kNN-Prompt: Nearest Neighbor Zero-Shot Inference

Weijia Shi Julian Michael Suchin Gururangan Luke Zettlemoyer
Paul G. Allen School of Computer Science & Engineering,
University of Washington, Seattle, WA
{swj0419, julianjm, sg01, lsz}@cs.washington.edu

Abstract

Retrieval-augmented language models (LMs) use non-parametric memory to substantially outperform their non-retrieval counterparts on perplexity-based evaluations, but it is an open question whether they achieve similar gains in few- and zero-shot end-task accuracy. We extensively study one such model, the k-nearest neighbor LM (kNN-LM), showing that the gains marginally transfer. The main challenge is to achieve coverage of the verbalizer tokens that define the different end-task class labels. To address this challenge, we also introduce kNN-Prompt, a simple and effective kNN-LM with automatically expanded fuzzy verbalizers (e.g. to expand “terrible” to also include “silly” and other task-specific synonyms for sentiment classification). Across nine diverse end-tasks, using kNN-Prompt with GPT-2 large yields significant performance boosts over strong zero-shot baselines (13.4% absolute improvement over the base LM on average). We also show that other advantages of non-parametric augmentation hold for end tasks; kNN-Prompt is effective for domain adaptation with no further training, and gains increase with the size of the retrieval model.

1 Introduction

Retrieval-augmented language models (LMs) have access to a non-parametric memory, allowing them to directly access a large external text collection during inference. Previous work has shown that these models substantially outperform their non–retrieval-based counterparts on language modeling tasks (Khandelwal et al., 2020; He et al., 2021; Borgeaud et al., 2021), but it is an open question whether they also achieve similar gains in few-shot and zero-shot end task evaluations (Radford et al., 2019; Brown et al., 2020a). In this paper, we demonstrate that, with some extensions to improve coverage of the verbalizer tokens, the performance gains of retrieval-augmented LMs generalize well to a wide range of downstream tasks.

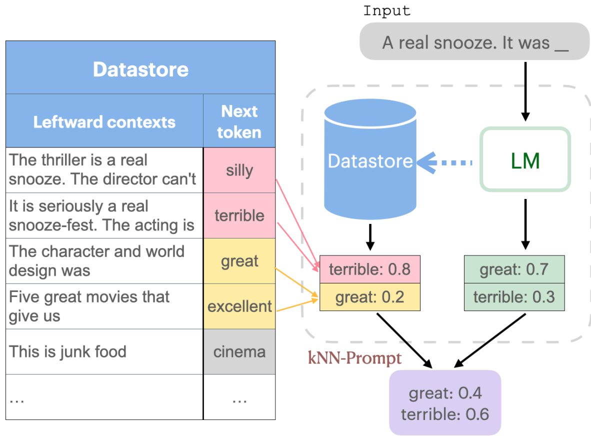

Figure 1: kNN-Prompt incorporates information from a large, heterogeneous corpus (unlabeled texts from different domains) to facilitate few- and zero-shot inference. The datastore contains key-value pairs where the key is an encoding of a leftward context and the value is the next token following the context. Our fuzzy verbalizer expands "terrible" to include "silly" and "great" to include "excellent". Because the encoded corpus is unlabeled plain text, some datastore entries contain next tokens not in the verbalizer tokens (e.g., "cinema").

We study the k-nearest neighbors language model (Khandelwal et al., 2020, kNN-LM), which interpolates the LM softmax distribution with a nearest-neighbor distribution. The nearest neighbours are computed based on the distance in LM output embeddings and can be drawn from any text corpus, in our case, a heterogeneous corpus that contains unlabeled data from different domains. We are the first to study the zero-shot application of kNN-LM to end tasks, and we find that applying the technique naïvely produces only marginal improvements (Section 4). The main challenge is that the support of the kNN distribution is sparse (covering at most k tokens, often less), as it only assigns probability mass to nearest neighbors. This means it often entirely misses the tokens that are used to verbalize the output label in the standard application of LMs to zero-shot classification: across the

Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, pages 3254–3265
December 7-11, 2022 ©2022 Association for Computational Linguistics

datasets we test, an output label receives nonzero probability under the kNN distribution only 44.2% of the time (see Section 6).

To address this challenge, we introduce kNN-Prompt, a simple and effective method built on kNN-LM for improving zero-shot inference with no further training. Key to our approach are *fuzzy verbalizers*, which automatically expand the set of tokens corresponding to each output label. For example, in Figure 1, the verbalized label of the negative sentiment is "terrible." Our fuzzy verbalizer also maps "silly" to negative sentiment, allowing the model to better leverage the information available in the kNN distribution. Extensive experiments (Section 3) show that applying kNN-Prompt using a purely unlabeled heterogeneous corpus consistently improves zero-shot performance on eleven tasks, including sentiment analysis, topic classification, entailment, fact retrieval and question answering. These improvements hold for every model in the GPT-2 family.

We also show that kNN-Prompt can be used to adapt LMs to new domains and tasks with no further training (Section 5). With a domain-specific datastore corpus, we achieve comparable or better performance to prompting the LM after domain-adaptive pretraining (Gururangan et al., 2020) on that corpus. To better understand these gains, we conduct a thorough analysis (Section 6), showing that fuzzy verbalizers are essential for leveraging the kNN distribution, the benefits of retrieval increase with retrieval model size, and even relatively small datastores can yield sizeable performance gains if they are tailored to the domain or task. Overall, our results show how retrieval can benefit zero-shot inference with LMs on a wide variety of tasks, and suggest that applying retrieval with larger models may yield even greater benefits. Code is available at `github.com/swj0419/kNN_prompt`.

