What Makes In-Context Learning Work On Generative QA Tasks?

Tanushree Banerjee and **Simon Park** and **Beiqi Zou**
Princeton University
{tb21, juhyunp, bzou} @ princeton.edu

Abstract

In-context learning, introduced by Brown et al., 2020, has been a popular way to adapt very large language models to new tasks in lieu of fine-tuning them. Despite its impressive ability in improving the performance of large language models on downstream tasks, relatively little was understood about exactly why in-context learning works. Min et al., 2022c is the first attempt at empirically analyzing exactly what aspects of the in-context demonstrations contribute to improvements in downstream task performance. However, their analysis is limited to multiple choice and classification tasks, which only involve predicting a single token from a fixed set of acceptable answers. In this paper, we extend their analysis to question-answering tasks, where the model needs to generate a free response. Consistent with the findings of Min et al., 2022c, we find that the correct input-output mapping has negligible contribution, while the output space has a significant impact on the model performance.

1 Introduction

Large language models (LMs) have shown surprisingly high performance on downstream tasks by a technique called "in-context learning," proposed by Brown et al., 2020. Using this technique, an LM learns a new task via inference alone by conditioning on a concatenation of the training data as demonstrations, without any gradient updates.

Despite being the focus of significant study since its introduction, there has been relatively little work done on understanding how and why in-context learning works, and what aspects of the demonstrations provided contribute to downstream task performance. Understanding this is crucial in order to determine how to best maximize the performance gains of large language models on downstream tasks over zero-shot inference.

Min et al., 2022c is the first paper (to the best of our knowledge) that investigates why in-context learning achieves performance gains over zero-shot inference through an empirical analysis. They find that, counter-intuitively, an LM does not rely on input-label mappings in the demonstrations to perform the downstream task well, but relies more on the demonstrations to learn the output space (i.e., the set of possible labels for classification tasks) and distribution of inputs.

However, the analysis in Min et al., 2022c mainly focuses on classification and multiple choice QA tasks where the output space is relatively small. Many other NLP tasks, such as open-domain QA and summarization, have a much larger output space, where outputs cannot be "in the same format" as those in the demonstrations. Investigating what features of the demonstrations impact performance on such more open-ended downstream tasks is crucial to help us develop a more complete understanding of how and why in-context learning works, not only for such more open-ended tasks but also in general.

In this work, we aim to investigate what aspects of the demonstrations provided to an LM affect its in-context learning ability for downstream tasks that have a much larger output space that does not follow a particular format (unlike classification and multiple choice QA). In our investigation, we plan to focus on free response question-answering tasks.

2 Related work

Large language models have had a very strong performance on a diverse range of downstream NLP tasks (Devlin et al., 2019; Radford et al., 2019; Liu et al., 2019; Raffel et al., 2020; Lewis et al., 2020). In the past, fine-tuning has been the approach of choice for adapting language models to a new task (Devlin et al., 2019). However, fine-tuning a very large language model such as GPT-3 is often too expensive to be practical (Brown et al., 2020).

In-context learning, introduced by Brown et al., 2020, has been a popular way to adapt very large language models to new tasks in lieu of fine-tuning them. In-context learning involves conditioning the language model on a few pairs of input-output examples provided as additional signal at inference time without performing any gradient updates. Due to the impressive improvement in the performance of language models using in-context learning as opposed to performing zero-shot inference, in-context learning has been a popular area of research in the NLP community since its introduction in 2020.

Prior work has focused on better ways of organizing the in-context examples provided to the language model at inference time, such as formulating the problem better (Zhao et al., 2021; Holtzman et al., 2021; Min et al., 2022a), selecting better demonstrations (Liu et al., 2022; Lu et al., 2022; Rubin et al., 2022), training models with an explicit in-context learning objective (Chen et al., 2022; Min et al., 2022b). Some works have reported the over-sensitivity of the performance of the LMs on the exact format of the demonstrations, exposing the unpredictability of the in-context learning technique (Lu et al., 2022; Zhao et al., 2021; Mishra et al., 2022). This unpredictability in the performance of in-context learning indicates a lack of understanding in the field about what exactly makes in-context learning work

Despite this gap in understanding, little work has been done in the field on exactly why in-context learning performs better than zero-shot inference, and what, how, and why exactly is a large language model able to learn from demonstrations simply provided at inference time as in-context examples. There is a theoretical approach to understand in-context learning as Bayesian inference that uses demonstrations to recover latent concepts (Xie et al., 2021). The performance of in-context learning is also found to be associated with the term frequencies in the pre-training data (Razeghi et al., 2022).

