*k*NN-CM: A Non-parametric Inference-Phase Adaptation of Parametric Text Classifiers

Rishabh Bhardwaj[*] **Yingting Li**[α] [*] **Navonil Majumder**
Bo Cheng[α] **Soujanya Poria**
Singapore University of Technology and Design, Singapore
[α] Beijing University of Posts and Telecommunications
rishabh_bhardwaj@mymail.sutd.edu.sg, cindyyting@bupt.edu.cn
navonil_majumder@sutd.edu.sg, chengbo@bupt.edu.cn
sporia@sutd.edu.sg

Abstract

Semi-parametric models exhibit the properties of both parametric and non-parametric modeling and have been shown to be effective in the next-word prediction language modeling task. However, there is a lack of studies on the text-discriminating properties of such models. We propose an inference-phase approach—*k*-Nearest Neighbor Classification Model (*k*NN-CM)—that enhances the capacity of a pre-trained parametric text classifier by incorporating a simple neighborhood search through the representation space of (memorized) training samples. The final class prediction of *k*NN-CM is based on the convex combination of probabilities obtained from *k*NN search and prediction of the classifier. Our experiments show consistent performance improvements on eight SuperGLUE tasks, three adversarial natural language inference (ANLI) datasets, 11 question-answering (QA) datasets, and two sentiment classification datasets. The source code of the proposed approach is available at https://github.com/Bhardwaj-Rishabh/kNN-CM.

1 Introduction

The recent advancements in Natural Language Processing (NLP) have largely been attributed to the learning of contextual representations of text from large language models acting as a backbone. Most of these language models, such as versions of BERT (Devlin et al., 2018), GPT (Radford et al., 2018), T5 (Raffel et al., 2020), are parametric, i.e., they encode information required to solve a task purely in its parameters.

A parametric model, irrespective of the size of the dataset, assumes that the output variables are dependent on the input variables through a predefined class of functions. The exact function is ascertained by learning its fixed set of parameters. For instance, a linear regression model fits a set of parameters in a function that defines a supposedly linear relationship between (independent) input variables and (dependent) output variables. As a complex composition of many linear regressions, many neural architectures, such as Transformer-based models (Vaswani et al., 2017), can thus be classified as purely parametric.

However, there has been little research on the utility of non-parametric models for NLP. In contrast to parametric models which need a predefined function, such as a linear model, non-parametric models seek training data to help define the function form itself. Thus, they provide flexibility to fit a wide range of possible shapes of ground truth function. A widely known non-parametric model is the *k*-nearest neighbor (*k*NN) where inference on test samples are drawn from the information provided by the neighborhood formed by train set samples (Fix and Hodges, 1989).

A *k*NN model provides memorization capabilities and captures rare patterns from the training set that otherwise are ignored by a parametric model (as studied by Khandelwal et al. (2019)). Language models (LMs) with non-parametric properties have shown impressive gains in next-word prediction tasks (Yogatama et al., 2021; Bhardwaj et al., 2022; He et al., 2021; Khandelwal et al., 2019). Additionally, these models do not need explicit parameter learning via optimization, thus cutting the model training time completely—the lack of such characteristics in a purely parametric model motivates the proposed approach.

This work explores the importance of querying neighbors to solve classification tasks in the text domain. We hypothesize that underlying language model representations have a high-dimensional spatial proximity relation between input instances which can be leveraged to enhance prediction performance—beyond the capacity of the classifier. Hence, we propose a semi-parametric model *k*NN-CM (***k* Nearest Neighbor Classification Model**)

[*]Equal Contribution.

which constitutes a parametric classifier and a non-parametric memory (i.e. datastore) for neighborhood retrieval. In reinforcement learning (RL), classifiers are often employed as tools to aid policy learning or state representation. They can help estimate the quality of different actions or policies, and classify text into different categories, thus possessing high utility for the recent paradigm shifts in generative models (von Werra et al., 2020).

Contributions. We propose an inference phase semi-parametric modeling approach kNN-CM that enhances the capacity of a given parametric classifier model by incorporating an external memory datastore. In the inference phase, kNN-CM performs a k-neighborhood search through the datastore and merges the neighborhood predictions with the prediction obtained from the parametric classifier. Since the expansion of the CM to kNN-CM happens in the inference phase, it allows one to enhance the capacity of most of the existing pre-trained neural classifiers. By performing an extensive set of experiments, we demonstrate the importance of neighborhood search through the memorized samples on eight SuperGLUE tasks, three NLI datasets, 11 QA tasks, and two aspect-based sentiment classification tasks. We also show how the semi-parametric method can still outperform CM in out-of-domain scenarios. Furthermore, we test kNN-CM by tasking it under various cases of domain adaptation. Since kNN-CM introduces prediction latency compared with CM, we demonstrate how one can employ an entropy-based divergence measure to filter out the samples that use kNN retrieval facility. Additionally, we illustrate the importance of memorization in the low-resource scenario. In the end, we point out potential extensions of the proposed approach in conversation modeling and continual learning.

2 Related Work

Computer vision. For the image captioning task, Karpathy and Fei-Fei (2015); Devlin et al. (2015) proposed nearest neighbor baselines where they assign an unseen sample the (consensus of) captions of the training set images closest to it. Wang et al. (2019b) studied the utility of (transformed) neighborhood while performing few-shot object classification tasks. kNN has also been used to analyze learned image representations (Wallace and Hariharan, 2020) as well as to classify images (Zhang et al., 2023). For instance, Wu et al. (2018) performed image classification without class supervision but considering every instance as a class.

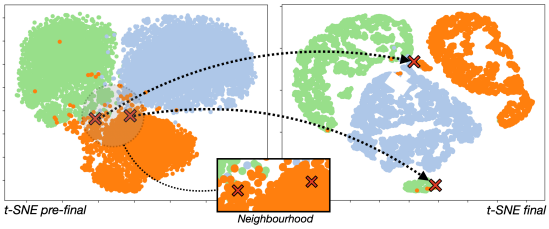

Figure 1: Motivation behind nearest neighborhood approach. The left and right figures show the pre-final layer and final layer mapping of ANLI training samples. The red crosses represent test samples where CM prediction is incorrect, which are corrected after incorporating predictions from kNN.