2 Method

To perform zero-shot prediction on a downstream task using a pretrained language model, we recast the task as language modeling (Radford et al., 2019) by converting each input instance into a natural language prompt (Section 2.1). We then augment the pretrained model with the k-nearest-neighbors language modeling technique from Khandelwal et al. (2020). To better benefit from the sparse kNN distribution, we introduce *fuzzy verbalizers* for mapping from the LM's outputs to a distribution over task-specific labels (Section 2.3). Finally, we decode the output from this label distribution using the domain-conditional PMI scoring method of Holtzman et al. (2021).

2.1 Prompting and Verbalizers

We address classification problems where an instance consists of an input sequence of tokens $\mathbf{x} = (x_0, x_1, ..., x_{|\mathbf{x}|})$ from a vocabulary $\mathcal{V}$ and an output label $y \in Y$. The output label set Y may be fixed for the task (*text classification*). For example, in the sentiment analysis example in Figure 2, the input is $\mathbf{x}$ = "Mr. Tsai is one of world cinema's most gifted artists." The output labels are $Y = \{y^+, y^-\}$, referring to positive and negative sentiment.

To cast the task as language modeling, we deterministically transform each input example $\mathbf{x}$ into a **prompt** $p(\mathbf{x})$. Providing this prompt to an LM yields a probability distribution $P_{LM}(\mathbf{v} \mid p(\mathbf{x}))$. To extract an output label from this, we apply *verbalizers* $V : y \to \mathcal{V}^*$ (Schick and Schütze, 2021) which map each output label $y \in Y$ to a label word $V(y) = \mathbf{v}$. We can then compute a probability for each label:

$$P(y \mid \mathbf{x}) \propto P_{LM}(V(y) \mid p(\mathbf{x})), \tag{1}$$

normalizing over all $y \in Y$.

For example, our prompt transformation for sentiment analysis adds *It was* after the input, and uses the verbalizer $V(y^+)$ = *great*, $V(y^-)$ = *terrible*, which classifies sentiment according to the relative probabilities of *It was great* and *It was terrible* after the input sequence (see Figure 2, bottom left).

2.2 k-Nearest Neighbors Language Modeling

Following Khandelwal et al. (2020), we augment the LM with a *datastore* from which it can retrieve tokens that inform its predictions, improving performance without further training.

The datastore is a key-value store generated by running the LM over a corpus of text. Each value is a token $w \in \mathcal{V}$ from the corpus, and its key is the vector hidden representation at the output layer of the LM running forward on the left context $\mathbf{c} \in \mathcal{V}^*$ (call this $f(\mathbf{c})$). At inference time, when predicting the next token for an input sequence $\mathbf{c}$, the kNN-LM retrieves the k nearest neighbors of $\mathbf{c}$ from the datastore according to the distance $d(\cdot, f(\mathbf{c}))$ of their key vectors (squared L^2 distance following Khandelwal et al.).

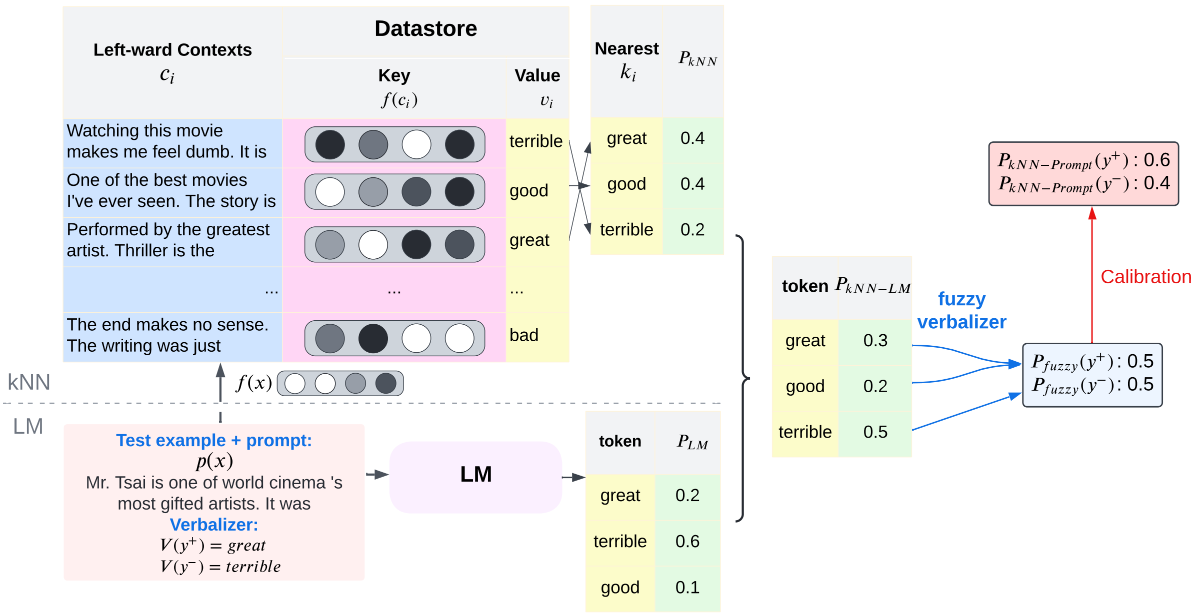

Figure 2: An illustration of kNN-Prompt applying to sentiment analysis tasks. Texts are encoded in the datastore, where each entry consists of a representation of a leftward context and its next token. During inference, a test example is mapped to a prompt form and used to retrieve the k most similar contexts and their next tokens from the datastore. The kNN distribution is a multinomial computed on the distance of the text example and similar contexts. The final prediction is formed by combining the kNN distribution with the language model output distribution.