Min et al., 2022c provides an empirical analysis on what aspects of the input demonstrations contribute to the improved performance of few-shot learning over zero-shot inference. Counter-intuitively, the input distribution and the output space has a more significant effect on downstream task performance than the mapping between the ground-truth outputs and inputs. However, Min et al., 2022c only discuss multiple choice and classification tasks, which only involve predicting a single token from a fixed output space.

To our best knowledge, this paper is the first approach to empirically analyze what aspects of in-context learning contributes to the model performance in open-ended tasks that involve generating multiple tokens and where there is no fixed output space. We find that the correct input-output mapping matters little. Instead, providing the correct example outputs in randomly permuted order, or choosing a random phrase within a context paragraph, if there is one, as the example output performs as well as standard few-shot learning. However, choosing random English words as the example output performed no better, if not worse, than a zero-shot setting.

3 Experimental Setup

Models. We experiment with three models, GPT-2 (124M parameters), GPT-2 Large (774M) accessed through the Huggingface API (Radford et al., 2019) and GPT-3 Curie (6.7B) accessed through the OpenAI API (Brown et al., 2020), which are all decoder-only dense models.

Datasets. We primarily focus on two datasets, SQuAD (Rajpurkar et al., 2016) and Natural Questions (Kwiatkowski et al., 2019). SQuAD is a popular question-answering dataset, where each data point contains a context paragraph, a related question, and an answer. Some questions may require the existence of the context for an answer. Natural Questions (NQ) is an open-domain question-answering dataset, where the questions are obtained from the search history of the Google search engine, and answers are annotated by humans. There are long answers (typically a paragraph) and short answers (a few words). In our experiment, we primarily use the short answers for evaluation.

Evaluation Metric. We use two metrics for evaluation: Exact Match accuracy (EM) and F1 score. Let $\mathbf{w}^{(o)} = w_1^{(o)} \cdots w_m^{(o)}$ be the output of the model and $G = \{\mathbf{w}_1^{(g)}, \cdots, \mathbf{w}_k^{(g)}\}$ be the set of acceptable gold answers. EM is computed as

$$EM(G, \mathbf{w}^{(o)}) = \mathbb{1}(\mathbf{w}^{(o)} \in G) \qquad (1)$$

Let $\mathbf{w}_i^{(g)} = w_{i,1}^{(g)} \cdots w_{i,n_i}^{(g)}$ be one acceptable gold answer. Then the *precision* is defined as $p_i = c_i/n_i$ and the *recall* is defined as $r_i = c_i/m$ where c_i is the number of words that appear both in the gold answer $\mathbf{w}_i^{(g)}$ and the output $\mathbf{w}^{(o)}$. Then the F1 score

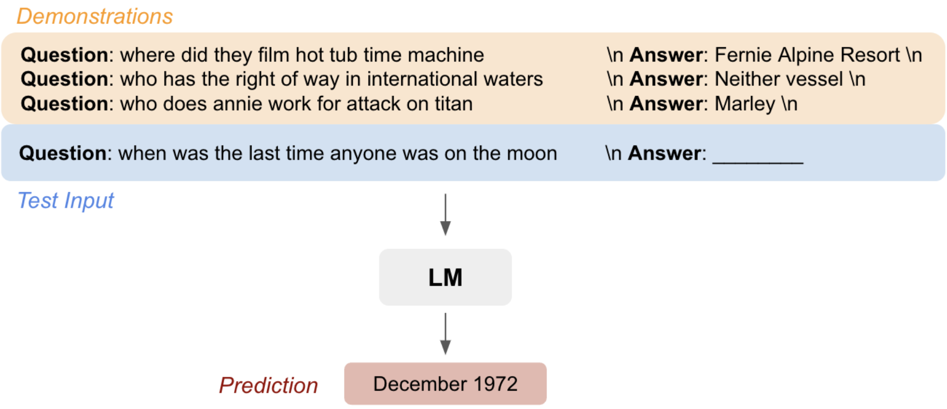
Demonstrations
Question: where did they film hot tub time machine
\n Answer: Fernie Alpine Resort \n
Question: who has the right of way in international waters
\n Answer: Neither vessel \n
Question: who does annie work for attack on titan
\n Answer: Marley \n
Question: when was the last time anyone was on the moon
\n Answer: ________
Test Input
LM
Prediction
December 1972