Recommendation systems. For session-based recommendation (based on current user interaction with the system), Kamehkhosh et al. (2017); Jannach and Ludewig (2017) showed that a neighborhood-based model output performs a GRU-based neural model on the studied tasks. kNN has been widely popular in collaborative filtering (memory-based) where the recommendation is done by the user or item neighborhood search (Su and Khoshgoftaar, 2009; Sarwar et al., 2001).

Language models. The advantage of querying nearest neighbors from a set of pre-trained LM's representations of the training set (datastore) was first observed by Khandelwal et al. (2019) by proposing kNN-LM. This is followed by several works such as improving retrieval speed (He et al., 2021), kNN-LM adaptation (Bhardwaj et al., 2022), adaptive interpolation (Drozdov et al., 2022; Yogatama et al., 2021), and masked language modeling (Min et al., 2022).

To the best of our knowledge, this work is the first attempt to extensively study the importance of neighborhood search for text classification to enhance the capacity of parametric classifiers in the inference phase. Figure 1 gives an illustrative to motivate the idea.

3 Methodology

For a given domain, we obtain training data $S{:=}\{(x_1, y_1) \ldots (x_N, y_N)\}$ where x_i denotes input text in instance space $\mathcal{X}$ and y_i is the label of its class defined in label space $\mathcal{Y}$. A learning system, a classifier, comes up with a prediction rule (hypothesis) h that maps the input from $\mathcal{X}$ to a probability distribution over class labels in $\mathcal{Y}$.

Classifier (CM). Without the loss of generality, we consider a CM constitutes a backbone large language model and a trainable classification head. During the pre-training phase, the language model is assumed to have learned semantic high-dimensional text representations that can be tuned to solve a given downstream task. The input to CM is a sequence of tokens $x = \{w_1, \ldots, w_n\}$ and output is a probability distribution over the class labels. CM is tasked to approximate the ground truth input-output mapping by learning parameters of a predefined function, i.e., the neural connections within CM. We denote the task-specific trained parametric predictor by h_{CM}.

***k*NN.** We use a well-known algorithm *k*-nearest neighbors for non-parametric modeling (Fix and Hodges, 1989). Using training samples of a task, we construct a datastore $\mathcal{D}$=$\{v(x_i), y_i\}_{i=1}^{N}$, where $v(x)$ denotes the high-dimensional vector embeddings of text x obtained from the classifier model. For a given classification task, an unseen test sample $\hat{x}$ is classified based on the nearest neighborhood training samples $\{x_1, \ldots, x_N\}$. Let $\arg\min_k$ denote the index of k training samples that return least k distance values from $\hat{x}$,

$$\begin{aligned} \mathcal{K} &\coloneqq \underset{i \in [N]}{\arg\min_k} \; d(v(\hat{x}), v(x_i)) \\ p(y \in \mathcal{Y}) &\coloneqq \frac{\sum_{i \in \mathcal{K}} \mathbb{1}[y_i == y]}{k} \end{aligned} \tag{1}$$

where $d(\cdot)$ denotes the distance function between $v(\hat{x})$ and $v(x_i)$ [1], y_i denotes the label of x_i. Similar to Khandelwal et al. (2019); Bhardwaj et al. (2022), we use euclidean distance for $d(\cdot)$. Hence, we obtain a non-parametric hypothesis $h_{k\text{NN}}$. We define the semi-parametric classifier model *k*NN-CM as a linear combination of the two probability distributions with coefficient $\lambda \in [0, 1]$

$$h_f \coloneqq \lambda \; h_{k\text{NN}} + (1 - \lambda) \; h_{\text{CM}}. \tag{2}$$

There are several aspects of this formulation:

- While performing *k*NN search, parametric classifier parameters are kept frozen.
- Strong dependence of $h_{k\text{NN}}$ on h_{CM}: Unlike commonly used ensemble methods where the underlying classifiers undergo independent training and inference, the errors made by nearest neighbor classifier highly depends on the effectiveness of its search space (datastore), which is defined by the vector representations of text provided by CM.
- Explicit control over a model capacity: Integrating *k*NN with CM provides explicit control over the model's capacity. For instance, a change in the *k* value changes the model's bias and variance as shown in the non-parametric estimator's study by Geman et al. (1992). Changing a model's bias-variance characteristics directly affects the model's capacity to fit a wider class of functions[2].
- We hypothesize that neighborhood search is important when the classifier is confused between classes and prone to do mistakes around the decision boundary. We quantify this aspect and call a model more confused if the classifier's output probabilities resemble a uniform distribution. Thus, one can choose between CM and *k*NN-CM depending on the unseen sample under testing. We study this aspect in detail in Section 5.

Next, we define the error made by h_f over m test samples

$$\epsilon \coloneqq \frac{|\{i \in [m] : \arg\max_{j \in \mathcal{Y}} [h_f^j(x_i)] \neq y_i\}|}{m} \tag{3}$$

where h_f^j is probability assigned by the hypothesis to class j, $|\cdot|$ is the cardinality of the set and $[m] = \{1, \ldots, m\}$. Note that $1 - \epsilon$ denotes the accuracy of the semi-parametric classifier.

Time and Space Complexity. Similarity search can be computationally expensive and introduce high memory requirements for each task, thus we use Faiss (Johnson et al., 2019)—an efficient similarity search algorithm and clustering (indexing) algorithm of high dimensional data. The clustering of similar vectors in high dimensions obviates the need to search through the whole training set (datastore). For small-scale datasets, we use IndexFlatL2 which queries through all the vectors in the datastore. The complexity is thus $\mathcal{O}(n)$ where n is the number of elements in the datastore. For large-scale datastores, we use IndexFlatIVF to first

[1] Moreover, one can weight each prediction with $\exp(-d(v(\hat{x}), v(x_i)))$.

[2] Broadly, a model with $k = N$ will learn fewer patterns from data while a model with $k = 1$ can learn as many patterns as N, where N is the number of training samples.