A softmax over the (negative) distances induces a distribution over the the tokens w_i in the nearest neighbor set:

$$P_{kNN}(v \mid \mathbf{c}) \propto \sum_{(f(\mathbf{c}_i), w_i)} \mathbb{1}_{v=w_i} e^{\frac{-d(f(\mathbf{c}_i), f(\mathbf{c}))}{t}}$$

where t is a temperature parameter.[1] We can then interpolate this with the original LM as follows:

$$P_{kNN\text{-}LM}(v \mid c) = (1-\lambda)P_{LM}(v|c) + \lambda P_{kNN}(v|c).$$

The hyperparameters for the kNN-LM approach are the number k of nearest neighbors, the interpolation constant λ, the temperature t, and the choice of datastore.

2.3 Fuzzy verbalizers

One challenge in performing zero-shot inference with LMs on downstream tasks is the choice of verbalizer. On one hand, LMs may be highly sensitive to the particular surface form in ways that are irrelevant to the classification task (Holtzman et al., 2021). On the other hand, for a kNN model, the k nearest neighbor set is sparse and may fail to cover any of the tokens in the set of verbalizers (i.e., $P_{kNN}(V(y)) = 0$ for all $y \in Y$), limiting its utility in those cases. To address these issues, we introduce *fuzzy verbalizers*, which associate each label y with a neighborhood of token sequences around a specific verbalization $V(y) \in \mathcal{V}^*$.

To do this, we first associate each token $v \in \mathcal{V}$ with a neighborhood $\mathcal{N}(v) \subseteq \mathcal{V}$ of similar tokens. In particular, we use v's top-5 most similar words according to the cosine similarity of their GloVe embeddings (Pennington et al., 2014), as well as any of v's synonyms in ConceptNet (Speer et al., 2017).[2] Then, for the purposes of calculating the probability of a verbalized label $\mathbf{v} = V(y)$, we treat a prediction of any token in each neighborhood of v as a viable substitute for it, marginalizing over $\mathcal{N}(z)$:

$$P_{FV}(y \mid x) \propto \sum_{v_i \in \mathcal{N}(v)} P(v_i \mid p(\mathbf{x})) \qquad (2)$$

The fuzzy verbalizer helps mitigate the effect the sparsity of the kNN distribution has on zero-shot prediction (see Section 6).

[1] We have added the temperature adjustment in the softmax on top of the kNN-LM formulation.

[2] https://conceptnet.io

Corpus	Size	# Tokens
Wikitext-103	181MB	114M
Amazon Reviews	89MB	19M
CC-NEWS	457MB	324M
IMDB	45MB	8M
Total	722MB	465M

Table 1: Statistics of our heterogeneous datastore corpora.

2.4 Full model

To make a zero-shot prediction for an input $\mathbf{x}$, we first transform it into a prompt $p(\mathbf{x})$ and obtain a distribution over the label word $\mathbf{v}$ with a kNN-LM: $P_{kNN-LM}(\mathbf{v} \mid p(\mathbf{x}))$. We then apply *domain-conditional PMI* scoring rule (Holtzman et al., 2021) to calibrate the distribution:

$$\mathrm{PMI}_{DC}(v, p(\mathbf{x})) = \frac{P(v \mid p(\mathbf{x}))}{P(v \mid \mathbf{p})}$$

where $\mathbf{p}$ is a task-dependent string which is independent of the particular input (generally the local context at the end of the prompt, e.g., we use $\mathbf{p}$ = "It was" for sentiment analysis, as shown in Figure 2).

Finally, we convert this to the output label score $P(y \mid p(\mathbf{x}))$ using a fuzzy verbalizer (Section 2.3). When using the fuzzy verbalizer together with PMI calibration, instead of marginalizing over the tokens in the fuzzy verbalizer $v_i \in \mathcal{N}(v)$ (Equation 2), we score each label according to the sum of the PMIs of its associated tokens:

$$P(y \mid \mathbf{x}) \propto \sum_{v_i \in \mathcal{N}(v)} \mathrm{PMI}_{DC}(v_i, p(\mathbf{x}))$$

3 Experimental Setup

3.1 Tasks

We experiment with 9 tasks, including topic classification, sentiment analysis, entailment and partisanship classification.

Topic Classification We use the AG News (**AGN**) and Yahoo! Answers (**Yahoo**) corpora from Zhang et al. (2015) for topic classification.

