

What Makes Good In-Context Examples for GPT-3?

Jiachang Liu^{1*}, Dinghan Shen², Yizhe Zhang³, Bill Dolan³, Lawrence Carin¹, Weizhu Chen²

¹Duke University

²Microsoft Dynamics 365 AI

³Microsoft Research

¹{jiachang.liu, lcarin}@duke.edu

^{2,3}{dishen, yizzhang, billdol, wzchen}@microsoft.com

Abstract

GPT-3 (Brown et al., 2020) has attracted lots of attention due to its superior performance across a wide range of NLP tasks, especially with its powerful and versatile in-context few-shot learning ability. Despite its success, we found that the empirical results of GPT-3 depend heavily on the choice of in-context examples. In this work, we investigate whether there are more effective strategies for judiciously selecting in-context examples (relative to random sampling) that better leverage GPT-3’s few-shot capabilities. Inspired by the recent success of leveraging a retrieval module to augment large-scale neural network models, we propose to retrieve examples that are semantically-similar to a test sample to formulate its corresponding prompt. Intuitively, the in-context examples selected with such a strategy may serve as more informative inputs to unleash GPT-3’s extensive knowledge. We evaluate the proposed approach on several natural language understanding and generation benchmarks, where the retrieval-based prompt selection approach consistently outperforms the random baseline. Moreover, it is observed that the sentence encoders finetuned on task-related datasets yield even more helpful retrieval results. Notably, significant gains are observed on tasks such as table-to-text generation (41.9% on the ToTTo dataset) and open-domain question answering (45.5% on the NQ dataset). We hope our investigation could help understand the behaviors of GPT-3 and large-scale pre-trained LMs in general and enhance their few-shot capabilities.

1 Introduction

GPT-3 (Brown et al., 2020) is a new breakthrough in NLP research. Previously, NLP models are pre-trained on large quantities of data and fine-tuned

*Work was done during an internship at Microsoft Dynamics 365 AI.

Trial	1	2	3	4	5
Accuracy	94.6	95.0	95.8	93.9	86.9

Table 1: Results of GPT-3 on the task of sentiment analysis on the SST-2 dataset. Five different in-context examples are randomly selected from the training set. We observe different contexts induce different accuracies on the test set.

on a specific task and dataset. What sets GPT-3 apart from other pre-trained language models is its impressive “in-context” few-shot learning ability. Provided with a few in-context examples, GPT-3 is able to generalize to unseen cases without further fine-tuning. This opens up many new technological possibilities that are previously considered unique to human. For example, NLP systems can be developed to expand emails, extract entities from text, generate code based on natural language instructions with a few demonstration examples.

Despite its powerful and versatile in-context learning ability, GPT-3 has some practical challenges/ambiguities. The original paper (Brown et al., 2020) utilizes task-relevant examples that are randomly sampled from the training set to construct the context. In practice, we observe that the performance of GPT-3 tends to fluctuate with different choices of in-context examples. As shown in Table 1, the variance of the empirical results with distinct in-context examples can be significant. The results are highly sensitive to the examples. Our work aims to carefully examine this issue to gain a deeper understanding on how to better select in-context examples to unleash GPT-3’s few-shot capabilities and further improve its performance.

A brute-force approach would be to perform combinatorial search over the entire dataset. Unfortunately, this strategy is computationally expensive and thus impractical in many cases. To

this end, we investigate the influences of employing different in-context examples on the empirical results. Interestingly, we found that the in-context examples that are closer to the test sample in the embedding space consistently give rise to stronger performance (relative to the farther ones). Inspired by this observation and the recent success of retrieval-augmented models (Hashimoto et al., 2018), we propose to utilize nearest neighbors of a given test sample (among all the training instances available) as the corresponding in-context examples. The retrieved examples, along with the test sample, are provided to GPT-3 for the final prediction.

To verify the effectiveness of the proposed method, we evaluate it on several natural language understanding and generation tasks, including sentiment analysis, table-to-text generation and open-domain question answering. It is observed that the retrieval-based in-context examples unleash the few-shot capabilities of GPT-3 much more effectively than a random sampling baseline. Even with a smaller number of in-context examples, the proposed strategy empowers GPT-3 to achieve stronger performance. Moreover, we find that the specific sentence encoders employed for the retrieval procedure play a critical role. Thus, an extensive exploration regarding different pre-trained encoders is conducted, and it is shown that encoders fine-tuned on natural language matching tasks serve as more effective in-context examples selector on the QA task. Detailed analysis and case study further validate the effectiveness of proposed methods. In summary, our contributions in this paper are as follows:

i) to the best of our knowledge, we take a first step towards understanding the sensitivity of GPT-3’s few-shot capabilities with respect to the selection of in-context examples;

ii) to alleviate the sensitivity issue, an additional retrieval module is introduced to find semantically-similar in-context examples of a test instance to construct its corresponding input, which greatly outperforms the baseline based on random sampled examples;

iii) fine-tuning the retrieval model on task-related dataset(s) leads to even stronger empirical results with GPT-3;

iv) the performance of GPT-3 improves as the number of examples available for retrieval increases.

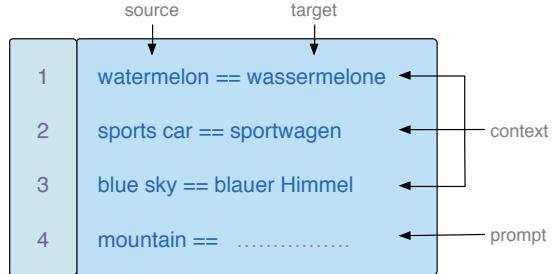


Figure 1: The figure above shows how to perform in-context learning with a language model. Three in-context examples and the test prompt are concatenated as a single string input for GPT-3, with a special character ”\n” inserted between two adjacent examples. GPT-3 keeps generating tokens until there is a special character ”\n”.

