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

Jiachang Liu[1]*, **Dinghan Shen**[2], **Yizhe Zhang**[3], **Bill Dolan**[4], **Lawrence Carin**[1], **Weizhu Chen**[2]
[1]Duke University [2]Microsoft Dynamics 365 AI [3]Meta AI [4]Microsoft Research
[1]{jiachang.liu, lcarin}@duke.edu
[3]yizhe.zhang@hotmail.com
[2,4]{dishen, billdol, wzchen}@microsoft.com

Abstract

GPT-3 has attracted lots of attention due to its superior performance across a wide range of NLP tasks, especially with its in-context learning abilities. 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 in-context learning capabilities. Inspired by the recent success of leveraging a retrieval module to augment neural networks, we propose to retrieve examples that are semantically-similar to a test query sample to formulate its corresponding prompt. Intuitively, the examples selected with such a strategy may serve as more informative inputs to unleash GPT-3's power of text generation. We evaluate the proposed approach on several natural language understanding and generation benchmarks, where the retrieval-based prompt selection approach consistently outperforms the random selection baseline. Moreover, it is observed that the sentence encoders fine-tuned on task-related datasets yield even more helpful retrieval results. Notably, significant gains are observed on tasks such as table-to-text generation (44.3% on the ToTTo dataset) and open-domain question answering (45.5% on the NQ dataset).

1 Introduction

GPT-3 (Brown et al., 2020) is a new breakthrough in NLP research. Previously, NLP models are firstly pre-trained and then fine-tuned on a specific task. What sets GPT-3 apart from other models is its impressive "in-context" learning ability. Provided with a few in-context examples, GPT-3 can generalize to unseen cases without further fine-tuning. This opens up many new technological possibilities that are previously considered unique to human. Future NLP systems can be developed to expand emails, extract entities from text, generate code based on natural language instructions with a few demonstration examples.

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 SST-2 sentiment analysis dataset. Five different examples are randomly selected from the training set for each trial. Different contexts induce different accuracies on the test set.

Despite its powerful and versatile in-context learning ability, GPT-3 has some practical challenges. The original paper utilizes task-relevant examples that are randomly sampled from the training set. However, 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 with distinct in-context examples can be significant. Our work aims to carefully examine this issue to gain a deeper understanding on how to better select in-context examples to improve GPT-3's performance without fine-tuning. Note that our approach requires a training set to select examples. With such a training dataset, it is possible to fine-tune GPT-3 to take full advantage of the model's strength. However, currently GPT-3 has not been released to public for fine-tuning. Even if it is available, fine-tuning GPT-3 requires hundreds of GPUs to load the 175B model, which is prohibitively expensive and time-consuming for ordinary research labs. Another issue is that storing large fine-tuned model checkpoints require huge storage space. Consequently, we resort to prompt/example engineering strategy. Nevertheless, the fine-tuning results using T5 are provided for reference.

A brute-force approach for selecting the optimal in-context instances would be to perform combinatorial search over the entire dataset. Unfortunately, this strategy is computationally impractical. To this

*Work was done when Jiachang (intern) and Yizhe were at Microsoft.

Proceedings of Deep Learning Inside Out (DeeLIO 2022):
The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures, pages 100 - 114