Sentiment and Partisanship We study sentiment analysis using the Rotten Tomatoes (**RT**) and **SST-2** corpora of Socher et al. (2013), movie reviews from Pang and Lee (2005, **MR**), the customer review dataset from Hu and Liu (2004, **CR**) consisting of Amazon and Yelp reviews, and the hyperpartisan news detection dataset from Kiesel et al. (2019, **HYP**), which focuses on classifying whether a text exhibits extreme political views.

Entailment Entailment datasets focus on classifying whether one sentence plausibly implies another to be true or false. We evaluate on the CommitmentBank (De Marneffe et al., 2019, **CB**) and the Recognizing Textual Entailment (Dagan et al., 2010, **RTE**) dataset provided in GLUE (Wang et al., 2018).

3.2 kNN-Prompt Model Details

Inference Model For our main experiments, we directly use GPT-2 large from Huggingface[3] as our base LM. We consider other model sizes in Section 6.

Retriever Model Following the inference model, we use GPT-2 large to build the datastore. The keys are the 1280-dimensional hidden representations before the final MLP which predicts the token distribution at each timestep, produced using a single forward pass over the datastore corpus. For efficient similarity search, we create a FAISS (Johnson et al., 2019) index and search for nearest neighbors by Euclidean distance.

Datastore Corpus For our datastore, we aim to curate a large, heteregenous corpus of data broadly relevant to the tasks we evaluate. To this end, we combine four sources of data including Wikitext-103 (Merity et al., 2016), the Amazon review corpus of He and McAuley (2016), and subsets of CC-NEWS[4] and IMDB[5] sampled uniformly from each. Table 1 lists the specifics of each data source.

Inference Procedure We retrieve k=1024 neighbors, soften the kNN distribution with a temperature value of 3 and use an interpolation factor of $\lambda = 0.3$. Our primary evaluation is zero-shot. All hyperparameters were chosen on the basis of development experiments (see Section 6 for more details).

3.3 Baselines

LM is the result of prompting the base language model (GPT-2 Large), choosing the output label whose verbalizer has the highest probability under the language model $P_{LM}(V(y) \mid p(\mathbf{x}))$.

[3] https://github.com/huggingface/transformers
[4] https://huggingface.co/datasets/cc_news
[5] http://ai.stanford.edu/~amaas/data/sentiment

	RTE	CB	Yahoo	RT	SST-2	CR	MR	HYP	AGN	Avg
LM	53.1	48.2	49.7	53.0	55.3	66.2	54.6	58.5	67.4	56.2
LM+PMI	54.2	50.0	48.8	74.1	76.5	82.8	74.6	58.5	65.1	65.0
kNN-LM	53.1	48.2	49.5	54.5	55.4	67.2	56.4	58.5	67.0	56.6
kNN-Prompt	**55.6**	**53.5**	**51.0**	**80.6**	**84.2**	**84.3**	**78.2**	**60.0**	**78.8**	**69.6**

Table 2: Zero-shot results on a variety of tasks. Our model, kNN-Prompt, handily outperforms Holtzman et al. (2021)'s PMI scoring method alone (LM+PMI) as well as the base kNN-LM method of Khandelwal et al. (2020).

	CR	HYP	MR
LM	79.5 $_{4.1}$	56.7 $_{0.5}$	78.2 $_{1.4}$
LM+PMI	79.8 $_{5.5}$	52.7 $_{2.6}$	76.3 $_{1.5}$
kNN-LM	79.5 $_{4.2}$	56.7 $_{1.5}$	77.5 $_{2.3}$
kNN-prompt	**80.5** $_{\mathbf{1.7}}$	**57.1** $_{\mathbf{1.1}}$	**79.4** $_{\mathbf{1.5}}$

Table 3: The mean and standard deviation for 4 uniformly sampled sets of 4 demonstration examples used for few-shot inference.

	CR	HYP	MR
LM + PMI	82.8	58.5	74.6
kNN-prompt	**84.3**	60.0	**78.2**
DAPT (LM + PMI)	84.1	**61.1**	77.8

Table 4: Domain adaptation experiments using domain-specific datastores. DAPT requires training the LM on the corresponding datastore, while kNN-Prompt can use it as the datastore with no further training.

LM+PMI is the approach of Holtzman et al. (2021), calibrating **LM** with domain-conditional PMI scoring (Section 2.4).

k**NN-LM** directly applies the kNN-LM of Khandelwal et al. (2020) in the same way as **LM**, choosing the highest-probability output label.

4 Experimental Results

Results for zero-shot prediction are in Table 2. kNN-Prompt outperforms all baselines in all tasks, improving over the base LM by 13.4% on average. The gains are particularly pronounced for MR and RT (sentiment analysis on movie reviews), Yahoo (topic classification). For MR and RT, the gains seem to come mostly from PMI calibration.

Interestingly, the kNN-LM alone yields a fairly small improvement over the base LM (about 0.4% on average). This suggests that the fuzzy verbalizer and PMI calibration methods may help kNN-Prompt better leverage the information in the k-nearest neighbors distribution. We examine possible sources of kNN-Prompt's performance gains in Section 6.

Few-shot inference For a subset of tasks, we additionally compare to a few-shot setting where we prepend four examples uniformly sampled from the training data to the input (Table 3). The demonstration examples are converted to prompt and verbalizer format. We report the mean accuracy and standard deviation with 4 different random seeds. We find that kNN-Prompt consistently outperform baselines, demonstrating that kNN-Prompt is applicable to the few-shot setting as well. We leave further exploration of this phenomenon to future work.

