-1.44 F1-macro improvements over pre-trained models on GoEmotions/ISEAR.

The more data available for retrieval, the better our model works. The effectiveness of KNN-EC depends on the quality of retrieved nearest neighbors. The quality of retrieved nearest neighbors depends on the amount of data available for retrieval. p_{kNN} average achieves 0.63/0.55 F1-macro improvements over p_{Model}. The KNN-EC's improvement is more pronounced on larger datastore.

4.4 The Effectiveness of BERT-whitening

From Table 4, we can see p_{kNN} with BERT-whitening average improves the 11.77/12.11 F1-macro over p_{kNN} without BERT-whitening on GoEmotions/ISEAR. The sentence representations from the pre-trained model perform poorly in retrieving semantically similar sentences while perform well with BERT-whitening. Besides, p_{kNN} without BERT-whitening is very unstable, whose results are much lower than p_{Model} and the standard deviation(μ) equals 7.56/9.81 on GoEmotions/ISEAR. While p_{kNN} with BERT-whitening is not only stable($\mu = 0.89/1.35$) but also better than p_{Model}.

To demonstrate the effectiveness of our method, we further visualize the sentence representations from RoBERTa on GoEmotions, where RoBERTa achieves the best performance. From Figure 2, sentence representations in the sadness category almost overlap with sentence representations in

Dataset	GoEmotions		ISEAR	
Model	p_{kNN}-BW	p_{kNN}+BW	p_{kNN}-BW	p_{kNN}+BW
bert-base	66.79	71.55↑4.76	58.03	69.28↑11.25
bert-large	50.64	72.23↑21.59	40.24	70.47↑30.23
xlnet-base	67.72	71.81↑4.09	66.65	69.66↑3.01
roberta-base	52.24	72.39↑20.15	60.12	70.40↑10.28
roberta-large	65.85	74.10↑8.25	67.34	73.12↑5.78
μ	60.65	72.42↑11.77	58.48	70.59↑12.11
σ	7.56	0.89	9.81	1.35

Table 4: The F1-macro scores of the k nearest neighbors(p_{kNN}) without/with BERT-whitening on GoEmotions and ISEAR.We use the following abbreviations. BW for BERT-whitening, - for without and + for with. μ/σ indicates the mean value/standard deviation of results among different pre-trained models.

Whitening-n	8	16	32	64	128	256	default
bert-base	70.64	71.62	71.73	72.06	72.11	**72.71**	70.52
xlnet-base	71.35	72.20	**72.77**	71.28	71.90	72.09	70.43
roberta-base	72.27	72.05	72.53	73.05	**73.51**	72.66	59.22
Datastore size	0.01×	0.02×	0.04×	0.08×	0.17×	0.33×	1×
bert-large	**72.85**	71.16	72.06	71.96	71.74	71.85	71.73
roberta-large	73.01	72.64	74.09	74.11	**74.51**	74.06	73.32
Datastore size	0.01×	0.02×	0.03×	0.06×	0.13×	0.25×	1×

Table 5: The F1-macro scores of KNN-EC among different **Whitening-n**s. Underline results indicate the worst results of KNN-EC for each pre-trained model, and the best results are marked bold. The default means using the same dimension as the pre-trained model's outputs without dimension reduction.

the surprise category, seriously affecting the nearest neighbor retrieval result. After adding BERT-whitening, the sentence representations of sadness and surprise are separated.

Retrieval-based methods generally have significant storage overhead and inference overhead. From Table 5, BERT-whitening not only improves the model's performance but also significantly reduces the memory storage of the datastore(more than 10×). Meanwhile, with the datastore size reduced, the retrieval speed is also greatly accelerated, significantly decreasing the inference overhead and erasing the limitation of non-parametric KNN-EC in practical applications.

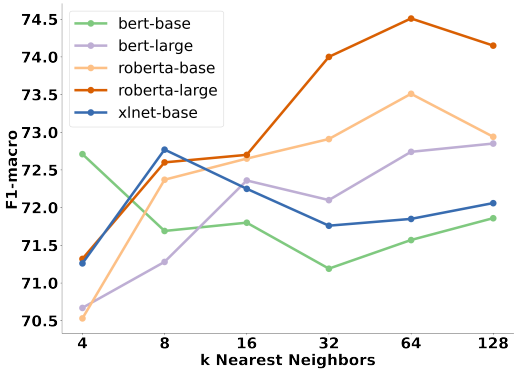

Figure 3: The F1-macro scores of KNN-EC among different ks.