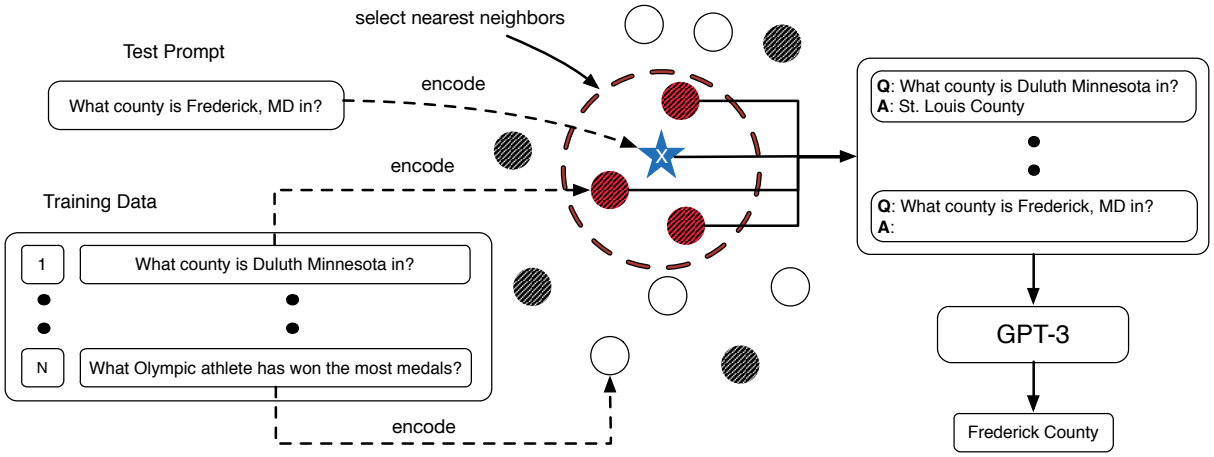

Figure 1: In-context example selection for GPT-3. White dots: unused training samples; grey dots: randomly sampled training samples; red dots: training samples selected by the k-nearest neighbors algorithm in the embedding space of a sentence encoder.

end, we empirically investigate the influences of employing different in-context examples. Interestingly, we find 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 in-context examples.

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 in-context learning capabilities of GPT-3 much more effectively than the random sampling baseline, even when the number of examples is small. Moreover, we find that the specific sentence encoders employed for the retrieval procedure play a critical role. Thus, an extensive exploration is conducted and shows that encoders fine-tuned on natural language matching tasks serve as more effective in-context examples selector on the QA task. In summary, our contributions are as follows:

i) to the best of our knowledge, we take a first step towards understanding the sensitivity of GPT-3's in-context learning ability with respect to the choice 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, which greatly outperforms the baseline based on randomly sampled in-context examples;

iii) empirically, the better selected examples lead GPT-3 to achieve comparable performance to a fine-tuned T5 model on the table-to-text task and *outperforms* the T5 model on the QA tasks;

iv) fine-tuning the retrieval model on task-related dataset(s) leads to stronger empirical results;

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

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 probability can be expressed as:

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

where LM denotes the parameters of the language model, and $C = \{x_1, y_1, x_2, y_2, ..., x_k, y_k\}$ is a context string concatenating k training instances with the special character "\n". A concrete illustration can be found in the Appendix.

For GPT-3, this generation process is implemented through a giant transformer-based architecture (Vaswani et al., 2017). Due to the computational burden of fine-tuning, GPT-3 is leveraged in an in-context learning manner as described above. Unfortunately, as shown in Table 1, the results of GPT-3 tend to fluctuate significantly with different in-context examples. We aim to alleviate this issue via judicious in-context example selection.

2.2 The Impact of In-Context Examples

We start the investigation by looking at the role of in-context examples from an empirical perspective. Previous retrieve-and-edit literature 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 the 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 selection strategies. Given a test example, the first method utilizes the 10 farthest training instances as the in-context examples, while the second employs the 10 closest neighbors. We use the CLS embeddings of a pre-trained RoBERTa-large model as 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 Table 2. It can be observed that the nearest neighbors, used 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 kNN-augmented Example Selection

Based on the findings above, we propose KATE[1], a strategy to select good examples for in-context learning. The process is visualized in Figure 1. Specifically, we first use a 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 $x_1, x_2, ..., x_k$ from the training set (according to the distances in the embedding space). Given some pre-defined similarity measure s such as the negative Euclidean distance or the cosine similarity, the neighbors are ordered so that $s(x_i, x) \geq s(x_j, x)$ when $i < j$.

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

[1]KATE: **K**nn-**A**ugmented in-con**T**ext **E**xample selection

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.

Algorithm 1 kNN In-context Example Selection

Given: test prompt $\boldsymbol{x}_{\text{test}}$, training set $\mathcal{D}_T = \{\boldsymbol{x}_i, \boldsymbol{y}_i\}_{i=1}^N$, sentence encoder $\mu_\theta(\cdot)$, and number of in-context examples k (hyperparameter).

1: $\boldsymbol{v}_{\text{test}} = \mu_\theta(\boldsymbol{x}_{\text{test}})$
2: **for** $\boldsymbol{x}_i \in \mathcal{D}_T$ **do**
3: $\quad \boldsymbol{v}_i = \mu_\theta(\boldsymbol{x}_i)$
4: $\quad \boldsymbol{s}_i = -\|\boldsymbol{v}_{\text{test}} - \boldsymbol{v}_i\|_2 \qquad (\text{or } \frac{\boldsymbol{v}_{\text{test}} \cdot \boldsymbol{v}_i}{\|\boldsymbol{v}_{\text{test}}\|_2 \|\boldsymbol{v}_i\|_2})$
5: **end for**
6: Select largest k similarities $\boldsymbol{s}_i$'s (in descending order) with indices $\{\sigma(1), ..., \sigma(k)\}$
7: $C = [\boldsymbol{x}_{\sigma(1)}; \boldsymbol{y}_{\sigma(1)}; ...; \boldsymbol{x}_{\sigma(k)}; \boldsymbol{y}_{\sigma(k)}]$
8: $\hat{\boldsymbol{y}}_{\text{test}} = \text{GPT-3}([C; \boldsymbol{x}_{\text{test}}])$