5 kNN-Prompt for Domain Adaptation

One of the advantages of retrieval-based LMs is that they can be adapted to new domains with no further training.

To test this capability, we replace our heterogeneous datastore (Section 3.2) with domain-specific ones for several tasks. To build these domain-specific datastores, we select Amazon Reviews for CR, CC-NEWS for HYP and IMDB for MR, and encode them separately.

For comparison, we consider domain-adaptive pretraining (Gururangan et al., 2020, DAPT), which further trains the LM on the domain-specific corpus. We train GPT-2 Large on each domain corpus for a single pass, then apply it to downstream tasks using our prompting and verbalizer setup and domain-conditional PMI scoring.

As shown in Table 4, kNN-Prompt performs comparably with DAPT. Specifically, kNN-Prompt slightly outperforms DAPT on CR and MR. These results indicate that kNN-Prompt is an effective method for domain adaptation. Critically, unlike DAPT, kNN-Prompt does not require further training, which is more practical and efficient for adapting very large LMs.

Effect of datastore distribution and size To better understand kNN-Prompt's potential for domain adaptation, we experiment with varying sizes and

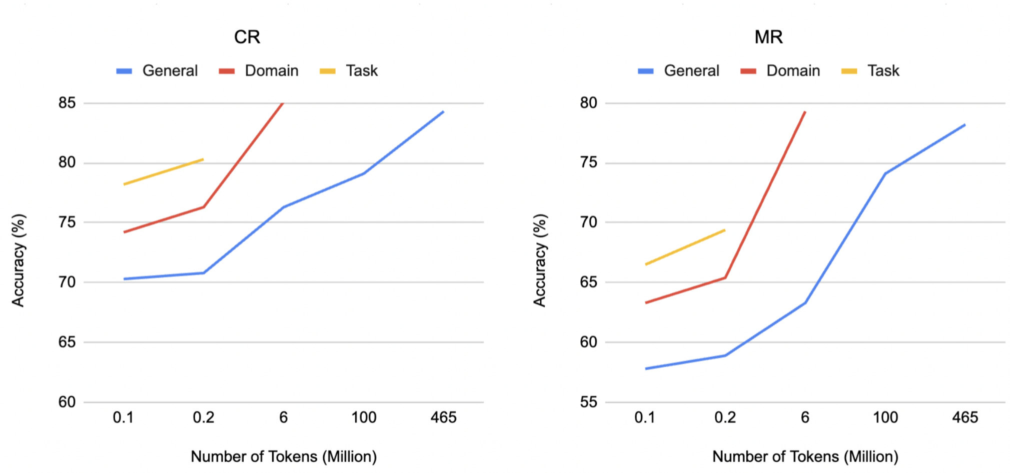

Figure 3: Effect of the number of tokens in the datastore on CR and MR. Each line represents the kNN-Prompt model with a different datastore and the line ends when the entire available datastore is used. General, Domain, and Task refer to the heterogeneous corpus (Table 1), domain-specific corpus, and task-specific corpus, respectively. We use IMDB as the domain-specific corpus for MR, and Amazon Reviews for CR. The task-specific corpus is the unlabeled training data of each task. GPT-2 Large is used for both retriever and inference models.

distributions of the datastore. For each task, we consider three options for the datastore corpus: our heterogeneous corpus (Section 3.2), a domain-specific corpus, and a task-specific corpus constructed from the task's (unlabeled) training data. Each of these data sources exhibits increasing levels of relevance to the task.

Figure 3 shows how model performance varies with the choice of datastore across different datastore sizes. For a fixed number of tokens, retrieving from a task-specific datastore is best. Furthermore, token-for-token, adding task-specific data leads to more gains than domain-specific data, which in turn is better than our heterogeneous corpus.

Using domain-specific data is always better than retrieving from the large heterogeneous corpus. For example, for CR, using 6M tokens of domain-specific data outperforms using our 465M token heterogeneous corpus. These results suggest that while having a large datastore is beneficial, curating task-specific or domain-specific data can also be an effective way of improving model performance, especially if datastore size is limited (e.g., due to memory constraints).

6 Analysis

We perform several experiments to understand the contribution of each component of kNN-Prompt and inform our choice of hyperparameters.

Model ablations kNN-Prompt incorporates three features on top of the base LM: kNN retrieval and interpolation (Section 2.2), fuzzy verbalizers (Section 2.3), and PMI scoring (Section 2.4). Table 5 shows the results of ablation experiments analyzing the contribution of each component.

Model	Acc.	ΔAcc.
LM	56.2	0
LM+kNN (kNN-LM)	56.6	+0.4
LM+Fuzzy	63.4	+7.2
LM+PMI	65.0	+8.8
LM+Fuzzy+PMI	67.1	+10.9
LM+kNN+Fuzzy	66.5	+10.3
LM+kNN+PMI	64.2	+8.0
LM+kNN+Fuzzy+PMI (kNN-Prompt)	**69.6**	**+13.4**

Table 5: Effect of different components on the average zero-shot accuracy across the eleven tasks.