2 Method

2.1 GPT-3 for In-Context Learning

The in-context learning scenario of GPT-3 can be regarded as a conditional text generation problem. Concretely, the probability of generating a target y is conditioned on the context C , which includes k examples, and the source x . Therefore, the prediction y corresponding to the source x can be expressed as:

$$p_{LM}(y|C, x) = \prod_{t=1}^T p(y_t|C, x, y_{<t}) \quad (1)$$

where LM denotes the parameters of the language model, and $C = \{x_1, y_1, x_2, y_2, \dots, x_k, y_k\}$ is a context string. In GPT-3, the C is created by concatenating k training instances along with their corresponding labels. As shown in the illustration of Figure 1, GPT-3 is asked to translate “mountain” to its German version based on the three examples given as part of the input.

For GPT-3, this generation process is implemented through a giant transformer-based model architecture (Vaswani et al., 2017; Brown et al., 2020). Given the large size of the GPT-3 model, it would be very computationally-involved to fine-tune it on task-specific samples. Thus, GPT-3 is typically leveraged in a in-context learning manner as described above. It has been shown that GPT-3 has powerful few-shot capabilities, where it can perform quite well with only a small number of demonstrations provided. Unfortunately, as shown in Table 1, the results of GPT tends to fluctuate significantly with different in-context examples chosen. Here we aim to alleviate this issue via judicious in-context examples selection.

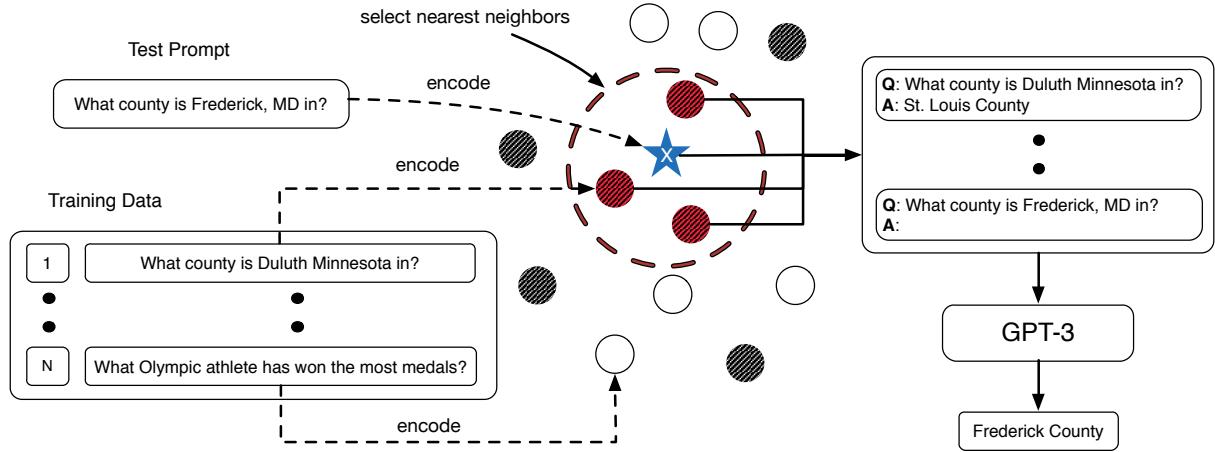


Figure 2: In-context example selection for GPT-3. White dots: unused training samples; grey dots: randomly sampled training samples; red dots: training samples selected by the k -nearest neighbors algorithm in the embedding space of a sentence encoder.

2.2 The Impact of In-Context Examples

Given the observation that the empirical results of GPT-3 are sensitive to the chosen in-context examples, we look at the role of in-context examples from an empirical perspective. Previous retrieve-and-edit literatures usually retrieve prototypes that are close to the test source x in some embedding space. These examples and the test source x often share semantic or lexical similarities. This hints on how we may select in-context examples for GPT-3.

To this end, we examine the impact of the distance between in-context example and the test sample on GPT-3’s performance. Concretely, a comparison is made on the the Natural Questions (NQ) dataset between two in-context example selection strategies. Given each test example, the first method utilizes the 10 farthest training instances to construct the context provided to GPT-3, while the second employs the 10 closest neighbors. We use the CLS embeddings of a pre-trained RoBERTa-large model as the sentence representations to measure the proximity of two sentences (using the Euclidean distance).

For evaluation, 100 test questions are randomly sampled and the average Exact Match (EM) scores with the two distinct strategies are reported in Ta-

ble 2. It can be observed that the nearest neighbors, as the in-context examples, give rise to much better results relative to the farthest ones. Moreover, the pre-trained RoBERTa model serves as effective sentence embeddings for the retrieval procedure.

2.3 k NN-augmented In-Context Example Selection

Based on the findings above, we propose KATE¹, a strategy to select good in-context examples for in-context learning. The process is visualized in Figure 2. Specifically, we first use a certain sentence encoder to convert sources in both the training set and test set to vector representations. For online prediction, we can convert the training set first and encode each test source on the fly. Then, for each test source x , we retrieve its nearest k neighbors x_1, x_2, \dots, x_k from the training set (according to the distances in the sentence encoder’s embedding space). Given some predefined similarity measure d such as the cosine similarity, the neighbors are ordered in such a way that $d(x_i, x) \leq d(x_j, x)$ when $i < j$.

Afterwards, the k sources are concatenated with their corresponding targets to form the context $C = \{x_1, y_1, x_2, y_2, \dots, x_k, y_k\}$, which is further sent to GPT-3 along with the test input. The algorithm chart is presented in Algorithm 1. Note that different numbers of in-context examples can be employed here, and we conduct ablation study on its impact in a later section.