Figure 1: An example of in-context learning with $k = 3$ for the Natural Questions dataset. Demonstrations contain $k = 3$ question-answer pairs. We want LM to generate correct answers to the question provided in test input.

is computed as

$$F1(G, \mathbf{w}^{(o)}) = \max_{1 \leq i \leq k} \frac{2p_i r_i}{p_i + r_i} \qquad (2)$$

During evaluation, we remove all punctuation marks, articles, and whitespace from the output and groudtruth text before finally converting it all lowercase characters.

Other details. When evaluating few-shot performance, we use $k \in \{1, 2, 4, 8, 16\}$ demonstrations for GPT-2 and GPT-2 Large models and $k \in \{1, 2, 4\}$ for GPT-3 model due to compute budget. Each example is sampled uniformly at random from the entire training data. We run the experiments 3 times with different random seeds and average the performance.

4 Methodology

In Figure 1, we illustrate the process of in-context learning with an example. An LM is provided with k example input-output pairs followed by the test input and is asked to predict the test output.

Min et al., 2022c identified the following four key components of in-context learning on multiple choice and classification tasks: Format (F), Label Space (L), Input Distribution (I), Input-Label Mapping (M). Format refers to prepending each example input and output with a identifiable marker such as the word "*Question:*" or a newline character "*\n*." Label space refers to the set of acceptable output tokens that are provided as a part of the example output tokens. Input distribution refers to distribution of the example input tokens, and the input-label mapping refers to whether the gold label was provided with each example input.

For classification and multiple choice tasks, there is a fixed label space, but free response QA tasks have an open output space. However, for the case where there is a context paragraph, we still conjecture that the set of tokens present in the context serves a similar role as the output space. We devise the following experiments to test the model performance on settings where one or more of the components of the demonstrations have been perturbed. Table 1 summarizes our experiments and which component of in-context learning is being tested against for each experiment.

4.1 Modifying Example Output

Changing the example output disrupts the input-label mapping (M) and potentially also affects the label space (L). We propose the following three experiment settings:

(1) **Permute**: permute gold answers within in-context demonstrations

(2) **Random Word**: replace the example output with a random word

(3) **Random String**: replace the example output with a random string with the same length as the gold answer

When we permute the gold answers, the correct answers are still present in the context window, so we may state that the label space is partially preserved. When we choose a random word or string as the answer for NQ, we choose each word as a random vocabulary drawn uniformly at random from the set of English words. Then the label space is completely ignored. For SQuAD, we choose a

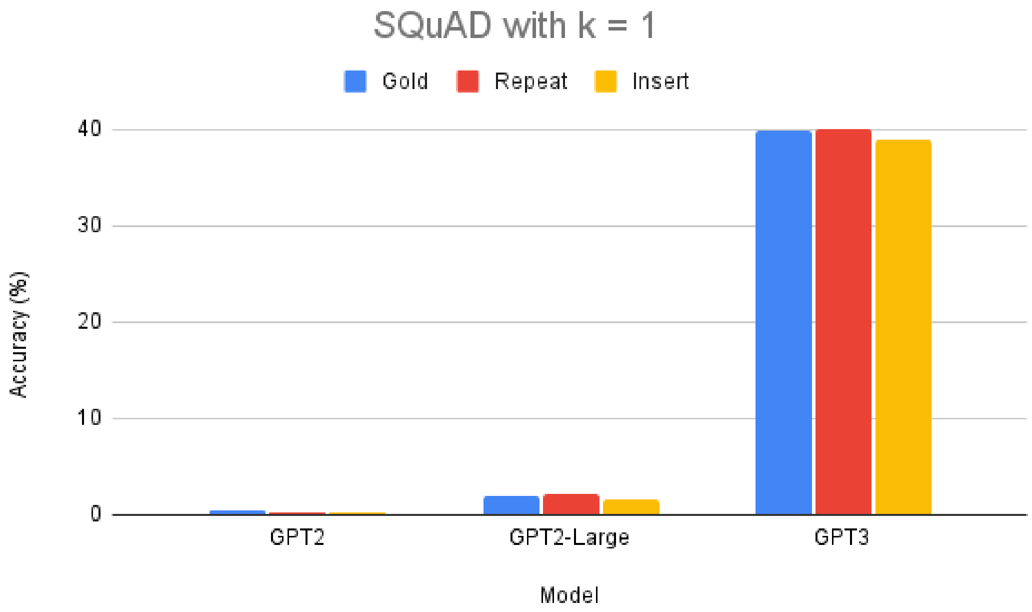

Figure 2: EM accuracy of GPT-2, GPT-2 Large, and GPT-3 on SQuAD with $k = 1$. The EM score is averaged over 3 random seeds.