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 generally pre-trained sentence encoders such as the BERT, RoBERTa, and 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 dataset should be able to assess similarities among different questions better than a generally pre-trained sentence encoder. Sentence-BERT (Wolf et al., 2019; Reimers and Gurevych, 2019, 2020) shows 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 our proposed method to the following three tasks: sentiment analysis, table-to-text generation, and question answering. Dataset split setups and prompt templates are shown in Table 9 and 11 in the Appendix. For the hyper-parameters in the GPT-3 API, we set the temperature to 0.

3.1 Sentence Embeddings for Retrieval

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

- The original RoBERTa-large model (Liu et al., 2019), which is abbreviated as $\text{KATE}_{\text{roberta}}$;
- The RoBERTa-large models which are: *i*) fine-tuned on the SNLI and MultiNLI datasets (KATE_{nli}) (Bowman et al., 2015; Williams et al., 2017); *ii*) first fine-tuned on the SNLI and MultiNLI dataset and then on the STS-B datasets ($\text{KATE}_{\text{nli+sts-b}}$) (Cer et al., 2017).

All sentence encoders share the same architecture. The only differences are the specific datasets used for fine-tuning. The negative Euclidean distance is used for $\text{KATE}_{\text{roberta}}$, while the cosine similarity is employed for KATE_{nli} and $\text{KATE}_{\text{nli+sts-b}}$.

Sentiment Analysis For this task, we conduct experiments under the dataset-transfer setting. In-context examples are selected from one dataset, and the evaluation is made on another dataset. This setting is designed to simulate a real-world scenario where we want to leverage an existing labeled dataset for a unlabeled one (of a similar task).

Specifically, we select examples from the SST-2 training set (Socher et al., 2013; Wang et al., 2018) and ask GPT-3 to predict on the IMDB test set (Maas et al., 2011). To explore whether a sentence encoder fine-tuned on a similar task would benefit KATE, we also employ a pre-trained RoBERTa-large model fine-tuned on the SST-2 training set (dubbed as $\text{KATE}_{\text{sst-2}}$). The number of 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)[2] is utilized for evaluation due to its popularity. We use BLEU (Papineni et al., 2002) and PARENT (Dhingra et al., 2019) metrics for evaluation. Because the token length limit of GPT-3 is 2048, we add a preprocessing step by deleting the closing angle brackets such as </cell> and </table> to save space. The number of in-context examples is set as 2 so that the input length is within the token limit.

Question Answering We conduct experiments on three QA benchmarks: Natural Questions (NQ) (Kwiatkowski et al., 2019), Web Questions (WQ) (Berant et al., 2013), and TriviaQA (Joshi et al., 2017). For evaluation, we use the Exact Match (EM) score, which is defined as the proportion of the number of predicted answers being exactly one of the ground-truth answers. The matching is performed after string normalization, which includes article and punctuation removal. The number of examples is set to be 64 for NQ and WQ and 10 for TriviaQA (The retrieved 64 examples exceed the token limit). We evaluate 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. 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 in-context learning ability, we further consider a k-nearest neighbor baseline. Specifically, the target y_1 associated with the first retrieved example is considered as the predicted target for the test sample. For the sentiment analysis and QA tasks, the top k retrieved examples $\{y_1, ..., y_k\}$ are utilized, where the final prediction is determined by majority voting among the k examples' targets. If there is a tie case, we use the target of the example most similar to the test sentence. To ensure fair comparison, we compare the baseline kNN and KATE under the same embedding space of a pre-trained RoBERTa-large model. This baseline is abbreviated as $k\text{NN}_{\text{roberta}}$.

Fine-tuned T5 Although this work aims at improving the in-context learning abilities of GPT-3, we include a fine-tuned T5 (3B) model as a baseline. This comparison informs us where GPT-3 performs comparably or surpasses a fine-tuned model.

[2]The ToTTo code base and evaluation scripts can be found at https://github.com/google-research/language/tree/master/language/totto

Method	Accuracy
T5 (fine-tuned)	95.2
Ours	
Random	87.95 ± 2.74
kNN$_{\text{roberta}}$	50.20
KATE$_{\text{roberta}}$	91.99
KATE$_{\text{nli}}$	90.40
KATE$_{\text{nli+sts-b}}$	90.20
KATE$_{\text{sst-2}}$	**93.43**

Table 3: Results on the IMDB dataset. In-context examples are from the SST-2 dataset.