Method	Closest	Farthest
Accuracy	46.0	31.0

Table 2: Comparison of the EM score on the closest 10 neighbors and farthest 10 neighbors on a subset of 100 test samples of the NQ dataset.

¹KATE: Knn-Augmented in-conText Example selection

Algorithm 1 k NN In-context Example Selection			
Given: test prompt \mathbf{x}_{test} , training set $\mathcal{D}_T = \{\mathbf{x}_i, \mathbf{y}_i\}_{i=1}^N$, sentence encoder $\mu_\theta(\cdot)$, and number of in-context examples k (hyperparameter).			
1: $\mathbf{v}_{\text{test}} = \mu_\theta(\mathbf{x}_{\text{test}})$			
2: for $\mathbf{x}_i \in \mathcal{D}_T$ do			
3: $\mathbf{v}_i = \mu_\theta(\mathbf{x}_i)$			
4: $s_i = -\ \mathbf{v}_{\text{test}} - \mathbf{v}_i\ _2$ (or $\frac{\mathbf{v}_{\text{test}} \cdot \mathbf{v}_i}{\ \mathbf{v}_{\text{test}}\ _2 \ \mathbf{v}_i\ _2}$)			
5: end for			
6: Select largest k similarities s_i ’s (in descending order) with indices $\{\sigma(1), \dots, \sigma(k)\}$			
7: $C = [\mathbf{x}_{\sigma(1)}; \mathbf{y}_{\sigma(1)}; \dots; \mathbf{x}_{\sigma(k)}; \mathbf{y}_{\sigma(k)}]$			
8: $\hat{\mathbf{y}}_{\text{test}} = \text{GPT-3}([C; \mathbf{x}_{\text{test}}])$			

Choices of Retrieval Module A core step for our context selection approach is mapping sentences into a latent semantic space, leaving a question as what sentence encoders we should choose. We compared among existing pre-trained text encoders and found them sufficient to retrieve semantically similar sentences. The sentence encoders can be divided into two categories.

The first category includes most generally pre-trained sentence encoders such as a pre-trained BERT, RoBERTa, or XLNet models. These models have been trained on large quantities of unsupervised tasks and achieved good performance on many natural language tasks. The corresponding embeddings contain rich semantic information from the original sentences.

The second category includes sentence encoders fine-tuned on specific tasks or datasets. For example, a sentence encoder trained on the STS benchmark dataset should be able to assess similarities among different questions better than a generally pre-trained sentence encoder. (Reimers and Gurevych, 2019, 2020) have shown that these fine-tuned encoders have achieved great performance on tasks such as sentence clustering, paraphrase mining, and information retrieval.

3 Experimental Setup

We apply the k NN in-context selection method to the following three tasks: sentiment classification, table-to-text generation, and question answering (QA). Datasets and the common data split setups are shown in Table 3. In terms of the hyperparameters in the GPT-3 API, we set the temperature to 0. We let GPT-3 keep generating tokens until there is a special token “\n”.

Dataset	Train	Dev	Test
SST-2	67k	872	1.8k
IMDB	25k	-	25k
ToTTo	120k	7.7k	7.7k
NQ	79k	8.8k	3.6k
WQ	3.4k	361	2k
TriviaQA	78.8k	8.8k	11.3k

Table 3: Data split for different datasets. In-context examples are selected from the training set. Because ToTTo and TriviaQA require submitting to their leaderboards, the evaluation is done on the dev sets. For all other datasets, the evaluation is done on the test sets.

3.1 Sentence Embeddings for Retrieval

To retrieve semantically-similar training instances, we consider two types of sentence embeddings.

- The original pre-trained RoBERTa-large model (Liu et al., 2019), which is abbreviated as KATE_{roberta};
- The RoBERTa-large model fine-tuned on task-related datasets: *i*) fine-tuned on the SNLI and MultiNLI dataset (KATE_{nli}); *ii*) first fine-tuned on the SNLI and MultiNLI dataset and then on the STS-B dataset (KATE_{nli+sts-b}).

Notably, all the sentence encoders share the same the architecture, where the only differences are the specific datasets used for fine-tuning. Euclidean distance is used for the KATE_{roberta} case, while cosine similarity is employed for KATE_{nli} and KATE_{nli+sts-b}.

Sentiment Analysis For sentiment classification, we select in-context examples under the transfer setting, where one dataset is treated as the training set and the evaluation is made on another dataset. This transfer setting is designed to simulate a real-world scenario where we would like to leverage an existing labeled dataset for a unlabeled one (of a similar task).

Specifically, we select in-context examples from the SST-2 training set (Socher et al., 2013; Wang et al., 2018) and ask GPT-3 to make predictions on the IMDB test set (Maas et al., 2011). To explore whether a sentence encoder fine-tuned on a similar task would benefit KATE’s performance, we also employ a pre-trained RoBERTa-large model fine-tuned on the SST-2 training set (dubbed as KATE_{sst-2}). The performance is measured by the accuracy over the entire IMDB test

set. The number of in-context examples is chosen to be 3 since adding more examples does not further improve the performance.

Table-to-Text Generation Given a Wikipedia table and a set of highlighted cells, this task focuses on producing human-readable texts as descriptions. ToTTo (Parikh et al., 2020) is utilized for evaluation due to its popularity. We use BLEU (Papineni et al., 2002) and PARENT (Dhingra et al., 2019). metrics for evaluation. The ToTTo code base contains both evaluation and preprocessing scripts². Due to the input length limit of GPT-3 (currently the token limit is 2048), we add an extra preprocessing step by deleting the closing angle brackets such as `</cell>` and `</table>` to save some space. The number of in-context examples is set as 2 .