random consecutive substring of the context paragraph, and we understand this setting to be partially preserving the label space.

4.2 Modifying Example Input

Changing the example input primarily affects the input distribution (I). We propose the following two experiments:

(1) **Repeat**: repeat one sentence from the context paragraph multiple times

(2) **Insert**: insert one sentence consisting of random words into the context paragraph

Both of these experiments assume the existence of a context paragraph and are applicable only for the SQuAD dataset. As some questions in the SQuAD dataset are dependent on the context, we chose to only augment the context with additional information, instead of modifying or deleting content.

5 Experiment Results

We mainly present our results on GPT-2 Large and GPT-3 models as we find the results on GPT-2 to be significantly lower than GPT-2 Large and GPT-3, with EM accuracy close to 0, as shown in Figure 2. Any trend observed from the result of this model is not expected to be statistically significant enough.

Experiment	**F**	**L**	**I**	**M**
Gold	O	O	O	O
Permute	O	△	O	X
Random Output (NQ)	O	X	O	X
Random Output (SQuAD)	O	O	O	X
Repeat	O	O	X	X
Insert	O	O	X	X
No demonstration	X	X	X	X

Table 1: Table summarizing how each proposed experiment affects each of the key components of in-context learning

5.1 Modifying Example Output

The EM scores of the GPT-3 model on NQ with $k = 4$ and on SQuAD with $k = 2$ demonstrations are reported in Figure 3. The full result of the experiments can be found in Appendix A. We summarize some of the key findings as follows:

(1) On both tasks, using demonstrations with **Gold** answers significantly improve the performance over the zero-shot setting, consistent with the findings of Min et al., 2022c.

(2) **Permute** barely hurts the performance. When using GPT-3 on NQ, we even observe slight performance gain when permuting the answers.

(3) For NQ, replacing gold answers with a **Ran-**

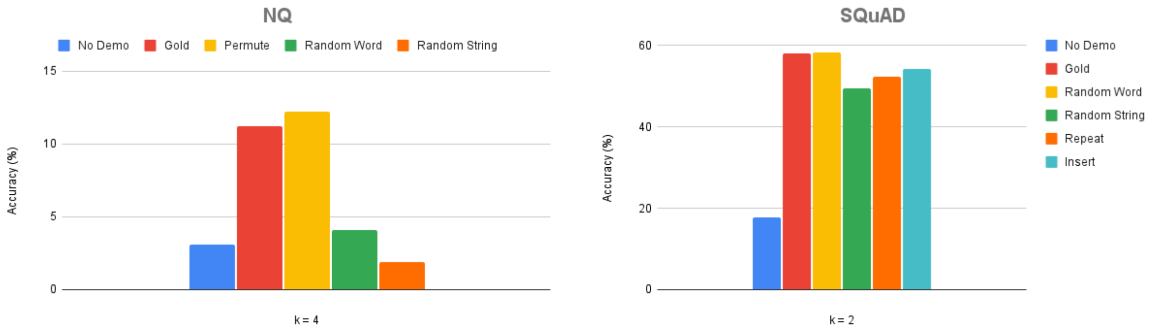

Figure 3: EM accuracy of GPT-3 model on NQ and SQuAD. The EM score is averaged over 3 random seeds.

dom word or a **Random string** drawn from the English vocabulary hurts the model, offsetting any performance boost gained from the demonstrations.

(4) For SQuAD, replacing gold answers with a **Random word** or a **Random string** drawn from the context paragraph *does not* hurt the performance.

5.2 Modifying Example Input

We summarize some of the key findings as follows:

(1) **Repeating** one of the sentences of the context paragraph, or **Inserting** a sentence of random English words both do not hurt the model performance.

(2) The model performance improves slightly more for **Insert** than **Repeat** as k, the number of demonstrations, increases.