4 Experimental Results

4.1 Sentiment Analysis

We first evaluate KATE on the sentiment analysis task. The results are in Table 3. KATE consistently produces better performance relative to the random selection baseline. Notably, there is no variance with the obtained results since the fixed retrieved in-context examples are employed. For KATE, when the pre-trained sentence encoder is fine-tuned on NLI or NLI+STS-B datasets, the performance slightly decreases. Since the objectives of the IMDB and the NLI+STS-B datasets are different, this shows that fine-tuning on a dissimilar task hurts KATE's performance. In contrast, KATE$_{\text{sst-2}}$ obtains the best accuracy, showing that fine-tuning on a similar task improves KATE's performance. To verify that the gains are not merely from the retrieval step, we further compare KATE$_{\text{roberta}}$ with the kNN$_{\text{roberta}}$. It turns out that the performance of kNN$_{\text{roberta}}$ is close to random guessing. This observation is consistent when one neighbor or three neighbors are retrieved. Notably, with the sentence encoder fine-tuned on the SST-2 dataset, the accuracy of kNN$_{\text{sst-2}}$ is 92.46, which is lower than that of KATE$_{\text{sst-2}}$. These results suggest that GPT-3 is critical to the final results, and the retrieval module is complementary to GPT-3.

The fine-tuned T5 model works better since its parameters has been adapted to this specific task. However, fine-tuning requires access to model parameters, lots of memory storage, and time. The fine-tuning result here is just for reference. Through KATE, the performance of GPT-3 has increased significantly without fine-tuning.

4.2 Table-to-text Generation

We next evaluate KATE on the ToTTo dataset and present results in Table 4. KATE gives rise to considerable gains over the random baseline, according to both the BLEU and PARENT scores. Notably, KATE enables GPT-3 to achieve performance comparable to a fine-tuned T5 model. 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. KATE improves results on both subsets, meaning that the retrieval module is helpful even when the dev set is out of distribution of the training set. Similar to sentiment analysis, there is a slight drop in performance from KATE$_{\text{roberta}}$ to KATE$_{\text{nli}}$ and KATE$_{\text{nli+sts-b}}$. This is due to the difference between the objectives of the ToTTo dataset and NLI+STS-B datasets. The drop from KATE$_{\text{nli}}$ to KATE$_{\text{nli+sts-b}}$ further validates the idea that fine-tuning on a dissimilar task can hurt KATE's performance. For the kNN baseline, it performs much worse than the random selection method and KATE, suggesting that the retrieval process and GPT-3 work collaboratively to achieve better results.

To understand how the retrieval mechanism helps GPT-3, we conduct a case study on the retrieved examples (see Table 5). 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

Lastly, we evaluate KATE on the open-domain QA tasks, as shown in Table 6. We compare with some state-of-the-art fine-tuned methods such as RAG (Lewis et al., 2020) and T5 (Raffel et al., 2019). The T5 results were reported in (Brown et al., 2020) using the 11B model, which needs specialized TPUs to do fine-tuning. KATE again improves GPT-3's performance substantially across various benchmarks. Moreover, KATE helps GPT-3 to even outperform the fine-tuned T5 model. It is worth noting that this time both KATE$_{\text{nli}}$ and KATE$_{\text{nli+sts-b}}$ improve upon KATE$_{\text{roberta}}$ because 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 evaluate the baseline kNN$_{\text{roberta}}$ by using the top-1 nearest neighbor. The kNN baseline results again suggest that

Method	Overall		Overlap Subset		Nonoverlap Subset	
	BLEU	PARENT	BLEU	PARENT	BLEU	PARENT
T5 (fine-tuned)	41.2	53.0	46.7	56.1	35.8	50.0
Ours						
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
kNN$_{\text{roberta}}$	14.1	12.6	20.1	17.9	8.0	7.52
KATE$_{\text{roberta}}$	**41.0**	**50.6**	**48.4**	**55.9**	**33.6**	**45.5**
KATE$_{\text{nli}}$	39.9	49.5	47.4	54.6	32.5	44.5
KATE$_{\text{nli+sts-b}}$	38.8	48.2	46.2	53.1	31.5	43.4

Table 4: 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
Retrieved Examples	**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.
Predictions	**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.