4.5 Performance with Different k

As shown in Figure 3, as p_{Model} performs better, KNN-EC tends to select larger k, and retrieve more relevant sentences to improve performance, such as bert-large, roberta-base, roberta-large. While p_{Model} performs relatively poorly, selecting a larger k will introduce more noise, so it tends to select a small k, such as bert-base, xlnet-base.

5 Conclusion

In this paper, we propose a simple and effective method using nearest neighbor retrieval that can be applied to many emotion classification models without further training. Meanwhile, we reduce memory storage of datastore and accelerate retrieval speed, significantly decreasing the inference overhead and erasing the limitation of non-

parametric KNN-EC in practical applications. Experimental results show that our proposed model achieves significant improvement.

Limitations

There are two major limitations in this study that could be addressed in future research. First, the study focused on using the sentence representations from the pre-trained model for nearest neighbor retrieval. We did not study whether KNN-EC works on the other models. Second, the results retrieved from the training set may suffer from dataset bias.

Acknowledgements

This work is supported by the National Natural Science Foundation of China (No.51975294).

References

Hervé Abdi and Lynne J Williams. 2010. Principal component analysis. *Wiley interdisciplinary reviews: computational statistics*, 2(4):433–459.

Ankur Bapna and Orhan Firat. 2019. Non-parametric adaptation for neural machine translation. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 1921–1931.

Danqi Chen, Adam Fisch, Jason Weston, and Antoine Bordes. 2017. Reading wikipedia to answer open-domain questions. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1870–1879.

Dorottya Demszky, Dana Movshovitz-Attias, Jeongwoo Ko, Alan Cowen, Gaurav Nemade, and Sujith Ravi. 2020. Goemotions: A dataset of fine-grained emotions. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4040–4054.

Jacob Devlin, Ming Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. *NAACL HLT 2019 - 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies - Proceedings of the Conference*, 1(Mlm):4171–4186.

Jacob Devlin, Saurabh Gupta, Ross Girshick, Margaret Mitchell, and C Lawrence Zitnick. 2015. Exploring nearest neighbor approaches for image captioning. *arXiv preprint arXiv:1505.04467*.

P Ekman. 1992. Are there basic emotions? *Psychological review*, 99(3):550–553.

Jun Gao, Di He, Xu Tan, Tao Qin, Liwei Wang, and Tieyan Liu. 2018. Representation degeneration problem in training natural language generation models. In *International Conference on Learning Representations*.

Gene H Golub and Christian Reinsch. 1971. Singular value decomposition and least squares solutions. In *Linear algebra*, pages 134–151. Springer.

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

Shir Gur, Natalia Neverova, Chris Stauffer, Ser-Nam Lim, Douwe Kiela, and Austin Reiter. 2021. Cross-modal retrieval augmentation for multi-modal classification. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 111–123.

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

Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. In *International Conference on Learning Representations*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Kenton Lee, Ming-Wei Chang, and Kristina Toutanova. 2019. Latent retrieval for weakly supervised open domain question answering. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6086–6096.

Bohan Li, Hao Zhou, Junxian He, Mingxuan Wang, Yiming Yang, and Lei Li. 2020. On the sentence embeddings from pre-trained language models. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 9119–9130.

Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. Roberta: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.

Juri Opitz and Sebastian Burst. 2019. Macro f1 and macro f1. *arXiv preprint arXiv:1911.03347*.

Nils Reimers and Iryna Gurevych. 2019. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992.

Klaus R Scherer and Harald G Wallbott. 1994. Evidence for universality and cultural variation of differential emotion response patterning. *Journal of personality and social psychology*, 66(2):310.

Jianlin Su, Jiarun Cao, Weijie Liu, and Yangyiwen Ou. 2021. Whitening sentence representations for better semantics and faster retrieval. *arXiv preprint arXiv:2103.15316*.

Shuohang Wang, Yichong Xu, Yuwei Fang, Yang Liu, Siqi Sun, Ruochen Xu, Chenguang Zhu, and Michael Zeng. 2022. Training data is more valuable than you think: A simple and effective method by retrieving from training data. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3170–3179.

Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. Xlnet: Generalized autoregressive pretraining for language understanding. *Advances in neural information processing systems*, 32.

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.