5.3 Ablations on k

We study the impact of the number of question-answer pairs (k) in the demonstrations. Figure 4 shows the EM scores of the GPT-2 Large model on NQ and SQuAD on $k \in \{1, 2, 4, 8, 16\}$. The full result of the experiments can be found in Appendix A. We summarize some of the key findings as follows:

(1) For SQuAD, using demonstrations in any method outperforms the zero-shot setting in all cases, except when applying **Insert** on $k = 2$. In every method, the model performs best when $k = 16$.

(2) For NQ, for **Gold** and **Permute**, we observe a general performance increase when k increases. However, for **Random word** or **Random string**, the EM score does not increase and the F1 score even drops as k increases.

6 Discussion & Conclusion

In this project, we study the role of different components of demonstrations on the performance of in-context learning on open-set generative question-answering tasks. We find that the correct input-output mapping has negligible contribution, while the output space has a significant impact on the model performance. When the model had access to the gold example output, even if it was matched with a wrong example input, or when the model was given a hint that the example output should be a substring of the context paragraph, the model performed comparatively to the few-shot setting with gold outputs.

One thing to note from the results is the difference in behavior when the number of demonstration examples (k) increases. For NQ, we hypothesize that **Random word** and **Random string** introduce too many out-of-distribution texts and ultimately hurt the model performance. Although **Insert** for SQuAD also introduces out-of-distribution text, the in-distribution sentences from the context paragraph are still take the majority control when the model is asked to generate text.

6.1 Limitations and Future Work

In this paper, we have made an attempt at testing the importance of each component of in-context demonstrations on open-set generative tasks. However, the variety of the modifications we experimented on were limited due to time and budget constraints. As a follow-up, it would be worthwhile to consider some of the following experiments:

(1) Intermix some strategies. For example, permute some gold answers, while choosing a random answer for the remaining examples.

(2) For SQuAD, choose random English words

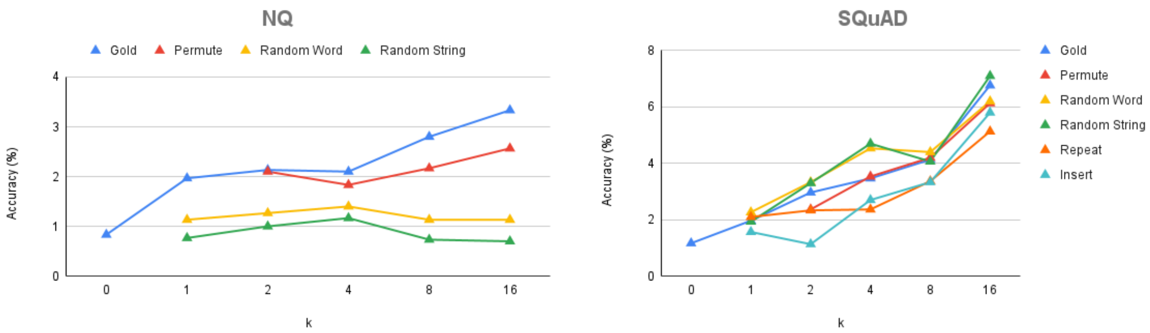

Figure 4: EM accuracy of GPT-2 Large model on NQ and SQuAD. The EM score is averaged over 3 random seeds.

instead of choosing a random sentence from the context paragraph

(3) For SQuAD, delete or modify some content from the context paragraph

(4) For each gold answer, replace it only with an answer with a similar semantical meaning (e.g., "December 1st, 1992" will be replaced with "July 4th, 1776")

(5) Experiment with a different prompt format.

References

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *CoRR*, abs/2005.14165.

Yanda Chen, Ruiqi Zhong, Sheng Zha, George Karypis, and He He. 2022. Meta-learning via language model in-context tuning. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 719–730, Dublin, Ireland. Association for Computational Linguistics.

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

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

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.

Mike Lewis, Yinhan Liu, Naman Goyal, Marjan Ghazvininejad, Abdelrahman Mohamed, Omer Levy, Veselin Stoyanov, and Luke Zettlemoyer. 2020. BART: Denoising sequence-to-sequence pre-training for natural language generation, translation, and comprehension. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 7871–7880, Online. Association for Computational Linguistics.