Table 5: A sample of retrieved in-context examples from the ToTTo dataset. 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
kNN$_{\text{roberta}}$	24.0	23.9	26.2
KATE$_{\text{roberta}}$	40.0	47.7	57.5
KATE$_{\text{nli}}$	40.8	**50.6**	60.9
KATE$_{\text{nli+sts-b}}$	**41.6**	50.2	**62.4**

Table 6: Results on QA datasets. (*) We used 10 examples for TriviaQA and 64 examples for NQ and WQ.

the retrieval module and GPT-3 work together to achieve better performance. 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 are similar to retrieving the top-1 nearest neighbor.

To investigate why the retrieved examples are helpful, we present a case study. Concretely, the retrieval examples from the NQ dataset are shown in Table 7. 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 facilitate GPT-3 to answer questions. For the third case, the random baseline leads GPT-3 to misinterpret the question as asking for a specific location. In contrast, KATE selects similar types of questions asking for the origins of objects. Using these in-context examples, GPT-3 is able to interpret and answer the question correctly.

5 Analysis of Different Factors

5.1 Number of In-context Examples

We first investigate the impact of the number of examples on KATE's performance. Concretely, on the NQ dataset, we choose the number of examples to be 5, 10, 20, 35, and 64, and KATE$_{\text{nli+sts-b}}$ is compared with the random baseline and KATE$_{\text{roberta}}$ across different settings. As shown in the left plot of Figure 2, both KATE and the random baseline benefit from utilizing more 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 examples 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

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 7: Three samples of retrieved in-context examples from the NQ dataset. 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 baseline are shown in **Red**.

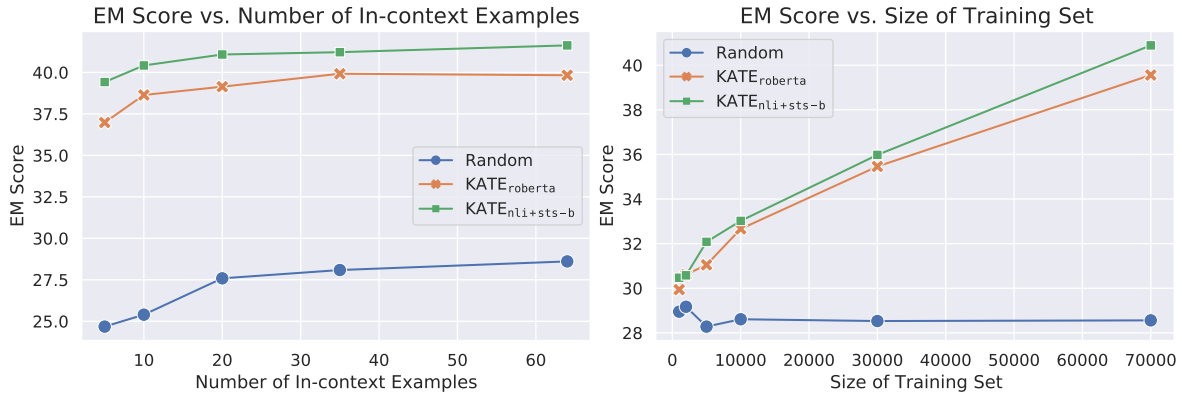

Figure 2: Left: Effect of number of in-context examples for different selection methods. Right: Effect of the size of training set for retrieval on KATE. Two representative sentence encoders are used in these studies.

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 2. For KATE$_{roberta}$ and KATE$_{nli+sts-b}$, as the size of the training set 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 7, the retrieved in-context examples could provide critical detailed information to GPT-3, thus helping GPT-3 to better answer the questions.

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 $s(x_i, x) \geq s(x_j, x)$ 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 $s(x_i, x) \leq s(x_j, x)$ whenever $i < j$. The results are presented in Table 8. On this particular NQ dataset, the reverse order performs the best. 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,

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

Table 8: Analysis on the effect of orders of in-context example on the NQ dataset using KATE$_{nli+sts-b}$. The default order puts the most similar example in the front, and the reverse order does the opposite.