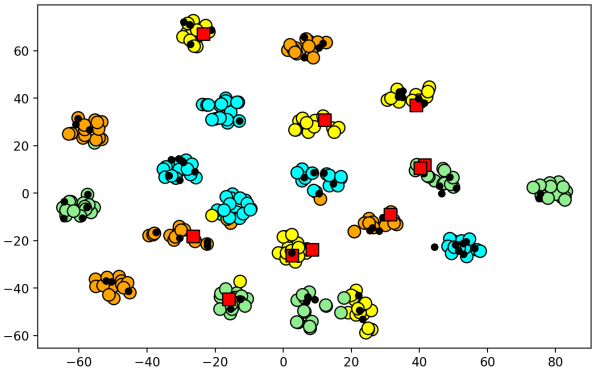

Figure 2: Error analysis of a neural classifier on clustered synthesised data.

cluster the vectors in datastore and then perform an NN search in each cluster. The time complexity of this method is $\mathcal{O}(n_c m)$ where n_c and m denote the number of clusters and the average number of elements in clusters, respectively. The space complexity of IndexFlatL2 is $\mathcal{O}(nd_s)$ and IndexFlatIVF is $\mathcal{O}(nd_s + md_s)$, where d_s denotes the dimensionality of the vector in the datastore. Contrary to non-parametric, the time and space complexity of a parametric model, such as CM, is predefined and does not vary with the number of train samples.

4 Experiments

4.1 Toy dataset

We hypothesize that a CM is prone to miss neighborhood cues from the training data. To test this, we set up a toy experiment on a neural network comprised of one hidden layer of 100 nodes activated with ReLU. To test the capacity of this network, we synthesize a dataset by randomly initializing 20 cluster centers $\{c_i : c_i \sim N(0, 1.5), i \in 1 \ldots 20\}$, each of which constitutes 20 points $\{c_{ij} : c_{ij} = c_i + p_j; p_j \sim N(0, 1), j \in 1 \ldots 20\}$, where cluster center and p_j are independently sampled for each of their dimensions. All the data points lie in the space of $\mathbb{R}^{100}$. We randomly split the clusters into four classes. Figure 2 shows the 2-dimensional t-SNE plot of the generated data with samples shown in the same color belonging to the same class and vice versa. Circles represent samples used to learn the network parameters, black dots denote the correctly classified test cases and the red squares denote the test samples incorrectly classified by the network. Red squares provide evidence for the hypothesis, i.e., while the model is able to identify correct clusters of several test cases, it still fails to capture the nuance of the neighborhood precisely.

4.2 NLP datasets

We base our main experiments on the SuperGLUE benchmark and a large variety of existing NLP datasets to solve NLI, Question Answering (QA), and Sentiment Classification.

SuperGLUE (Wang et al., 2019a). It is a benchmark dataset to evaluate a model on its language understanding capabilities. `BoolQ` (Clark et al., 2019) is a QA task where a yes/no question is asked on a short passage. `CB` (De Marneffe et al., 2019) is a textual entailment task where given a premise, the model is asked to predict how committed the author is toward the truth of the (clause) hypothesis. `COPA` (Roemmele et al., 2011) is a causal reasoning task to identify the cause or effect of a premise from the set of given choices. `MultiRC` (Khashabi et al., 2018) is a multi-choice QA task where a question is asked about a context paragraph and the answer choices are provided. `ReCoRD` (Zhang et al., 2018) is a multi-choice QA task where, from a passage, a masked-out entity is to be predicted from a set of entities. `RTE` (Haim et al., 2006) is another textual entailment dataset with two classes, entailment and not entailment. `WiC` (Pilehvar and Camacho-Collados, 2018) is a task of word sense disambiguation. Given two texts and a word appearing in both sentences, the task is to determine if the word is used in the same sense in both sentences. `WSC` (Levesque et al., 2012) is a conference resolution task where an example consists of a pronoun and a list of noun phrases, the task is to identify the correct pronoun referent.

`BoolQ`, `COPA`, `COPA`, `WiC`, `WSC`, and `RTE` are binary classification tasks, `CB` three-class classification, `MultiRC` and `ReCoRD` are cast as binary class classification where the correct choice (or entity) is labeled 1 and incorrect is labeled as 0.

ANLI (Nie et al., 2020). Adversarial Natural Language Inference is a large-scale benchmark NLI dataset constructed by an adversarial human-model-in-loop. The dataset is subdivided into three datasets `A1`, `A2`, and `A3` with increasing task difficulty. ANLI aims to solve a textual entailment task where given a premise, the model is asked to predict if a hypothesis entails, contradicts, or is neutral to the premise. We use `ANLI` to represent combination of `A1`, `A2`, and `A3`.

Question Answering. For QA tasks, we experiment on ten datasets: QASC (Question Answering via Sentence Composition) (Khot et al., 2020) is a fact retrieval from a large corpus to answer a question given eight choices, only one of which is correct. PIQA (Physical IQA) (Bisk et al., 2020) tests physical knowledge of language models by asking them to select the correct choice from the given two. SIQA (Social IQA) (Sap et al., 2019) is a common sense reasoning dataset for the context of social situations. Given a social situation and three choices to select from, the task is to select the correct choice. CQA (CommonsenseQA) (Talmor et al., 2019) is a commonsense QA based on ConceptNet knowledge (Speer et al., 2017). For a question, the task is to choose one of five given choices. CQA-2 (CommonsenseQA 2.0) (Talmor et al., 2021) is another recent commonsense QA dataset constructed with model-in-loop approach. It consists of commonsense questions from various categories of reasons, with the answer being *yes* or *no*. SWAG and (H-SWAG) (Zellers et al., 2018) are datasets for grounded inference. Given an incomplete event description, the task is to find the correct ending from a set of four choices. CosmosQA (Huang et al., 2019) is a dataset for commonsense-based reading comprehension. The task is to identify a correct choice from the given four for the question asked about a paragraph. CICERO v1, v2 (Ghosal et al., 2022b; Shen et al., 2022) are dialogue QA dedicated datasets. Given a question about a given utterance taken from a dialogue, the task is to choose the correct answer from the choices.