Jiachang Liu, Dinghan Shen, Yizhe Zhang, Bill Dolan, Lawrence Carin, and Weizhu Chen. 2022. What makes good in-context examples for GPT-3? In *Proceedings of Deep Learning Inside Out (DeeLIO 2022): The 3rd Workshop on Knowledge Extraction and Integration for Deep Learning Architectures*, pages 100–114, Dublin, Ireland and Online. Association for Computational Linguistics.

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.

Yao Lu, Max Bartolo, Alastair Moore, Sebastian Riedel, and Pontus Stenetorp. 2022. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages

8086–8098, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022a. Noisy channel language model prompting for few-shot text classification. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5316–5330, Dublin, Ireland. Association for Computational Linguistics.

Sewon Min, Mike Lewis, Luke Zettlemoyer, and Hannaneh Hajishirzi. 2022b. MetaICL: Learning to learn in context. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2791–2809, Seattle, United States. Association for Computational Linguistics.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022c. Rethinking the role of demonstrations: What makes in-context learning work?

Swaroop Mishra, Daniel Khashabi, Chitta Baral, Yejin Choi, and Hannaneh Hajishirzi. 2022. Reframing instructional prompts to GPTk's language. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 589–612, Dublin, Ireland. Association for Computational Linguistics.

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

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

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. SQuAD: 100,000+ questions for machine comprehension of text. In *Proceedings of the 2016 Conference on Empirical Methods in Natural Language Processing*, pages 2383–2392, Austin, Texas. Association for Computational Linguistics.

Yasaman Razeghi, Robert L. Logan, Matt Gardner, and Sameer Singh. 2022. Impact of pretraining term frequencies on few-shot reasoning.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2655–2671, Seattle, United States. Association for Computational Linguistics.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2021. An explanation of in-context learning as implicit bayesian inference. *CoRR*, abs/2111.02080.

Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 12697–12706. PMLR.

A Figures

In this appendix, we present the full result of the experiment results.

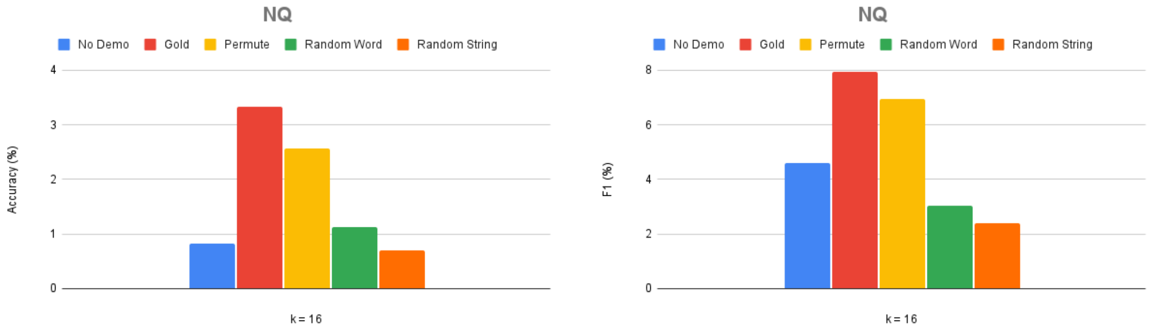

Figure 5: EM accuracy and F1 score of GPT-2 Large model on NQ. The scores are averaged over 3 random seeds.

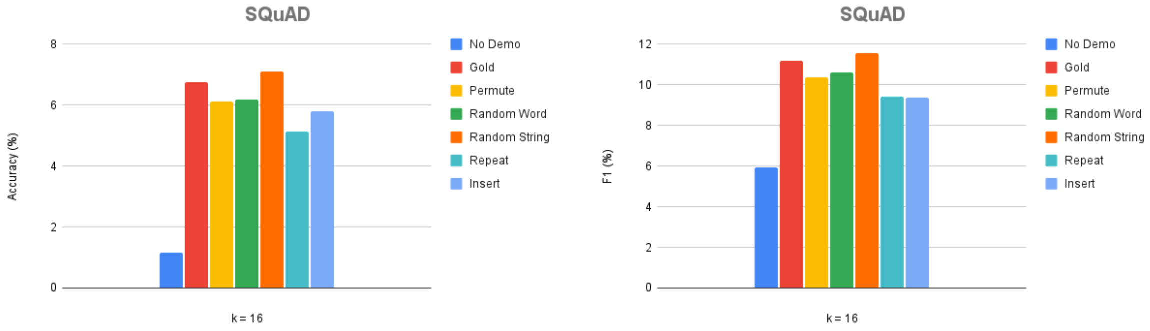

Figure 6: EM accuracy and F1 score of GPT-2 Large model on SQuAD. The scores are averaged over 3 random seeds.