First, we find that adding kNN to LM gives trivial improvement (+0.4%), but much greater once we add fuzzy verbalizers on top of them (+10.3%), exceeding the contribution of the two components independently (with fuzzy verbalizers alone at +7.2%). This supports the argument that fuzzy verbalizers allow the model to make better use of the sparse support of the kNN distribution. Indeed, we find that across all tasks, an output label receives nonzero probability under the kNN distribution in kNN-LM only 45.8% of the time. With fuzzy verbalizers, this increases to 78%.

Second, we find that for the base LM, fuzzy verbalizers bring gains (+7.2%) similar to PMI scoring (+8.8%), but the gains are only partially addi-

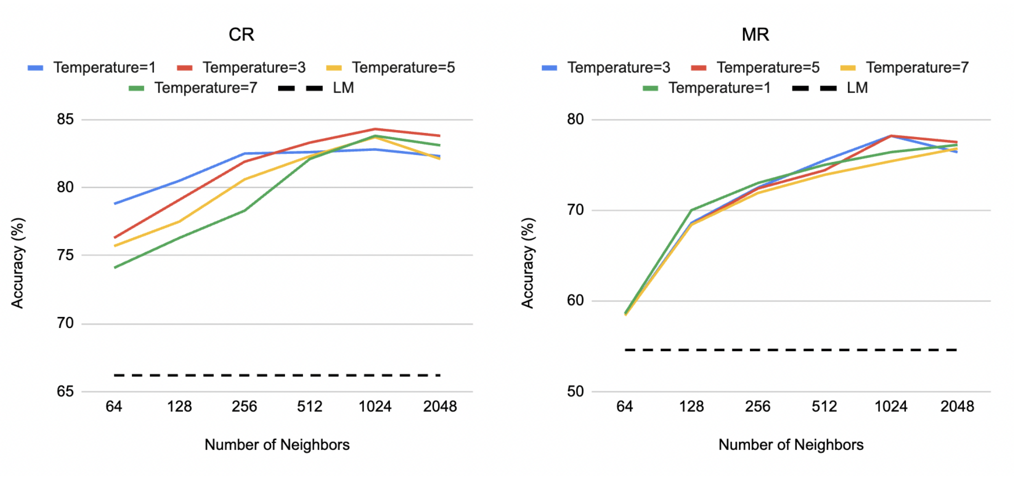

Figure 4: Effect of the number of retrieved neighbors and softmax temperature on kNN-Prompt performance for three tasks: CR and MR. Task performance monotonically improves with the number of neighbors as k is increased to 1024.

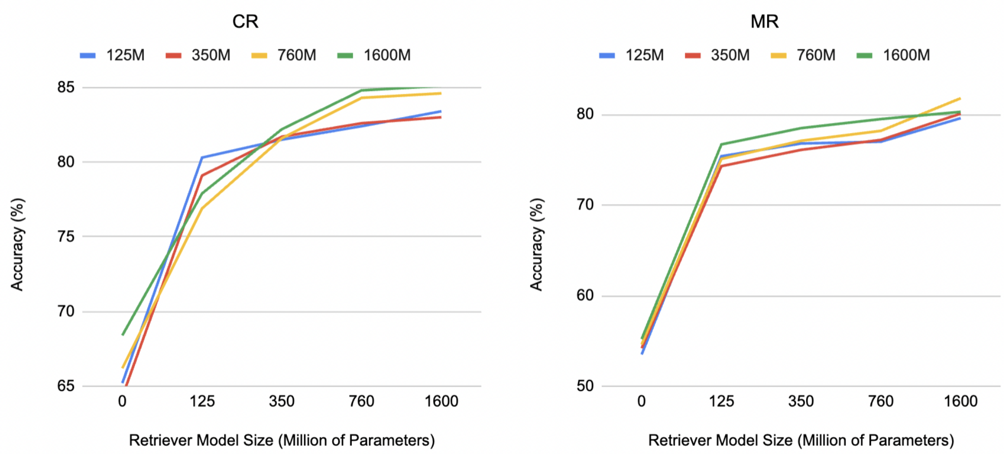

Figure 5: Effect of the retriever model size (GPT-2) on CR and MR. A size of 0 indicates that no retriever is used. Different lines represent different-sized inference models (GPT-2). The benefits of kNN-Prompt scale with the retriever model size.

tive when combining the two techniques (+10.9%). This suggests that by incorporating more varied surface forms into the score for each label, fuzzy verbalizers may partially — but not completely — mitigate the surface form competition problem which PMI scoring was designed to tackle (Holtzman et al., 2021). The effect of PMI scoring is increased, however, when we use fuzzy verbalizers and kNN retrieval together (+13.4% for the full model versus +10.3% for kNN with fuzzy verbalizers), suggesting that the kNN distribution might suffer from more surface form competition problems than the base LM distribution.

kNN retrieval hyperparameters Figure 4 shows how the number of retrieved nearest neighbors (k) and softmax temperature affect model performance on three datasets. In most cases, performance monotonically improves with the number of neighbors when k is smaller than 1024 and deteriorates after that. When k < 256, a temperature of 1 performs best, while flattening the distribution is useful when retrieving more neighbors. Overall, using 1024 neighbors and a temperature value of 3 performs consistently well across the tasks we test.

Retrieval model size and inference model size Figure 5 shows how performance varies with the size of the retriever and inference models on three tasks. We observe substantial gains as the size of the retriever increases, which hold regardless of inference model size.