Aspect-Based Sentiment Classification. We also compare the proposed approach on two aspect-based sentiment classification datasets—Laptop and Restaurant. The datasets are a set of restaurant and laptop reviews obtained from Pontiki et al. (2015, 2016)[3]. We convert the given review in the form $w_1\ w_2 \ldots$ <aspect term> $\ldots\ w_n$, where <·> encloses the aspect term for which the sentiment (positive/negative/neutral) is to be predicted.

4.3 Experimental Setup

The *k*NN-CM is an inference phase approach that does not require task-specific CM parameter tuning. We either train a classifier or utilize existing pre-trained task-specific classifiers to obtain a baseline CM on a given task.

[3]We utilize the data collected by Zhou et al. (2021)

CM Setup. For all the tasks in the SuperGLUE benchmark, we utilize RoBERTa-base ((Liu et al., 2019)) as the backbone language model. Following the success of parameter-efficient adapters (Houlsby et al., 2019), and their competitive performance with full-mode fine-tuning (Liu et al., 2022; Hou et al., 2022; Bhardwaj et al., 2022), we obtain a task-specific classifier (CM) by training adapter modules inserted in between LM layers and attaching a classification module (head) on top of the LM [4]. All the tasks are formulated as a classification problem [5]. We follow a similar setup for language inference tasks (ANLI) and sentiment analysis tasks. For QA datasets, we use the DeBERTa-large (He et al., 2020) based classifier. Following TEAM (Ghosal et al., 2022a) which has shown a better than baseline performance on numerous QA tasks, we formulate all the multi-choice QA tasks as binary classification where the correct choices are labeled as 1 and incorrect choices are labeled as 0. Therefore, in the training phase, the classifier model aims to minimize the binary-cross entropy objective function. During the inference phase, we select the choice with the maximum class 1 probability score. Since our approach improves the model performance in the inference phase, we liberate ourselves from classifier training by downloading the model checkpoints generously provided by Ghosal et al. (2022a). The classification head uses <s> from RoBERTa and [CLS] from DeBERTa (generally used as classification tokens).

***k*NN Setup.** For each task under study, we use the task-specific trained CM obtained via the method described above and construct a datastore using the train set samples. We obtain hidden representations of each sample by performing one forward pass through CM. For fast neighbor search and making the datastore memory-efficient, the obtained vectors are indexed using Faiss.

Hyperparameters. Since the approach is applicable to the inference phase, the primary set of hyperparameters comes from *k*NN search. For each task, we find the best interpolation parameter λ (Equation (2)) and the optimal number of neighbors to search k using the validation set, $\lambda \in \{0.001, 0.01, 0.1, 0.2, \ldots, 0.8, 0.9, 0.99, 0.999\}$ and $k \in \{1, 2, 4, 8, 16, 32, 64, 128, 256, 512\}$.

[4]Trainable parameters are $\approx$1.2% of classifier parameters.

[5]Datasets downloaded from https://huggingface.co/datasets/super_glue

Task	CM		*k*NN-CM	
	Acc.	F1	Acc.	F1
CB	91.07	91.04	**92.86**$_{(\uparrow 1.96\%)}$	**92.37**$_{(\uparrow 1.46\%)}$
COPA	56.00	55.84	**64.00**$_{(\uparrow 14.29\%)}$	**63.87**$_{(\uparrow 14.38\%)}$
WSC	63.46	44.29	63.46$_{(\uparrow 0.00\%)}$	**47.41**$_{(\uparrow 7.04\%)}$
RTE	77.98	77.6	**79.42**$_{(\uparrow 1.85\%)}$	**79.18**$_{(\uparrow 2.04\%)}$
WiC	69.12	69.08	**69.28**$_{(\uparrow 0.23\%)}$	**69.28**$_{(\uparrow 0.29\%)}$
BoolQ	80.49	78.62	**81.16**$_{(\uparrow 0.83\%)}$	**79.86**$_{(\uparrow 1.57\%)}$
MultiRC	66.56	60.73	**70.40**$_{(\uparrow 5.77\%)}$	**69.26**$_{(\uparrow 14.04\%)}$
ReCoRD	61.17	61.82	**62.05**$_{(\uparrow 1.43\%)}$	**62.72**$_{(\uparrow 1.45\%)}$

Table 1: Performance comparison of *k*NN-CM vs. CM on the SuperGLUE development set. The results are reported on the validation set.

5 Results and Discussion

Table 1 shows results on SuperGLUE datasets. We observe *k*NN to correct predictions of CM in textual entailment tasks such as CB and RTE. The assistance of neighbors from the hidden space representations also shows a huge improvement (by $\approx 7\%$) to resolve the ambiguity in pronoun in WSC. However, the improvement in WiC is comparably less ($\approx 0.3\%$). After investigation, we found that CM and *k*NN share the same set of samples where they make erroneous predictions. The conclusions are made on relatively low improvements on BoolQ when compared with MultiRC and ReCoRD. Amongst all, we observe a huge improvement for over 14% in COPA. We notice that *k*NN alone can surpass the baseline COPA accuracy by over three points. While a combination of both gives a boost of over eight points in both performance matrices.

Table 2 shows the improvement due to *k*NN involvement during predictions. We find the neighborhood search to help more as the task complexity increases, thus the observed improvement for A3 is more than the other adversarial datasets. More improvement in F1 score indicates that neighborhood search is less impacted by the class imbalance when compared with the CM-only setting. The best k identified for ANLI tasks is between 1-4 with high λ (≈ 0.99), thus, reflecting the information provided by the closest few neighborhoods is important and sufficient.

In QA tasks (Table 3), the observed improvement is relatively lower when compared with SuperGLUE and ANLI. After investigation, we observed the prediction that the predictions made by CM and *k*NN are similar. This indicates an effective clustering by the (DeBERTa-large) CM to perform the binary classification task. For instance, on SIQA, the instance accuracy of *k*NN is 80.45% while the CM performance is 80.98% with errors made on a similar set of samples.