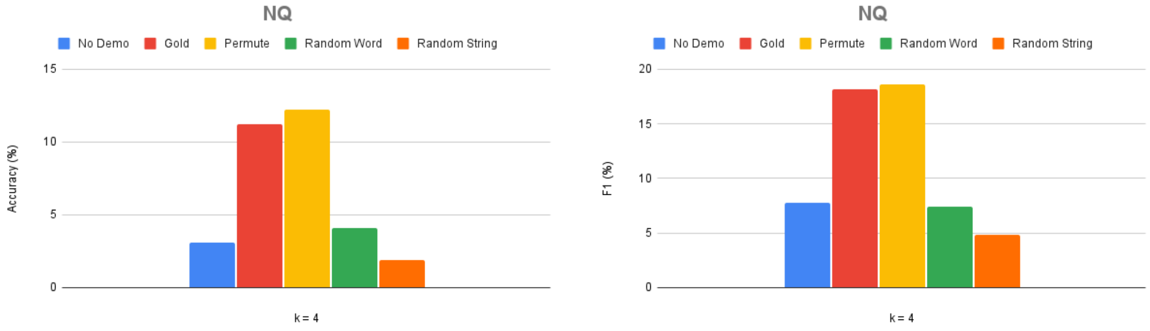

Figure 7: EM accuracy and F1 score of GPT-3 model on NQ. The scores are averaged over 3 random seeds.

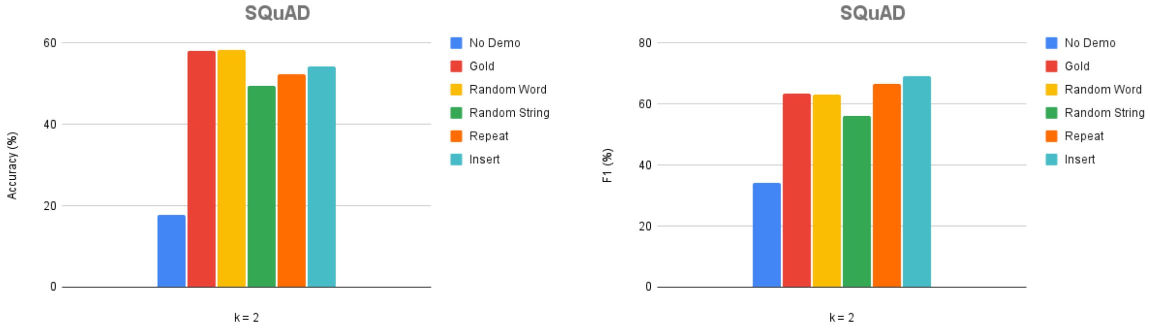

Figure 8: EM accuracy and F1 score of GPT-3 model on SQuAD. The scores are averaged over 3 random seeds.

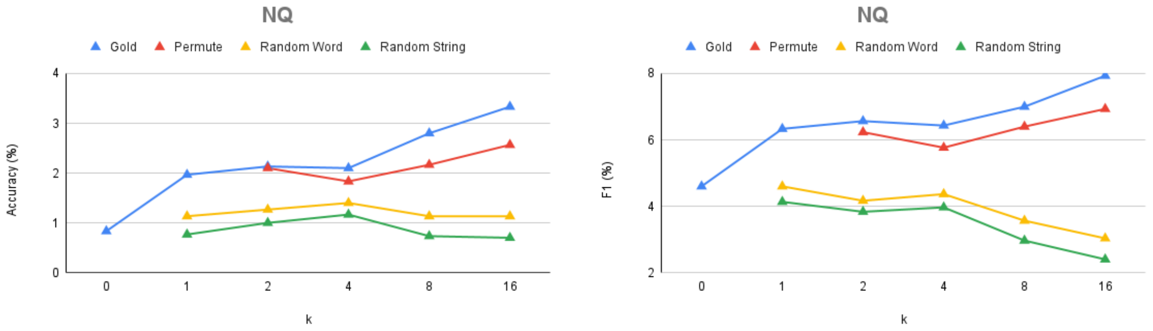

Figure 9: EM accuracy and F1 score of GPT-2 Large model on NQ. The scores are averaged over 3 random seeds.