Question Answering Given a factual question, the model is asked to generate the correct answer. Following prior studies, we use the Exact Match (EM) score to measure the performance of GPT-3 on open-domain QA tasks. The EM score is defined as the proportion of the number of predicted answers being exactly the same as (one of) the ground-truth answer(s). The matching is performed after string normalization, which includes article and punctuation removal. We conduct experiments on three open-domain QA benchmarks: Natural Questions (NQ) (Kwiatkowski et al., 2019), Web Questions (WQ) (Berant et al., 2013), and Trivia Question Answering (TriviaQA) (Joshi et al., 2017). For this task, we pick the nearest 64 neighbors as the in-context examples for NQ and WQ and nearest 10 neighbors for TriviaQA (The retrieved 64 examples could not fit into 2048 token limit for TriviaQA. For fair comparison, we set the number of in-context examples to be 10 for TriviaQA for both the baseline and KATE method). The evaluation is done on the test sets of NQ and WQ and the dev set of TriviaQA.

3.2 Baseline Methods

Random Sampling For each test sentence, we randomly select in-context examples from the training set. We refer to this method as *Random* in the experimental results. To have a fair comparison with KATE, the number of in-context examples in this random baseline is the same as KATE

²The ToTTo code base can be found at <https://github.com/google-research/language/tree/master/language/totto>

Method	Accuracy
Random	87.95 ± 2.74
$k\text{NN}_{\text{roberta}}$	50.20
KATE _{roberta}	91.99
KATE _{nli}	90.40
KATE _{nli+sts-b}	90.20
KATE _{sst-2}	93.43

Table 4: Accuracy of sentiment prediction for GPT-3 on IMDB with different choices of in-context examples. In-context examples are from the SST-2 dataset.

to ensure fair comparison. On the test set, the random baseline is repeated for five times to obtain the average score and corresponding standard deviation.

k -Nearest Neighbor Additionally, to investigate whether the retrieval module is complementary to GPT-3’s few-shot learning ability, we further consider a k -nearest neighbor baseline. Specifically, for text generation tasks, the target y_1 associated with the first retrieved example is considered as the predicted target for the test sample. As to the sentiment analysis and QA tasks, the top k retrieved examples $\{y_1, \dots, y_k\}$ are utilized, where the final prediction is determined by majority voting among the k examples’ targets. If there is a tie case, we take the target of the example that is most similar to the test sentence as the prediction. To ensure fair comparison, we compare the baseline $k\text{NN}$ and KATE under the same embedding space of a pre-trained RoBERTa-large model. This baseline is abbreviated as $k\text{NN}_{\text{roberta}}$.

4 Experimental Results

4.1 Sentiment Analysis

We first evaluate KATE on the sentiment analysis task. The results are shown in Table 4. It can be observed that KATE consistently produces better performance relative to the random selection baseline. Notably, there is no variance with the obtained results since the same set of retrieved in-context examples are employed. For the KATE method, when a pre-trained sentence encoder is fine-tuned on NLI or NLI+STS-B datasets, the performance slightly decreases. Since the objectives of the IMDB dataset and the NLI+STS-B datasets are different, this shows that fine-tuning on a dissimilar task can hurt KATE’s performance. Moreover, KATE_{nli+sts-b} performs worse than KATE_{nli} because the sentence encoder has been further fine-tuned on the STS-B dataset.

In contrast, KATE_{sst-2} obtains the best accuracy, showing that fine-tuning on a similar task can benefit KATE’s performance. To verify that the gains are not merely from the retrieval step, we further compare KATE_{roberta} with the $k\text{NN}_{\text{roberta}}$. It turns out that the performance of the $k\text{NN}_{\text{roberta}}$ method is similar to random guessing. This observation is consistent when one neighbor or three neighbors are retrieved. Notably, with the embeddings of the RoBERTa-large model fine-tuned on the SST-2 dataset, the accuracy of $k\text{NN}_{\text{sst-2}}$ is 92.46, which is lower than that obtained with KATE_{sst-2}. These results suggest that the GPT-3 model is critical to the final results, and the retrieval module is complementary to GPT-3’s few-shot capabilities.

4.2 Table-to-text Generation

We utilize the ToTTo dataset to evaluate KATE on the table-to-text generation task. The results are shown in Table 5. The KATE method gives rise to considerable gains over the random baseline, according to both the BLEU and PARENT scores. On a finer scale, the evaluation can be done on the overlap subset and the nonoverlap subset. The overlap dev subset shares a significant number of header names with the training set, while the nonoverlap one does not share any header names. It can be observed that the KATE method improves the results on both the overlap and the nonoverlap subsets, meaning that the retrieval module is helpful for both situations where the test set follows the distribution of the training set and where the test set is out of distribution of the training set. Similar to sentiment analysis, there is a slight drop in performance from KATE_{roberta} to KATE_{nli} and KATE_{nli+sts-b}. This is due to the difference between the objectives of the ToTTo dataset and NLI+STS-B datasets. The drop from KATE_{nli} to KATE_{nli+sts-b} further validates the idea that fine-tuning on a dissimilar task can hurt KATE’s performance. For the $k\text{NN}$ baseline, it performs much worse than the random selection method and the KATE method, again suggesting that the retrieval process and GPT-3 work together collaboratively to achieve better results.

To understand how the retrieval mechanism helps GPT-3’s predictions, we conduct a case study on the retrieved examples (see Table 6). By retrieving relevant examples from the training set, KATE provides useful detailed information within the table, *e.g.*, the number of points, rebounds, and assists, to GPT-3 for more accurate description.

On the other hand, the random selection method has the issue of hallucination, where the generated sequences contain information (*i.e.*, “senior year” and “University of Texas”) not present in the table.