Task	CM		*k*NN-CM	
	Acc.	F1	Acc.	F1
A1	46.90	46.70	**47.30**$_{(\uparrow 0.85\%)}$	**47.04**$_{(\uparrow 0.73\%)}$
A2	43.20	42.33	**44.10**$_{(\uparrow 2.08\%)}$	**43.53**$_{(\uparrow 2.83\%)}$
A3	42.33	40.16	**45.08**$_{(\uparrow 6.50\%)}$	**44.72**$_{(\uparrow 11.35\%)}$
ANLI	41.84	40.32	**44.72**$_{(\uparrow 6.88\%)}$	**43.71**$_{(\uparrow 8.41\%)}$

Table 2: Results on the test set of NLI tasks. ANLI is a combined dataset A1, A2, A3.

Task	CM		*k*NN-CM	
	Bin.	Inst.	Bin.	Inst.
QASC	91.27	75.05	91.10	**75.92**$_{(\uparrow 1.15\%)}$
QASC-IR	95.39	88.01	95.40	**88.44**$_{(\uparrow 0.49\%)}$
PIQA	73.59	86.45	73.91	**86.94**$_{(\uparrow 0.57\%)}$
SIQA	81.01	80.71	80.98	**81.27**$_{(\uparrow 0.69\%)}$
CQA	88.06	82.80	88.12	**83.29**$_{(\uparrow 0.59\%)}$
CQA-2	54.37	57.97	61.20	**60.84**$_{(\uparrow 4.95\%)}$
SWAG	91.46	93.14	91.47	**93.24**$_{(\uparrow 0.11\%)}$
CosmosQA	87.63	86.30	87.86	**86.77**$_{(\uparrow 0.54\%)}$
H-SWAG	94.44	96.15	94.74	**96.24**$_{(\uparrow 0.09\%)}$
CICERO-v1	88.53	83.55	88.50	**83.70**$_{(\uparrow 0.18\%)}$
CICERO-v2	87.62	90.34	87.67	**90.56**$_{(\uparrow 0.24\%)}$

Table 3: *k*NN-CM vs. CM on QA datsets. Bin. and Inst. denote binary and instance classification accuracy.

Table 4 shows the results on sentiment analysis tasks. Similar to WiC, the reasons for poor performance on Restaurant were found to be the same set of erroneous predictions, and thus no explicit probability corrections were made by the neighbor samples. In contrast to this, we observed good performance on Laptop because the nearest neighbors help boost recall and precision.

Out-Of-Domain Performance. We retrieve SuperGLUE diagnostic datasets AX_b and AX_g (test only) and perform ANLI out-of-domain evaluations [6]. Table 5 shows that the neighbor search on ANLI datastore not only improves on in-domain datasets but also shows over 12% F1 improve-

[6]Since we are evaluating OOD performance, we use the same metric as used in ANLI evaluation.

Task	CM		*k*NN-CM	
	Acc.	F1	Acc.	F1
Laptop	88.42	77.96	**88.56**$_{(\uparrow 0.16\%)}$	**79.94**$_{(\uparrow 2.54\%)}$
Restaurant	88.01	79.13	**88.19**$_{(\uparrow 0.20\%)}$	**79.36**$_{(\uparrow 0.29\%)}$

Table 4: *k*NN-CM vs. CM performance on test split of sentiment classification datasets.

Task	CM		*k*NN-CM	
	Acc.	F1	Acc.	F1
AX_b	59.78	49.30	**61.32**$_{(\uparrow 2.58\%)}$	**55.58**$_{(\uparrow 12.74\%)}$
AX_g	**50.84**	37.80	50.28$_{(\downarrow -1.10\%)}$	**39.37**$_{(\uparrow 4.15\%)}$

Table 5: ANLI out of domain evaluation on AX_b, AX_g.

Metric	CM_a	kNN_c CM_u	kNN_a CM_a	kNN_c CM_a	kNN_{a+c} CM_a
Acc.	41.07	50.00	39.29	**75.00**	60.71
F1	32.19	34.73	32.47	**53.82**	46.58

Table 6: ANLI→CB domain adaptation without CM fine-tuning, by adding domain-specific datastore. kNN_c-CM_u is a *k*NN-only classifier that constructs a datastore from RoBERTa-base LM. CM_a denotes an ANLI classifier. *k*NN subscripts 'a' and 'c' indicate the datastore is constructed using ANLI or/and CB training set.

ment on AX_b and around 4% improvements on AX_g OOD datasets. There are improvements in Acc. for AX_b and an observed poorer performance (by ≈1%) of *k*NN-CM on AX_g. To investigate it further, we found *k*NN to improve the precision and recall score of the poor performing class by slightly trading-off with the precision and recall of higher performing class, the overall impact improves F1 significantly, however, accuracy degrades.

Domain Adaptation without Classifier Tuning. We carry out *k*NN-CM domain adaptation of ANLI to CB without explicitly fine-tuning the classifier but including the domain-specific datastore. In Table 6, we observe the datastore from CB domain is important to help boost the performance significantly. Even on an untrained classifier, merely including domain-specific datastore constructed on purely pre-trained LM (CM_u) and classifying using *k*NN-only, gives 50% accuracy. The best-performing model is kNN_c-CM_a (a CB datastore constructed on ANLI classifier) with an accuracy of around 75% and F1 score of over 53%. Merging available ANLI datastore with CB datastore, however, tends to reduce the performance. We posit the reason is a very small fraction of neighbors belonging to CB as compared to ANLI (≈0.15%)[7]. Rescoring methods can help adapt the existing datastore to other domains (Bhardwaj et al., 2022).