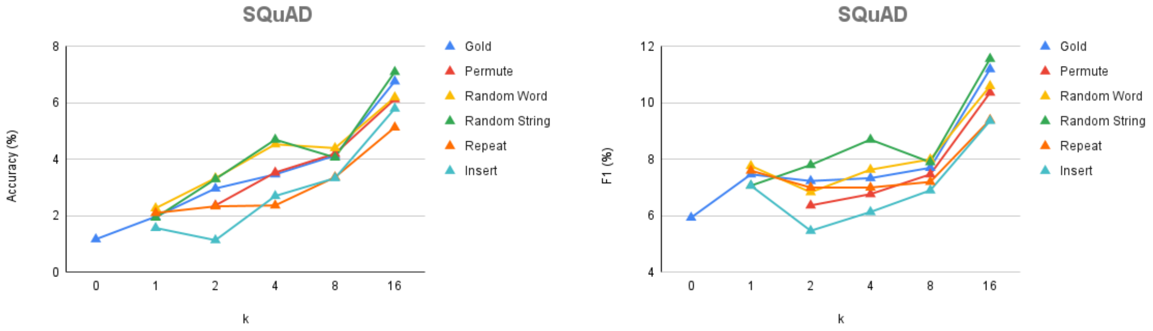

Figure 10: EM accuracy and F1 score of GPT-2 Large model on SQuAD. The scores are averaged over 3 random seeds.

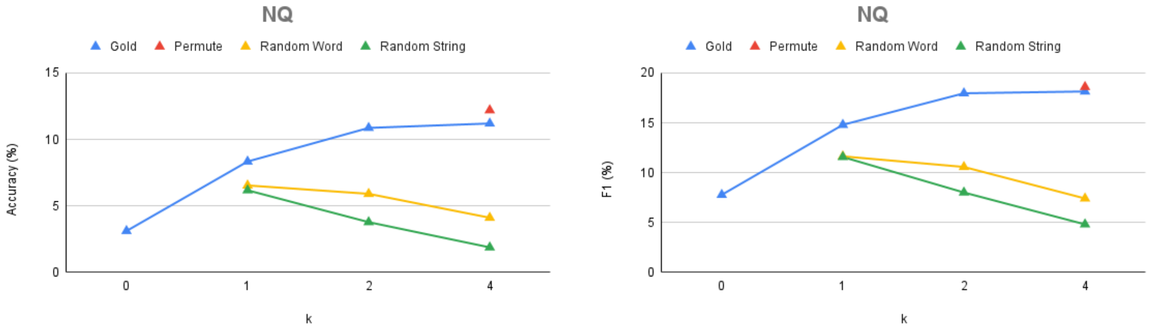

Figure 11: EM accuracy and F1 score of GPT-3 model on NQ. The scores are averaged over 3 random seeds.

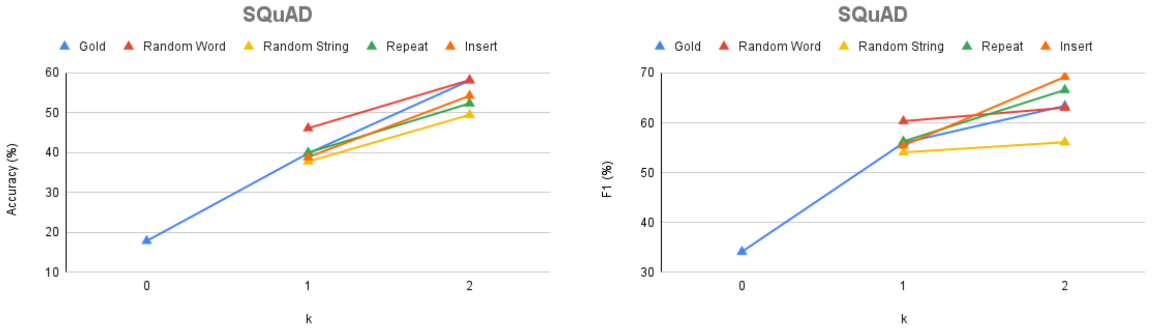

Figure 12: EM accuracy and F1 score of GPT-3 model on SQuAD. The scores are averaged over 3 random seeds.