4.3 Questing Answering

We also evaluate KATE on the open-domain QA tasks, as shown in Table 7. For the QA tasks, we compare with some state-of-the-art methods such as RAG (Lewis et al., 2020) and T5 (Raffel et al., 2019). Both methods require fine-tuning on the specific datasets. The KATE method again improves GPT-3’s few-shot prediction accuracies substantially across various benchmarks. It is worth noting that the fine-tuned transformer models serve as better sentence encoders for retrieval purpose (compared with the RoBERTa-large model without fine-tuning). KATE_{nli} and KATE_{nli+sts-b} improve upon KATE_{roberta} because this time fine-tuning on NLI or STS-B datasets is helpful for retrieving semantically similar questions from the QA datasets. Moreover, on the NQ and TriviaQA datasets, further fine-tuning on the STS-B dataset improves KATE’s results. We also try reducing the number of in-context examples to be as small as five for both the random and KATE methods, where KATE outperforms the baseline as well. Therefore, the advantage of KATE over the random baseline holds for both small and large numbers of in-context examples. More details can be found in Section 5.1. We evaluate the other baseline $k\text{NN}_{\text{roberta}}$ by using the top-1 nearest neighbor. We also explore using 64 nearest neighbors (10 for TriviaQA) to determine the answer (by majority voting explained in Section 3.2). The EM score tends to be similar to retrieving the top-1 nearest neighbor. These $k\text{NN}$ baseline results again suggest that the retrieval module and GPT-3 work together to achieve better performance.

To investigate why the retrieval examples are helpful, we further present a case study. Concretely, the retrieved in-context examples from the NQ dataset are shown in Table 8. For the first and second cases, the random baseline provides wrong answers because GPT-3 is unable to recall the exact detail. However, the in-context examples selected by KATE contain the correct details, which facilitates GPT-3 to answer the questions. For the third test question, the random baseline leads GPT-3 to misinterpret the question as asking for a specific location. In contrast, KATE selects similar

Method	Overall		Overlap Subset		Nonoverlap Subset	
	BLEU	PARENT	BLEU	PARENT	BLEU	PARENT
Random	28.4 ± 2.1	39.3 ± 2.6	31.2 ± 2.5	41.8 ± 3.0	25.6 ± 1.8	37.0 ± 2.3
$k\text{NN}_{\text{roberta}}$	14.1	12.6	20.1	17.9	8.0	7.52
KATE _{roberta}	40.3	49.7	47.8	55.0	32.9	44.6
KATE _{nli}	39.1	48.5	46.5	53.7	31.9	43.6
KATE _{nli+sts-b}	38.1	47.2	45.2	52.2	31.1	42.4

Table 5: Table-to-text generation results on the ToTTo dev dataset.

Test Table	Table: <page_title>Trey Johnson <section_title>College <table><cell>32 <col_header>GP <cell>4.8 <col_header>RPG <cell>2.3 <col_header>APG <cell>23.5 <col_header>PPG Table: <page_title>Dedric Lawson <section_title>College <table><cell>9.9 <col_header>RPG <cell>3.3 <col_header>APG <cell>19.2 <col_header>PPG Sentence: Dedric Lawson averaged 19.2 points, 9.9 rebounds and 3.3 assists per game. Table: <page_title>Carsen Edwards <section_title>College <table><cell>3.8 <col_header>RPG <cell>2.8 <col_header>APG <cell>18.5 <col_header>PPG Sentence: Edwards averaged 18.5 points, 3.8 rebounds and 2.8 assists per game.
Retrieved Examples	Ground-truth: Trey Johnson averaged 23.5 points, 4.8 rebounds, and 2.3 assists in 32 games. Random: Trey Johnson averaged 23.5 points per game in his senior year at the University of Texas. KATE: Johnson averaged 23.5 points, 4.8 rebounds and 2.3 assists per game.
Predictions	

Table 6: A sample of retrieved in-context examples from the ToTTo dataset. The first block shows a table string from the test set. The second block provides two retrieved examples. The last block includes the ground-truth sentence and sentences by the random selection and KATE methods. For the KATE method, GPT-3 pays more attention to detailed information such as the number of points, rebounds, and assists. In contrast, the random selection method leads GPT-3 to generate details which do not exist in the original table.

Method	NQ	WQ	TriviaQA*
RAG (Open-Domain)	44.5	45.5	68.0
T5+SSM (Closed-Book)	36.6	44.7	60.5
T5 (Closed-Book)	34.5	37.4	50.1
GPT-3 (64 examples)	29.9	41.5	-
Ours			
Random	28.6 ± 0.3	41.0 ± 0.5	59.2 ± 0.4
$k\text{NN}_{\text{roberta}}$	24.0	23.9	26.2
KATE _{roberta}	40.0	47.7	57.5
KATE _{nli}	40.8	50.6	60.9
KATE _{nli+sts-b}	41.6	50.2	62.4

Table 7: QA results on various datasets. (*) On TriviaQA, we used 10 examples. On NQ and WQ, we used 64 examples.

questions which ask for the origins of objects. Using these in-context examples, GPT-3 is able to interpret and answer the question correctly.

5 Analysis and Ablation Study

5.1 Number of In-context Examples

We first investigate the impact of the number of in-context examples on KATE’s performance. Concretely, on the NQ dataset, we choose the number of in-context examples to be 5, 10, 20, 35, and 64, and KATE_{nli+sts-b} is compared with the random baseline and KATE_{roberta} across different settings. As shown in the left plot of Figure 3, both KATE and the random baseline benefit from

utilizing more in-context examples. However, KATE consistently outperforms the random selection method, even when the number of in-context examples is as few as 5. This result is interesting because in practice, employing less in-context leads to more efficient inference with GPT-3.