Filtering Samples for *k*-NN Search. As discussed in section 3, we hypothesize the neighborhood search is important for CM to make important decisions around the decision boundaries which leads to model confusion. We assume the model to be more in need of external aid when the CM predictions are close to the uniform distribution over labels. Thus, we define a neighbor requirement score r for a given text x as the normalized KL-divergence of CM prediction with respect to a discrete uniform distribution over classes:

$$r(x) := \frac{\mathrm{KL}(h_{\mathrm{CM}}(x) || \, \mathcal{U}_{\mathcal{Y}})}{\log(|\mathcal{Y}|)}.$$

For a given input x, we redefine the predictor:

$$h_f(x) := \begin{cases} \lambda \; h_{k\mathrm{NN}}(x) + (1-\lambda) \; h_{\mathrm{CM}}(x), & \text{if } r(x) \leq \tau \\ h_{\mathrm{CM}}(x), & \text{otherwise} \end{cases}$$

where $\lambda \in [0, 1]$, $|\mathcal{Y}|$ (cardinality of the label set) denote the number labels, $\mathcal{U}_{\mathcal{Y}}$ denotes the uniform distribution over the labels in the set $\mathcal{Y}$. $h^i_{\mathrm{CM}}(x)$ is classifier's probability over label i, τ defines a threshold on divergence value below which *k*NN will be involved in model predictions. In Figure 3, we observe ANLI accuracy to converge at $\tau = 0.7$. Thus, using entropy-based measures, one can filter samples for *k*NN to reduce inference time.

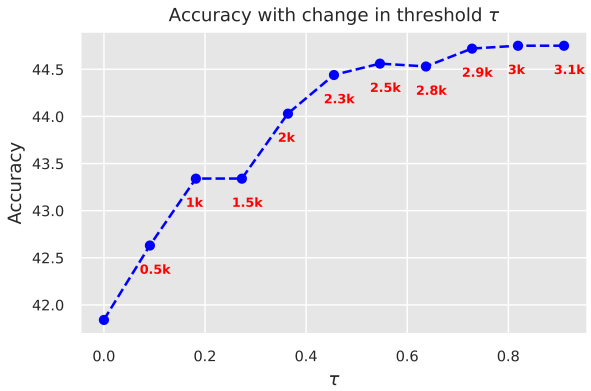

Figure 3: Impact of τ on ANLI accuracy. Red annotations are the number of samples query *k*NN.

Layer Importance for Retrieval. Following Khandelwal et al. (2019); Bhardwaj et al. (2022), we create datastore on representations obtained from different layers of ANLI-based classifier and perform hyperparameter search (k and λ) on ANLI development set. Figure 4 shows the layer-wise test set performance increases when we go deeper in the network. For ANLI, we find the best-performing neighborhood representations to be layer normalization in the pre-final layer. In our initial experiments on the SuperGLUE benchmark, on average, we found the final layer to be the best performing.

[7]We use hyperparameters k and λ, obtained from ANLI dev set, across the settings in Table 6.

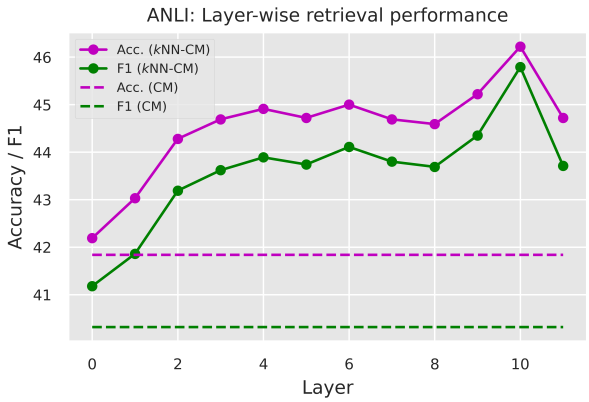

Figure 4: Impact of different layers on ANLI. The horizontal axis denotes layer representations considered for datastore construction and the vertical axis denotes the test accuracy/F1.

Task	CM		*k*NN-CM	
	Acc.	F1	Acc.	F1
20%	33.37	16.68	35.38	35.09
40%	33.37	16.68	33.37	31.18
60%	33.37	16.68	36.41	36.21
80%	46.47	44.90	48.09	47.43
100%	41.84	40.32	44.72	43.71

Table 7: Low-resource performance on ANLI.

Low-Resource. In Table 7, we observe that when data availability is reduced below 60%, the classification model performs worse equivalent to a random classification with uniform probabilities. When *k*NN is introduced, the performance of the *k*NN-CM model tends to be better than assigning the random labels to the test instances. It shows that, even in low resource cases, there are clustered vectors that *k*NN can exploit to boost the classification performance, while large parametric classifiers fail to capture proximity relations. This to some extent provides evidence to the hypothesis of Khandelwal et al. (2019), i.e., learning the similarity between texts is easier than predicting the next word, given our problem is reformulated as a label (text) generation. On the other hand, when the amount of training data is greater than 80% (of full-set), our baseline performed well and *k*NN adds further improved by nearly 4%-8%. Thus, irrespective of the baseline performance, *k*NN tends to maintain a better performance than random classification and it is evident that it supports the CM relentlessly even in the low-resource regime.

***k*NN-CM Time Overhead.** Being a semi-parametric model, the *k*NN search space tends to linearly increase with the increase in the number of training samples to memorize. Thus, we study the time-overhead, i.e., added inference-time latency due to the neighborhood search. Without the loss of generality, we base our analysis on the ANLI dataset. On the CPU [8], the per-sample CM inference speech is ≈72 ms, and for the *k*NN retrieval [9], it is ≈29 ms. On the GPU [10], the per-sample in the CM stage is ≈9 ms, and for the *k*NN stage is ≈2 ms. Thus, a flat *k*NN search increases inference time by around 40% on CPU and 20% on GPU.

Utterance Classification. We also trained classifiers on datasets for emotion recognition in conversation. Given an utterance from a conversation with an appended set of eight utterances preceding it, we aim to classify it in one of the emotion classes. Our experiments on MELD (Poria et al., 2018), DailyDialogue (Li et al., 2017), and IEMOCAP (Busso et al., 2008) shows very insignificant improvements in Accuracy and F1 scores when the model is equipped with *k*NN search. We leave the precise semi-parametric modeling for utterance classification as future work.