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 (Devlin et al., 2018; Liu et al., 2019; Yang et al., 2019; Lewis et al., 2019; Raffel et al., 2019; Xue et al., 2020; Lample and Conneau, 2019; Radford et al., 2018, 2019). These models can be fine-tuned for a wide range of downstream tasks. GPT-3 (Brown et al., 2020), however, can perform in-context learning without fine-tuning. People have just started trying to understand GPT-3 from different perspectives. (Hendrycks et al., 2020) studies which categories of questions GPT-3 is more capable of answering. (Zhao et al., 2021) proposes to improve the model by contextual calibration. However, their method is limited to predicting very few tokens because for long sequence generation, the contextual calibration step needs to be repeatedly performed after each newly generated token. In contrast, our work, KATE, only calls the API once and is suitable for both text classification and generation tasks. Another related work is LM-BFF (Gao et al., 2020), which uses a smaller language model (RoBERTa-large) to demonstrate that prompt-based fine-tuning can outperform standard fine-tuning on text classification tasks. Our work differs by showing that, without fine-tuning, relevant examples can still substantially improve the performance of GPT-3 for both text classification and generation tasks. Finally, AutoPrompt (Shin et al., 2020) explores adding some additional tokens to smaller language models to improve performance on classification tasks.

Retrieval-based Text Generation There is a long history of applying information retrieval to text generation (Sumita and Hitoshi, 1991). It is very related to the exemplar-based learning (Jäkel et al., 2008; Ziyadi et al., 2020). 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). All these retrieve-and-edit frameworks require their editors to be trained or fine-tuned on specific tasks. In contrast, our work uniquely examines how to better use GPT-3 as a universal editor without fine-tuning. We find that the more semantically similar context we provide to GPT-3, the better results the model can generate.

Improve NLP Systems with kNN Some recent works try to incorporate non-parametric methods to improve a given model's performance. For example, the newly introduced kNN-LM (Khandelwal et al., 2019), kNN-MT (Khandelwal et al., 2020), and BERT-kNN (Kassner and Schütze, 2020) generate the next token by retrieving the nearest k neighbors from the datastore. Another related work kNN classification model (Rajani et al., 2020) uses kNN as backoff when the confidence is low from the classification model. There are two key differences between our work and other approaches. First, we retrieve the nearest k neighbors to modify the conditional context instead of the prediction. Second, we do not have access to the parameters of GPT-3. Instead, we rely on some independently pre-trained models to get the sentence embeddings to retrieve the 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. Particularly, KATE enables GPT-3 to achieve performance comparable to a fine-tuned T5 model on the table-to-text generation task and *outperforms* T5 on the QA task. Moreover, we found that fine-tuning the sentence embeddings for retrieval on task-related datasets gave rise to further empirical gains. Detailed analysis was conducted to explore the robustness of KATE to different hyperprameters, such as the number of in-context examples, examples' order, *etc*. One limitation we notice is that despite the improved performance on sentiment analysis,

GPT-3 still lags behind the fine-tuned T5 model by a small margin. This suggests that our proposed method is more suitable and effective on long text generation tasks. We hope this work could provide insights for better understanding the behaviors of GPT-3 and represents a helpful step towards further improving its in-context learning capabilities.

8 Ethical and Broader Impacts

Risk Our proposed KATE method significantly improves the in-context learning ability of GPT-3 and makes long-text generation more easily without fine-tuning the pre-trained model. However, one risk implication is that our proposed method will benefit the research groups which are financially capable of using such huge models. For individual or small-group researchers, they cannot apply our proposed method to their specific applications since they don't have access to the model. Our work has suggested researchers should focus more on investigating the in-context learning of pre-trained models. One potential future direction is for researchers to scale-down the sizes of pre-trained models to find a balance between model performance and model size. Once a smaller model is obtained with comparable performance (enhanced by KATE), our proposed method can become more widely accessible to individual researchers.

Potential Bias During the experiment on table-to-text generation, we have pointed out that large pre-trained language models could be susceptible to hallucination (case study in Table 5). This problem is more pronounced when we use randomly sampled examples. This happens because the language model is biased toward the training dataset. As shown in Table 5, when random examples are used, the sentence generated by GPT-3 is grammatically correct, but some details never exist in the given table. In contrast, our proposed method, KATE, can significantly alleviate this problem by guiding GPT-3 to look for and generate the correct information. For similar reasons, large pre-trained models could be potentially susceptible to gender and racial bias. Since our KATE method shows that in-context examples are crucial for high-quality long-text generations, one way to alleviate the racial and gender bias is to incorporate an additional module to filter out offensive in-context examples. Since racial and gender bias are not our main research focus, a full investigation goes beyond the scope of our work. However, we believe this is an exciting opportunity for future work.

Code Availability

Implementations of the proposed KATE method discussed in this paper are available at https://github.com/jiachangliu/KATEGPT3.

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.

Samuel R Bowman, Gabor Angeli, Christopher Potts, and Christopher D Manning. 2015. A large annotated corpus for learning natural language inference. *arXiv preprint arXiv:1508.05326*.

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*.