5.2 Size of Training Set for Retrieval

We further examine how the size of the training set may influence the KATE method. On the NQ dataset, we create new subsets from the original training set, with sizes of 1k, 2k, 5k, 10k, 30k, and 70k, respectively. In-context examples are retrieved from these subsets instead of the original training set. The number of nearest neighbors is set to 64. We compare KATE_{nli+sts-b} with the random selection method and KATE_{roberta}, and the results are shown in the right plot of Figure 3. For KATE_{roberta} and KATE_{nli+sts-b}, as the size of the training set for retrieval increases, the EM scores also increase. In contrast, the result of the random sampling baseline does not change much. Intuitively, as the training size gets larger, it is more likely for KATE to retrieve relevant in-context examples to help GPT-3 answer a question correctly. As we have shown previously in Table 8, the retrieved in-context examples could provide critical detailed information to GPT-3, thus helping GPT-3 to better answer the questions.

In-Context Examples	Predictions
Question: The Mughal Gardens of Rashtrapati Bhavan is modelled on which garden?	
The Mughal Garden of Rashtrapati Bhavan is modelled on? The Persian style of architecture Who built the first Mughal Garden in India? Babur The landscape design of the Gardens of Versailles is known as which style? French garden	Ground-truth: Persian garden KATE: The Persian gardens Random Baseline: Shalimar gardens
Question: What city was Zeus the patron god of?	
What is the symbol of Zeus the Greek God? Bull Where did Zeus spend most of his time? Mount Olympus Where was the statue of Zeus at Olympia located? In the Temple of Zeus	Ground-truth: Olympia KATE: Olympia Random Baseline: Athens
Question: Where did the Dewey decimal system come from?	
Where did the formula for area of a circle come from? Archimedes Where did the name jack russell come from? Reverend John Russell Where did the letters of the alphabet come from? The Phoenician alphabet	Ground-truth: Melvil Dewey KATE: Melvil Dewey Random Baseline: the library of Congress

Table 8: Samples of retrieved in-context examples from the NQ dataset. Each block begins with a test question. Three retrieved Q-A pairs are shown on the left. Predictions by the KATE method and useful details from in-context examples are shown in **Green**. Gold-standard references are shown in **Blue**. Predictions by the random selection method are shown in **Red**.

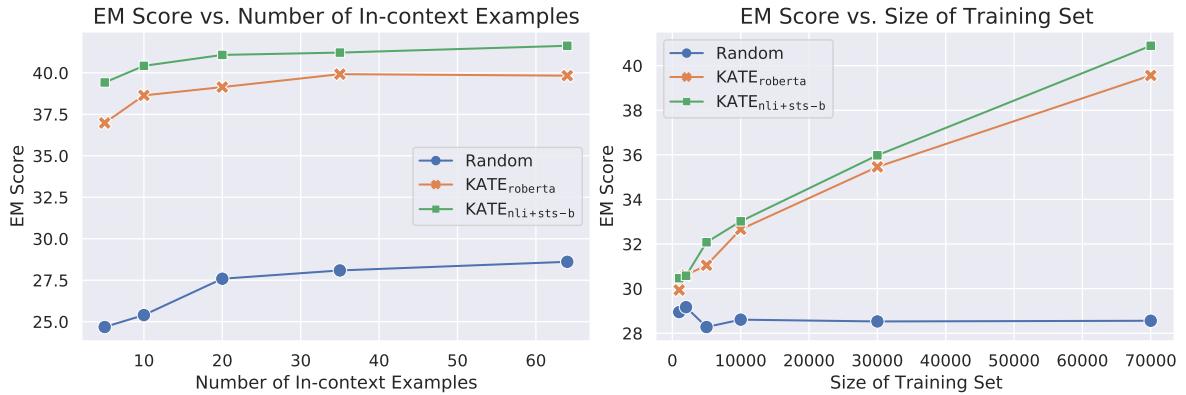


Figure 3: Left: Ablation study on the effect of number of in-context examples for GPT-3 for different selection methods. Right: Ablation study on the effect of the size of training set for retrieval on KATE. Two representative sentence encoders are used in the ablation study.

Trial	1	2	3	Default	Reverse
EM Score	42.0	42.5	42.0	41.6	42.8

Table 9: Ablation study on the effect of in-context example orders for GPT-3 on the NQ dataset using KATE_{nli+sts-b}. For the default order, the example *A* is to the left of example *B* if *A* is closer to the test prompt x than *B* in the embedding space. For the reverse order, the example *A* is to the right of example *B*.

5.3 Order of In-context Examples

Moreover, we explore how the order of in-context examples may affect KATE’s results. As mentioned in Section 2.3, under the standard setting, the retrieved in-context examples are ordered such that $d(x_i, x) \leq d(x_j, x)$ whenever $i < j$. Here, we randomly permute the order of in-context examples in the NQ dataset for the proposed KATE_{nli+sts-b} method, and conduct the

experiments for 3 different orders. Additionally, we explore the reverse order where $d(x_i, x) \leq d(x_j, x)$ whenever $i < j$. The results are presented in Table 9. On this particular NQ dataset, the reverse order performs the best. One possible explanation is that since tokens next to each other have similar positional embeddings, putting the most similar sentences close to the test example may be helpful for GPT-3 to leverage the corresponding information. However, we also did the experiments on the WQ and TriviaQA and find that the default order performs slightly better than the reverse order. Hence, the choice of orders is data-dependent. Additionally, it can be observed that the variation among the NQ results tends to be quite small (compared with the difference between the random baseline and KATE), indicating that the example order does not have a significant

impact on KATE’s performance.