Neighbors Weighting. We compare the frequency-based probability computations in Equation 1 with weighting neighbors with their distances from the query, thus

$$p(y \in \mathcal{Y}) := \sum_{i \in \mathcal{K}} \frac{\mathbb{1}_{[y_i == y]} \exp \frac{(-d(v(\hat{x}), v(x_i)))}{\beta}}{k}$$

In our initial experiments on QASC dataset with $\beta \in \{2, 10, 20\}$, we found the validation set performance to be $75.81\%, 75.92\%, 75.92\%$ all of which are higher than baseline CM but lower than frequency-based computations. We posit there is no generic weighting scheme that works for all the tasks, hence we leave methods to find a task-adaptive neighborhood weighting for classification for future work.

6 Conclusion

In this work, we presented *k*NN-CM, a semi-parametric paradigm that augments the inference

[8] AMD Ryzen Threadripper 3960X 24-Core.
[9] We implement IndexFlatL2 search.
[10] NVIDIA RTX A6000.

phase with neighborhood search through the training set. We studied the impact of adding non-parametric characteristics to a parametric classifier on 24 language understanding tasks. We further demonstrated the generalizability of kNN-CM by studying their out-of-domain and domain adaptation performance. We also showed its efficacy in low-resource scenarios where CM performance reduces dramatically and neighborhood search emerges as a savior. Toward the end, we leave a few important remarks on utterance classification and neighborhood weighting that carry the potential to motivate future research directions, which we elaborate on in the limitations section.

7 Limitations

We discuss the potential limitations of semi-parametric modeling and considerable future work:

- Non-parametric characteristics introduce challenges in the interpretability of the predictions.
- Since the function form highly depends on the size of the training set, the memory footprint grows linearly.
- Learning a good representation of the dataset is still a bottleneck task and predominantly relies on the parametric models. Thus, the performance and function form of non-parametric models depends on the effectiveness of the data representations.
- Since nearest neighbor computation requires pairwise similarity between test and samples in the train set, the inference time increases with the increase in the dimensionality of space and size of the train set. Several tools such as Faiss (Johnson et al., 2019) assist in a significant reduction of computational overhead with the trade-off in the performance of the model.
- One can compute kNN probabilities by using exponential of negative of distance (Khandelwal et al., 2019). However, simple averaging shows considerable improvements, and finding better probability computations is left for future work.

In the future, we see a huge potential of kNN's in tackling the catastrophic forgetting in continual learning applications involving text. Another interesting area will be to propose methods that allow task-specific datastore representation tuning, more interestingly through backpropagation. Since the datastore size increases linearly with the number of training samples, scaling semi-parametric systems can be a challenging task. Thus, deploying such systems on edge devices with constrained computational capacity and memory is another interesting future research direction.

Acknowledgement

We thank the anonymous reviewers for their constructive feedback. This project is supported by the AcRF MoE Tier-2 grant (Project no. T2MOE2008, and Grantor reference no. MOE-T2EP20220-0017) titled: "CSK-NLP: Leveraging Commonsense Knowledge for NLP", and the SRG grant id: T1SRIS19149 titled "An Affective Multimodal Dialogue System". This work is also generously supported by BUPT Excellent Ph.D. Students Foundation CX2021229.

References

Rishabh Bhardwaj, George Polovets, and Monica Sunkara. 2022. Adaptation approaches for nearest neighbor language models. *arXiv preprint arXiv:2211.07828*.

Yonatan Bisk, Rowan Zellers, Jianfeng Gao, Yejin Choi, et al. 2020. Piqa: Reasoning about physical commonsense in natural language. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 7432–7439.

Carlos Busso, Murtaza Bulut, Chi-Chun Lee, Abe Kazemzadeh, Emily Mower, Samuel Kim, Jeannette N Chang, Sungbok Lee, and Shrikanth S Narayanan. 2008. Iemocap: Interactive emotional dyadic motion capture database. *Language resources and evaluation*, 42(4):335–359.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*.

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*.

Andrew Drozdov, Shufan Wang, Razieh Rahimi, Andrew McCallum, Hamed Zamani, and Mohit Iyyer. 2022. You can't pick your neighbors, or can you? when and how to rely on retrieval in the k nn-lm. *arXiv preprint arXiv:2210.15859*.

Evelyn Fix and Joseph Lawson Hodges. 1989. Discriminatory analysis. nonparametric discrimination: Consistency properties. *International Statistical Review/Revue Internationale de Statistique*, 57(3):238–247.

Stuart Geman, Elie Bienenstock, and René Doursat. 1992. Neural networks and the bias/variance dilemma. *Neural computation*, 4(1):1–58.

Deepanway Ghosal, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2022a. Two is better than many? binary classification as an effective approach to multi-choice question answering. *arXiv preprint arXiv:2210.16495*.

Deepanway Ghosal, Siqi Shen, Navonil Majumder, Rada Mihalcea, and Soujanya Poria. 2022b. Cicero: A dataset for contextualized commonsense inference in dialogues. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5010–5028.

R Bar Haim, Ido Dagan, Bill Dolan, Lisa Ferro, Danilo Giampiccolo, Bernardo Magnini, and Idan Szpektor. 2006. The second pascal recognising textual entailment challenge. In *Proceedings of the Second PASCAL Challenges Workshop on Recognising Textual Entailment*, volume 7.

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

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. 2020. Deberta: Decoding-enhanced bert with disentangled attention. *arXiv preprint arXiv:2006.03654*.

Zejiang Hou, Julian Salazar, and George Polovets. 2022. Meta-learning the difference: Preparing large language models for efficient adaptation. *ArXiv*, abs/2207.03509.

Neil Houlsby, Andrei Giurgiu, Stanislaw Jastrzebski, Bruna Morrone, Quentin De Laroussilhe, Andrea Gesmundo, Mona Attariyan, and Sylvain Gelly. 2019. Parameter-efficient transfer learning for nlp. In *International Conference on Machine Learning*, pages 2790–2799. PMLR.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos qa: Machine reading comprehension with contextual commonsense reasoning. 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 2391–2401.

Dietmar Jannach and Malte Ludewig. 2017. When recurrent neural networks meet the neighborhood for session-based recommendation. In *Proceedings of the eleventh ACM conference on recommender systems*, pages 306–310.