Daniel Cer, Mona Diab, Eneko Agirre, Inigo Lopez-Gazpio, and Lucia Specia. 2017. Semeval-2017 task 1: Semantic textual similarity-multilingual and cross-lingual focused evaluation. *arXiv preprint arXiv:1708.00055*.

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*.

Tianyu Gao, Adam Fisch, and Danqi Chen. 2020. Making pre-trained language models better few-shot learners. *arXiv preprint arXiv:2012.15723*.

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. Roberta: 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. Sentence-bert: 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*.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *arXiv preprint arXiv:2010.15980*.

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*.

Adina Williams, Nikita Nangia, and Samuel R Bowman. 2017. A broad-coverage challenge corpus for sentence understanding through inference. *arXiv preprint arXiv:1704.05426*.

Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Rémi Louf, Morgan Funtowicz, et al. 2019. Huggingface's transformers: State-of-the-art natural language processing. *arXiv preprint arXiv:1910.03771*.

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.

Tony Z Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. *arXiv preprint arXiv:2102.09690*.

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

A An Example of In-context Learning

As shown in the illustration of Figure 3, GPT-3 is asked to translate “mountain” to its German version based on the three examples given as part of the input.

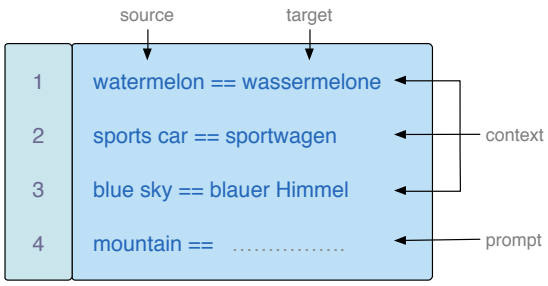

Figure 3: The figure above shows how to perform in-context learning with a language model. Three in-context examples and the test prompt are concatenated as a single string input for GPT-3, with a special character "\n" inserted between two adjacent examples. GPT-3 keeps generating tokens until there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

B Data Split

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 9: 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.

C Complete ToTTo Case Study

Due to the length limit of the main paper, we present in the appendix the full ToTTo case study comparing the random sampling baseline and our proposed KATE method. We present the case study in Table 10.

As we have discussed in the main paper, the in-context examples retrieved by KATE facilitates GPT-3 to effectively extract key information from the given table. Detailed numbers such as the number of points, rebounds, and assists have all been included in the sentence.

In contrast, the sentence generated by GPT-3 using randomly sampled in-context examples only extract partial information from the table. Only the number of points is included while the numbers of rebounds and assists are ignored. Moreover, the random sampling baseline could lead to the issue of hallucination. Both "senior year" and "University of Texas” are not present in the given table. One may wonder whether these wrong phrases were present in the randomly sampled in-context examples, which might have caused this issue. However, if we look at the randomly sampled in-context examples in the second block of the table, such information do not exist. This suggests such hallucinated phrases are generated by the language model itself.

This comparison provides some key insights on why KATE works better than the random sampling baseline. By retrieving semantically/syntactically similar in-context examples, KATE provides GPT-3 with a much more accurate template/structure to do text generation. Without such structure, GPT-3 can generate sentences that are fluent but do not meet the goal of a particular task.

D On Prompt Engineering vs. Fine-tuning

As we mentioned in the main paper, given a training dataset, we could take the full advantage of the GPT-3’s model strength through fine-tuning. However, there are several advantages of prompt engineering over fine-tuning. First, fine-tuning requires access to the model parameters and gradients. It is impossible to access this information via the current GPT-3’s API. Second, fine-tuning large models are time-consuming and costly. Ordinary research labs and individual developers do not have resources to accomplish such tasks. Third, storing large fine-tuned model checkpoints requires large storage space. Even if GPT-3 is fine-tuned and stored for many specific tasks/datasets, many fine-tuned checkpoints may not be frequently called. This is not energy efficient. Our proposed KATE method does not require costly fine-tuning and improves the random baseline on both text classification and generation tasks, sometimes by a significant margin. This makes it more practical to deploy the same GPT-3 model across all tasks.