6 Related Work

Pre-trained Language Models NLP systems have made tremendous progress by pre-training models on unlabeled text. For text classification tasks, notable models include BERT (Devlin et al., 2018), RoBERTa (Liu et al., 2019), and XLNet (Yang et al., 2019). For text generation tasks, notable models include BART (Lewis et al., 2019), T5 (Raffel et al., 2019), mT5 (Xue et al., 2020), XLM (Lample and Conneau, 2019), GPT (Radford et al., 2018), and GPT-2 (Radford et al., 2019). These models encapsulate rich information to facilitate a wide range of downstream tasks ranging from natural language understanding to generation. These models can be adapted to many different tasks via fine-tuning. GPT-3 (Brown et al., 2020), however, can be adapted to many downstream tasks without fine-tuning. Given just a few in-context examples, GPT-3 is able to quickly pick up patterns and produce answers analogously both in terms of the answer style and content. Thus, GPT-3 may be considered as a pattern recognizer to perform in-context learning. People have just started trying to understand GPT-3 from different perspectives. As mentioned in the introduction, (Hendrycks et al., 2020) studies which categories of questions GPT-3 is more capable of answering. Our work focuses on how to choose good in-context examples.

Retrieval-based Text Generation There is a long history of applying information retrieval in text generation (Sumita and Hitoshi, 1991). It is very related to the exemplar-based learning (Jäkel et al., 2008; Ziyadi et al., 2020). The central idea is to treat retrieved samples as exemplars/prototypes and perform some editings on them. Some representative applications in the field of deep learning include machine translation (Gu et al., 2018), sentiment transfer (Li et al., 2018; Guu et al., 2018), QA (Karpukhin et al., 2020; Mao et al., 2020), dialogue generation (Yan et al., 2016; Cai et al., 2018; Song et al., 2016; Pandey et al., 2018; Weston et al., 2018; Wu et al., 2019), text summarization (Cao et al., 2017; Peng et al., 2019), data-to-text generation (Peng et al., 2019), and text-to-code generation (Hashimoto et al., 2018). However, all these retrieve-and-edit frameworks require their decoders to be trained from scratch. This makes the editor network task- and data-

specific. In contrast, GPT-3 in one perspective can be regarded naturally as a universal editor, adaptive to a wide range of tasks. Our work uniquely examines how to maximize the advantage of using GPT-3 without fine-tuning. For example, the more semantically similar context we provide to GPT-3, the better results the model can generate. Other editors or generators do not have this ability.

Improve NLP Systems with k NN A recent line of work tries to incorporate nonparametric methods to improve a given model’s performance. These methods first access the test sample’s hidden representation and look for the nearest neighbors of this test sample in the database. Once the nearest neighbors are found, their labels are used to augment the model’s prediction. For example, the newly introduced k NN-LM (Khandelwal et al., 2019), k NN-MT (Khandelwal et al., 2020), and BERT- k NN (Kassner and Schütze, 2020) generate the next token by retrieving the nearest k neighbors from the datastore. Another related work is k NN classification model (Rajani et al., 2020), where they use k NN as backoff when the confidence is low from the fine-tuned classification model. There are two key differences between our work and other approaches. First, other approaches modifies the model’s next token distribution using the nearest k neighbors. However, we only changes the conditional text using the nearest k neighbors. Second, other approaches can access the model’s parameters and embeddings which we do not have access to. Instead, we use some other independently pre-trained models to get the sentence embeddings to retrieve nearest k neighbors.

7 Conclusion

This work presented a first step towards investigating the sensitivity of GPT-3 to in-context examples. To this end, we proposed KATE, a non-parametric selection approach that retrieves in-context examples according to their semantic similarity to the test samples. On several natural language understanding and generation tasks, the proposed method improves GPT-3’s performance, over the random sampling baseline, by a significant margin. Moreover, we found that fine-tuning the sentence embeddings for retrieval on task-related datasets gave rise to further empirical gains. Detailed ablation studies were conducted to explore the robustness of KATE to different hyperparameters, such as the number of in-context ex-

amples, examples’ order, *etc.* We hope this work could provide insights for better understanding the behaviors of GPT-3 and represents a helpful step towards further improving its few-shot capabilities.

References

- Jonathan Berant, Andrew Chou, Roy Frostig, and Percy Liang. 2013. Semantic parsing on freebase from question-answer pairs. In *Proceedings of the 2013 conference on empirical methods in natural language processing*, pages 1533–1544.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *arXiv preprint arXiv:2005.14165*.
- Deng Cai, Yan Wang, Wei Bi, Zhaopeng Tu, Xiaojiang Liu, Wai Lam, and Shuming Shi. 2018. Skeleton-to-response: Dialogue generation guided by retrieval memory. *arXiv preprint arXiv:1809.05296*.
- Ziqiang Cao, Furu Wei, Wenjie Li, and Sujian Li. 2017. Faithful to the original: Fact aware neural abstractive summarization. *arXiv preprint arXiv:1711.04434*.
- 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*.
- Bhuwan Dhingra, Manaal Faruqui, Ankur Parikh, Ming-Wei Chang, Dipanjan Das, and William W Cohen. 2019. Handling divergent reference texts when evaluating table-to-text generation. *arXiv preprint arXiv:1906.01081*.
- Jiatao Gu, Yong Wang, Kyunghyun Cho, and Victor OK Li. 2018. Search engine guided neural machine translation. In *AAAI*, pages 5133–5140.
- Kelvin Guu, Tatsunori B Hashimoto, Yonatan Oren, and Percy Liang. 2018. Generating sentences by editing prototypes. *Transactions of the Association for Computational Linguistics*, 6:437–450.
- Tatsunori B Hashimoto, Kelvin Guu, Yonatan Oren, and Percy S Liang. 2018. A retrieve-and-edit framework for predicting structured outputs. In *Advances in Neural Information Processing Systems*, pages 10052–10062.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.
- Frank Jäkel, Bernhard Schölkopf, and Felix A Wichmann. 2008. Generalization and similarity in exemplar models of categorization: Insights from machine learning. *Psychonomic Bulletin & Review*, 15(2):256–271.
- Mandar Joshi, Eunsol Choi, Daniel S Weld, and Luke Zettlemoyer. 2017. Triviaqa: A large scale distantly supervised challenge dataset for reading comprehension. *arXiv preprint arXiv:1705.03551*.
- Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. *arXiv preprint arXiv:2004.04906*.
- Nora Kassner and Hinrich Schütze. 2020. Bert-knn: Adding a knn search component to pretrained language models for better qa. *arXiv preprint arXiv:2005.00766*.
- Urvashi Khandelwal, Angela Fan, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2020. Nearest neighbor machine translation. *arXiv preprint arXiv:2010.00710*.
- Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.
- Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, et al. 2019. Natural questions: a benchmark for question answering research. *Transactions of the Association for Computational Linguistics*, 7:453–466.
- Guillaume Lample and Alexis Conneau. 2019. Cross-lingual language model pretraining. *arXiv preprint arXiv:1901.07291*.
- Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Ves Stoyanov, and Luke Zettlemoyer. 2019. Bart: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. *arXiv preprint arXiv:1910.13461*.
- Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *arXiv preprint arXiv:2005.11401*.
- Juncen Li, Robin Jia, He He, and Percy Liang. 2018. Delete, retrieve, generate: A simple approach to sentiment and style transfer. *arXiv preprint arXiv:1804.06437*.
- Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019.