It should be noted that a larger retriever leads to a larger datastore and slower retrieval: increasing the retriever size from 125M to 1600M parameters doubles the memory footprint of the datastore, which scales with the size of the retriever model's output embeddings. These computational tradeoffs may inform the retriever size best suited for a particular application.

7 Related Work

Retrieval-augmented LMs Several studies propose the use of retrieval mechanisms with external datastores to improve language modeling performance (Khandelwal et al., 2020) and open-domain question answering (Izacard and Grave, 2020; Lewis et al., 2020). Other work incorporates retrieval directly into the LM pretraining process (Guu et al., 2020; Borgeaud et al., 2021). Khandelwal et al. (2021) applies nearest neighbor retrieval to conditional sequence generation to improve the quality of machine translation systems. Our work is the first to show that retrieval augmentation, introduced at test time, improves the zero-shot inference of language models on a variety of end tasks.

Zero-shot and few-shot inference Brown et al. (2020b) demonstrate that large LMs can perform zero-shot (given only a prompt) and few-shot learning (using a concatenation of training examples as demonstrations) without any finetuning. Subsequent work further improves their zero-shot and few-shot abilities with calibration (Holtzman et al., 2021; Zhao et al., 2021; Min et al., 2021a), prompt engineering (Lu et al., 2021; Shin et al., 2020) and meta-tuning (Min et al., 2021b; Wei et al., 2022; Zhong et al., 2021). Rubin et al. (2021) and Liu et al. (2021) apply retrieval methods to select in-context learning examples that are semantically-similar to a test example for few-shot inference. However, these retrieval methods require access to a large set of labeled data. In contrast, kNN-Prompt only assumes the availability of a heterogeneous unlabeled corpus.

8 Conclusions

We present kNN-Prompt, a new technique to augment LMs with nearest neighbor retrieval for zero-shot inference on end tasks. kNN-Prompt substantially improves zero-shot performance on a wide range of multiple-choice and classification tasks. With a domain- or task-relevant datastore, kNN-Prompt enables efficient domain adaptation with no additional training, and its benefits scale with the size of the retrieval model.

9 Limitations

Although kNN-Prompt significantly improves GPT-2 family models' zero-shot and few-shot performance, it stores high-dimensional vectors for every token in the datastore corpus and performs k-nearest neighbor search for every next token, which incurs significant inference overhead. Future work may study compressing the datastore and approximating kNN-search for efficient retrieval. Careful analysis could also explore datastore curation methods to balance task-relevancy, domain generality, and size. In addition, compared with sentence or document-level retrieval, retrieving tokens at each time step may limit the language model's ability to reason about the retrieved information. Future work may explore if more coarse-grained retrieval and interpolation such as chunks, sentences and documents-level lead to better end task performance.

Our evaluation of kNN-Prompt is limited to GPT-2 family models and eleven end tasks. There are many other tasks and language models for which kNN-Prompt can be useful. Future work may study the usefulness of kNN-Prompt with larger inference models (i.e: GPT-3) and more diverse tasks. Potentially, large inference models combined with larger retrieval models may result in better zero-shot performance.

References

Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, et al. 2021. Improving language models by retrieving from trillions of tokens. *arXiv preprint arXiv:2112.04426*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020a. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario

Amodei. 2020b. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Ido Dagan, Bill Dolan, Bernardo Magnini, and Dan Roth. 2010. Recognizing textual entailment: Rational, evaluation and approaches – erratum. *Natural Language Engineering*, 16(1):105–105.

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.

Suchin Gururangan, Ana Marasović, Swabha Swayamdipta, Kyle Lo, Iz Beltagy, Doug Downey, and Noah A. Smith. 2020. Don't stop pretraining: Adapt language models to domains and tasks. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 8342–8360, Online. Association for Computational Linguistics.

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.

Junxian He, Graham Neubig, and Taylor Berg-Kirkpatrick. 2021. Efficient nearest neighbor language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5703–5714.

Ruining He and Julian McAuley. 2016. Ups and downs: Modeling the visual evolution of fashion trends with one-class collaborative filtering. In *proceedings of the 25th international conference on world wide web*, pages 507–517.

Ari Holtzman, Peter West, Vered Shwartz, Yejin Choi, and Luke Zettlemoyer. 2021. Surface form competition: Why the highest probability answer isn't always right. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 7038–7051, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Minqing Hu and Bing Liu. 2004. Mining and summarizing customer reviews. In *Proceedings of the Tenth ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '04, page 168–177, New York, NY, USA. Association for Computing Machinery.

Gautier Izacard and Edouard Grave. 2020. Leveraging passage retrieval with generative models for open domain question answering.

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

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

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Generalization through memorization: Nearest neighbor language models. In *International Conference on Learning Representations*.

Johannes Kiesel, Maria Mestre, Rishabh Shukla, Emmanuel Vincent, Payam Adineh, David Corney, Benno Stein, and Martin Potthast. 2019. SemEval-2019 task 4: Hyperpartisan news detection. In *Proceedings of the 13th International Workshop on Semantic Evaluation*, pages 829–839, Minneapolis, Minnesota, USA. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2021. What makes good in-context examples for gpt-3? *arXiv preprint arXiv:2101.06804*.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2021. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. *arXiv preprint arXiv:2104.08786*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *arXiv preprint arXiv:1609.07843*.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2021a. Noisy channel language model prompting for few-shot text classification. *arXiv preprint*.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2021b. Metaicl: Learning to learn in context. *arXiv preprint arXiv:2110.15943*.