Jeff Johnson, Matthijs Douze, and Hervé Jégou. 2019. Billion-scale similarity search with GPUs. *IEEE Transactions on Big Data*, 7(3):535–547.

Iman Kamehkhosh, Dietmar Jannach, and Malte Ludewig. 2017. A comparison of frequent pattern techniques and a deep learning method for session-based recommendation. In *RecTemp@ RecSys*, pages 50–56.

Andrej Karpathy and Li Fei-Fei. 2015. Deep visual-semantic alignments for generating image descriptions. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3128–3137.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. 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 252–262.

Tushar Khot, Peter Clark, Michal Guerquin, Peter Jansen, and Ashish Sabharwal. 2020. Qasc: A dataset for question answering via sentence composition. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 34, pages 8082–8090.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. Dailydialog: A manually labelled multi-turn dialogue dataset. *arXiv preprint arXiv:1710.03957*.

Haokun Liu, Derek Tam, Mohammed Muqeeth, Jay Mohta, Tenghao Huang, Mohit Bansal, and Colin Raffel. 2022. Few-shot parameter-efficient fine-tuning is better and cheaper than in-context learning. *ArXiv*, abs/2205.05638.

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*.

Sewon Min, Weijia Shi, Mike Lewis, Xilun Chen, Wentau Yih, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Nonparametric masked language modeling. *arXiv preprint arXiv:2212.01349*.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2018. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*.

Maria Pontiki, Dimitrios Galanis, Haris Papageorgiou, Ion Androutsopoulos, Suresh Manandhar, Mohammad Al-Smadi, Mahmoud Al-Ayyoub, Yanyan Zhao, Bing Qin, Orphée De Clercq, et al. 2016. Semeval-2016 task 5: Aspect based sentiment analysis. In *International workshop on semantic evaluation*, pages 19–30.

Maria Pontiki, Dimitrios Galanis, Harris Papageorgiou, Suresh Manandhar, and Ion Androutsopoulos. 2015. Semeval-2015 task 12: Aspect based sentiment analysis. In *Proceedings of the 9th international workshop on semantic evaluation (SemEval 2015)*, pages 486–495.

Soujanya Poria, Devamanyu Hazarika, Navonil Majumder, Gautam Naik, Erik Cambria, and Rada Mihalcea. 2018. Meld: A multimodal multi-party dataset for emotion recognition in conversations. *arXiv preprint arXiv:1810.02508*.

Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, Peter J Liu, et al. 2020. Exploring the limits of transfer learning with a unified text-to-text transformer. *J. Mach. Learn. Res.*, 21(140):1–67.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *AAAI spring symposium: logical formalizations of commonsense reasoning*, pages 90–95.

Maarten Sap, Hannah Rashkin, Derek Chen, Ronan Le Bras, and Yejin Choi. 2019. Social iqa: Commonsense reasoning about social interactions. 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 4463–4473.

Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl. 2001. Item-based collaborative filtering recommendation algorithms. In *Proceedings of the 10th international conference on World Wide Web*, pages 285–295.

Siqi Shen, Deepanway Ghosal, Navonil Majumder, Henry Lim, Rada Mihalcea, and Soujanya Poria. 2022. Multiview contextual commonsense inference: A new dataset and task. *arXiv preprint arXiv:2210.02890*.

Robyn Speer, Joshua Chin, and Catherine Havasi. 2017. Conceptnet 5.5: An open multilingual graph of general knowledge. In *Thirty-first AAAI conference on artificial intelligence*.

Xiaoyuan Su and Taghi M Khoshgoftaar. 2009. A survey of collaborative filtering techniques. *Advances in artificial intelligence*, 2009.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. 2019. Commonsenseqa: A question answering challenge targeting commonsense knowledge. 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 4149–4158.

Alon Talmor, Ori Yoran, Ronan Le Bras, Chandra Bhagavatula, Yoav Goldberg, Yejin Choi, and Jonathan Berant. 2021. Commonsenseqa 2.0: Exposing the limits of ai through gamification. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 1)*.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. *Advances in neural information processing systems*, 30.

Leandro von Werra, Younes Belkada, Lewis Tunstall, Edward Beeching, Tristan Thrush, and Nathan Lambert. 2020. Trl: Transformer reinforcement learning. https://github.com/lvwerra/trl.

Bram Wallace and Bharath Hariharan. 2020. Extending and analyzing self-supervised learning across domains. In *European Conference on Computer Vision*, pages 717–734. Springer.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019a. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Y Wang, WL Chao, KQ Weinberger, and L Simpleshot van der Maaten. 2019b. Revisiting nearestneighbor classification for few-shot learning. *Preprint*.

Zhirong Wu, Yuanjun Xiong, Stella X Yu, and Dahua Lin. 2018. Unsupervised feature learning via non-parametric instance discrimination. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 3733–3742.

Dani Yogatama, Cyprien de Masson d'Autume, and Lingpeng Kong. 2021. Adaptive semiparametric language models. *Transactions of the Association for Computational Linguistics*, 9:362–373.

Rowan Zellers, Yonatan Bisk, Roy Schwartz, and Yejin Choi. 2018. SWAG: A large-scale adversarial dataset for grounded commonsense inference. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing*, pages 93–104, Brussels, Belgium. Association for Computational Linguistics.

Renrui Zhang, Liuhui Wang, Ziyu Guo, and Jianbo Shi. 2023. Nearest neighbors meet deep neural networks for point cloud analysis. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 1246–1255.

Sheng Zhang, Xiaodong Liu, Jingjing Liu, Jianfeng Gao, Kevin Duh, and Benjamin Van Durme. 2018. Record: Bridging the gap between human and machine commonsense reading comprehension. *arXiv preprint arXiv:1810.12885*.

Yan Zhou, Fuqing Zhu, Pu Song, Jizhong Han, Tao Guo, and Songlin Hu. 2021. An adaptive hybrid framework for cross-domain aspect-based sentiment analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 35, pages 14630–14637.