- Robert: A robustly optimized bert pretraining approach. *arXiv preprint arXiv:1907.11692*.
- Andrew Maas, Raymond E Daly, Peter T Pham, Dan Huang, Andrew Y Ng, and Christopher Potts. 2011. Learning word vectors for sentiment analysis. In *Proceedings of the 49th annual meeting of the association for computational linguistics: Human language technologies*, pages 142–150.
- Yuning Mao, Pengcheng He, Xiaodong Liu, Yelong Shen, Jianfeng Gao, Jiawei Han, and Weizhu Chen. 2020. Generation-augmented retrieval for open-domain question answering. *arXiv preprint arXiv:2009.08553*.
- Gaurav Pandey, Danish Contractor, Vineet Kumar, and Sachindra Joshi. 2018. Exemplar encoder-decoder for neural conversation generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1329–1338.
- Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics*, pages 311–318.
- Ankur P Parikh, Xuezhi Wang, Sebastian Gehrmann, Manaal Faruqui, Bhuwan Dhingra, Diyi Yang, and Dipanjan Das. 2020. ToTTo: A controlled table-to-text generation dataset. In *Proceedings of EMNLP*.
- Hao Peng, Ankur P Parikh, Manaal Faruqui, Bhuwan Dhingra, and Dipanjan Das. 2019. Text generation with exemplar-based adaptive decoding. *arXiv preprint arXiv:1904.04428*.
- Alec Radford, Karthik Narasimhan, Tim Salimans, and Ilya Sutskever. 2018. Improving language understanding by generative pre-training.
- Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.
- Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J Liu. 2019. Exploring the limits of transfer learning with a unified text-to-text transformer. *arXiv preprint arXiv:1910.10683*.
- Nazneen Fatema Rajani, Ben Krause, Wengpeng Yin, Tong Niu, Richard Socher, and Caiming Xiong. 2020. Explaining and improving model behavior with k nearest neighbor representations. *arXiv preprint arXiv:2010.09030*.
- Nils Reimers and Iryna Gurevych. 2019. Sentencebert: Sentence embeddings using siamese bert networks. *arXiv preprint arXiv:1908.10084*.
- Nils Reimers and Iryna Gurevych. 2020. Making monolingual sentence embeddings multilingual using knowledge distillation. *arXiv preprint arXiv:2004.09813*.
- 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.
- Yiping Song, Rui Yan, Xiang Li, Dongyan Zhao, and Ming Zhang. 2016. Two are better than one: An ensemble of retrieval-and-generation-based dialog systems. *arXiv preprint arXiv:1610.07149*.
- Eiichiro Sumita and HDA Hitoshi. 1991. Experiments and prospects of example-based machine translation. In *29th Annual Meeting of the Association for Computational Linguistics*, pages 185–192.
- 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*, 30:5998–6008.
- Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R Bowman. 2018. Glue: A multi-task benchmark and analysis platform for natural language understanding. *arXiv preprint arXiv:1804.07461*.
- Jason Weston, Emily Dinan, and Alexander H Miller. 2018. Retrieve and refine: Improved sequence generation models for dialogue. *arXiv preprint arXiv:1808.04776*.
- Yu Wu, Furu Wei, Shaohan Huang, Yunli Wang, Zhoujun Li, and Ming Zhou. 2019. Response generation by context-aware prototype editing. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 33, pages 7281–7288.
- Linting Xue, Noah Constant, Adam Roberts, Mihir Kale, Rami Al-Rfou, Aditya Siddhant, Aditya Barua, and Colin Raffel. 2020. mt5: A massively multilingual pre-trained text-to-text transformer. *arXiv preprint arXiv:2010.11934*.
- Rui Yan, Yiping Song, and Hua Wu. 2016. Learning to respond with deep neural networks for retrieval-based human-computer conversation system. In *Proceedings of the 39th International ACM SIGIR conference on Research and Development in Information Retrieval*, pages 55–64.
- Zhilin Yang, Zihang Dai, Yiming Yang, Jaime Carbonell, Russ R Salakhutdinov, and Quoc V Le. 2019. XLnet: Generalized autoregressive pretraining for language understanding. In *Advances in neural information processing systems*, pages 5753–5763.

Morteza Ziyadi, Yuting Sun, Abhishek Goswami, Jade Huang, and Weizhu Chen. 2020. Example-based named entity recognition. *arXiv preprint arXiv:2008.10570*.