Bo Pang and Lillian Lee. 2005. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting of the Association for Computational Linguistics (ACL'05)*, pages 115–124, Ann Arbor, Michigan. Association for Computational Linguistics.

Jeffrey Pennington, Richard Socher, and Christopher Manning. 2014. GloVe: Global vectors for word representation. In *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 1532–1543, Doha, Qatar. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2021. Learning to retrieve prompts for in-context learning. *arXiv preprint arXiv:2112.08633*.

Timo Schick and Hinrich Schütze. 2021. Exploiting cloze-questions for few-shot text classification and natural language inference. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pages 255–269, Online. Association for Computational Linguistics.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. AutoPrompt: Eliciting Knowledge from Language Models with Automatically Generated Prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235, Online. Association for Computational Linguistics.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D Manning, Andrew Y Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1631–1642.

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*.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2018. GLUE: A multi-task benchmark and analysis platform for natural language understanding. In *Proceedings of the 2018 EMNLP Workshop BlackboxNLP: Analyzing and Interpreting Neural Networks for NLP*, pages 353–355, Brussels, Belgium. Association for Computational Linguistics.

Jason Wei, Maarten Bosma, Vincent Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V Le. 2022. Finetuned language models are zero-shot learners. In *International Conference on Learning Representations*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Tony Z Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models.

Ruiqi Zhong, Kristy Lee, Zheng Zhang, and Dan Klein. 2021. Adapting language models for zero-shot learning by meta-tuning on dataset and prompt collections. In *Findings of the Association for Computational Linguistics: EMNLP 2021*, pages 2856–2878, Punta Cana, Dominican Republic. Association for Computational Linguistics.

Test Example	Label	LM Prediction		
Too few games could set back PSP launch - Sony exec Signs of a delay, or just managing expectations? The text topic is about	technology	sports		
Retrieved Context	**Retrieved Value**	**kNN Prediction**	**Distance**	**Corpus**
...References to the game are commonly brought up in other articles about...	software	technology	33.5	Wikitext-103
...While it would be easy to point an accusatory finger at Sony and blame them for killing the Dreamcast by overselling the PS2... there's a certain level of intellectual dishonesty in such a stance... [Sega]'s poor U.S. support for ...	hardware	technology	33.7	Wikitext-103

Table 6: An example from AGN where the kNN gives the correct prediction while LM does not.

A Appendix

A.1 Case Study

We manually check examples where kNN-Prompt is better than LM to understand why kNN-Prompt improves performance. As shown in Table 6, the language model has to know the meaning of the entity "PSP" and "Sony", otherwise it may associate "games" with a sport. kNN is able to match "Sony" in one of the retrieved neighbors, resolving the ambiguity of the word "games".

A.2 Templates

Table 7 shows the template and verbalizer used for each dataset.

Dataset	Template + input	Verbalizer (Fuzzy verbalizer)
RTE	Time Warner is the world's largest media and Internet company. question: Time Warner is the world's largest company. true or false? answer:	**true** (true, yes, correct, faithful, accurate...) **false** (false, no, incorrect, wrong, untrue, unfaithful...)
CB	question: Given that What fun to hear Artemis laugh. She's such a serious child. Is I didn't know she had a sense of humor. true, false, or neither? Answer:	**true** (true, yes, correct, faithful, accurate...) **false** (false, no, incorrect, wrong, untrue, unfaithful...) **neither** (neither, none, nothing)
Yahoo	why doesn't an optical mouse work on a glass table? topic:	**society** (society, culture, sociality, group, tribal, organization...) **science** (science, math, scientist, knowledge, physics, bioscience...) **health** (health, disease, obesity, medicine, nutrition, well-being...) **education** (education, pedagogy, instruction, school, curriculum, college...) **computer** (computer, internet, network, laptop, progammer, hardware...) **sports** (sports, athletics, sportsman, play, football, basketball...) **business** (business, finance, economics, fund, banking, investment...) **entertainment** (entertainment, music, amusement, game, recreation...) **family** (family, relationships, marriage, household, friendship...) **politics** (politics, government, geopolitics, law, democracy, politician...)
AGN	Economic growth in Japan slows down as the country experiences. topic:	**politics** (politics, government, geopolitics, law, democracy, politician...) **sports** (sports, athletics, sportsman, play, football, basketball...) **business** (business, finance, economics, fund, banking, investment...) **technology** (technology, engineering, science, techinal, science, computer...)
HYP	Are you sick of Republicans? Or just right-wingers in general? ... neutral or partisan? Answer:	**neutral** (neutral, fair, objective, impartial, disinterested...) **partisan** (partisan, biased, unfair, prejudiced, unjust...)
SST-2, CR, MR, RT	Illuminating if overly talky documentary. It was	**great** (great, good, gorgeous, legendary, perfect, phenomenal...) terrible (terrible, plain, poor, hideous, upset, awful...)

Table 7: The template and the example (colored black) used for each dataset. We also include the **standard verbalizer** and (a sample of tokens used in the fuzzy verbalizer).