E T5 Baseline

Although our primary goal is to improve GPT-3’s in-context learning ability, we also include the fine-tuned T5 results as a reference (3B T5 on SST-2 and

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
Randomly Sampled Examples	**Table**: <page_title >List of RAGBRAI overnight stops <section_title >By year <table ><cell > 1986 <col_header ><col_header >Year <cell >Audubon (1) <col_header >Route - start to finish (number indicates occurrence) <col_header >Monday <cell >2006 <col_header ><col_header > Year <cell >Audubon (2) <col_header >Route - start to finish (number indicates occurrence) <col_header >Monday **Sentence**: Audubon has been an RAGBRAI overnight stop in 1986 and 2006. **Table**: <page_title >List of Administrators of British Brunei <section_title >British Brunei administrators <table ><cell >Malcolm Stewart Hannibal McArthur <col_header >Consul Generals to Brunei <col_header >British Consuls in Brunei <col_header >British Residents in Brunei **Sentence**: Malcolm Stewart Hannibal McArthur was the first British resident in Brunei.
KATE-Retrieved Examples	**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.
Predictions	**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.

Table 10: A sample of retrieved in-context examples from the ToTTo dataset. 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. Information such as "senior year" and "University of Texas" also do not exist in the randomly sampled in-context examples. This suggests that the wrong information was generated by the language model itself. Although the sentence by the random sampling baseline is fluent, it does meet the goal of the table-to-text task.

ToTTo datasets, and 11B T5 on the QA datasets). The reason for reporting the 3B T5 results on the SST-2 and ToTTo datasets is that this is the largest T5 model we can use. For the 3B T5 model, Google Colab [3] provides a free V2-8 TPU to fine-tune the 3B model. We used the Colab tutorial notebook to fine-tune the 3B T5 model on the SST-2 and ToTTo training sets. We couldn't fine-tune the 11B T5 model because the model size is too large. Fine-tuning such a large model requires a V3-8 TPU, which is not free of charge. Fortunately, the original GPT-3 paper (Brown et al., 2020) has already reported the finet-tuned 11B T5 results on the three QA datasets, so we reuse these results in our main paper for the QA task. Our proposed KATE method significantly improves GPT-3, performing comparably to the fine-tuned T5 model on the table-to-text task and outperforming the fine-tuned T5 model on the QA task.

F Details on Retrieval Modules

As we mention in the main paper, we use the pre-trained RoBERTa-large model (Liu et al., 2019) as the first retrieval module, which has 355M parameters and is pre-trained with the MLM (masked language modeling) objective. The result given by this module is denoted as $\text{KATE}_{\text{roberta}}$. We directly download this model from the HuggingFace Model Zoo (MIT license) [4]. All other retrieval modules share the same architecture as the RoBERTa-large module but are fine-tuned on specific datasets.

For the fine-tuned retrieval modules, the first we use is the RoBERTa-large model fine-tuned on the SNLI and MultiNLI datasets (KATE_{nli}) (Bowman et al., 2015; Williams et al., 2017); the next we use is the RoBERTa-large model fine-tuned on the SNLI and MultiNLI dataset and then on the STS-B datasets ($\text{KATE}_{\text{nli+sts-b}}$) (Cer et al., 2017). These fine-tuned models have already been accomplished and included by the Sentence-BERT family and are publicly available, so we directly download from the Sentence-BERT Model Zoo [5].

Lastly, specifically for the sentiment analysis task, we include a RoBERTa-large model fine-tuned on the SST-2 dataset ($\text{KATE}_{\text{sst-2}}$) (Socher et al., 2013; Wang et al., 2018). At the time of our

[3] The Colab notebook on how to fine-tune the 3B T5 model can be found at https://github.com/google-research/text-to-text-transfer-transformer.

[4] The HuggingFace Model Zoo can be found at https://huggingface.co/models.

[5] The Sentence-BERT Model Zoo can be found at https://huggingface.co/sentence-transformers.

research, we didn't find a good publicly available fine-tuned model, so we fine-tune the pre-trained RoBERTa-large model on SST-2 by ourselves. The exact fine-tuning procedure, including the hyperparameters and learning rate, can be found at the HuggingFace website [6]. We fine-tune the RoBERTa-large model using a single V100 GPU.

G Prompt Templates Used

For reproducibility, we show the prompt templates used for all tasks in Tables 11 .

[6]The fine-tuning script we use can be found at https://huggingface.co/transformers/v2.7.0/examples.html#glue.

Task	Prompt Template
SST-2 & IMDB	Sentence: comes from the brave , uninhibited performances. Label: Positive Sentence: This tearful movie about a sister and her battle to save as many souls as she can is very moving. The film does well in picking up the characters and showing how Sister Helen deals with each. A wonderful journey from life to death. Label:
ToTTo	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>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 Sentence:
QA	Q: The landscape design of the Gardens of Versailles is known as which style? A: The Persian style of architecture. Q: The Mughal Gardens of Rashtrapati Bhavan is modelled on which garden? A:

Table 11: The prompt templates used for all tasks discussed in the paper. We show only one in-context example per task for illustration purposes.