Policy Guided Prompts: Reinforcement Learning for In-Context Example Retrieval for Large Language Models

Yair Vaknin **Daniel Mor** **Roi Cohen**
The Blavatnik School of Computer Science, Tel Aviv University
{yairvaknin,danielmor1,roi1}@mail.tau.ac.il

Abstract

The efficacy of in-context learning with large language models (LLMs) is notably influenced by the quality of chosen training examples or prompts. While recent works have explored efficient methods for prompt retrieval using annotated data and dense retrievers, we introduce a novel approach employing reinforcement learning to optimize the selection of in-context prompts. By leveraging policy gradient methods, our framework dynamically selects samples to be added to the prompt, grounded on a reward model that evaluates the LLM's performance against the ground truth. Our experimental findings demonstrate that this approach can effectively optimize prompt selection, thereby leading to enhanced outcomes in in-context learning. This work bridges reinforcement learning strategies with the emergent domain of in-context learning, paving the way for more adaptive and dynamic LLM applications. The code and related resources are available at our GitHub repository: https://github.com/PolicyGuidedPrompts/NLP_project.

1 Introduction

The recent emergence and success of pre-trained large language models (Devlin et al., 2019; Raffel et al., 2023; Yenduri et al., 2023) has ushered in a new era of in-context learning (Brown et al., 2020b), a paradigm where models are prompted with few-shot examples and generate outputs without parameter updates. This stands in contrast to the conventional approach in machine learning, which usually demands extensive training on large labeled datasets. While in-context learning presents a notable advantage, especially in scenarios where labeled data is scarce or costly, its efficacy is critically dependent on the quality of the in-context examples presented (Liu et al., 2021; Min et al., 2022). Both empirical studies and recent literature have highlighted the LLMs' sensitivity to these examples, asserting that their performance can oscillate based on the relevance and quality of these prompts.

Notably, there has been a growing interest in the realm of prompt retrieval, which entails selecting apt training examples based on certain metrics of similarity. Current strategies largely rely on predefined unsupervised metrics or dense retrievers trained on semantic similarities. However, such methodologies either restrict their focus to smaller models or overlook harnessing the intricate feedback from LLMs.

In this paper, we introduce **P**olicy **G**uided **P**rompts (PGP), an innovative approach that leverages the strengths of both aforementioned strategies. Our approach employs reinforcement learning methods to train a policy using policy gradient methods which, once trained, select the optimal examples from a given dataset to maximize the reward generated by a LLM that receives the input question concatenated with the prompt (which contains zero or more examples depending on the action the policy chose). Moreover, our approach is especially pertinent in scenarios where researchers only have black-box access to LLMs. Such an approach allows for a more cost-effective way of interfacing with massive LLMs, which continue to evolve in size and capability.

Through comprehensive evaluations on diverse NLP datasets, we demonstrate that PGP outperforms our baselines . This should help provide insights into how prompts can be optimized for various tasks.

We believe that as LLMs continue to grow and dominate the landscape of NLP, the quest for more efficient and effective interaction techniques will become paramount. Our work on PGP underscores this direction, highlighting the significance of prompt quality in the realm of in-context learning.

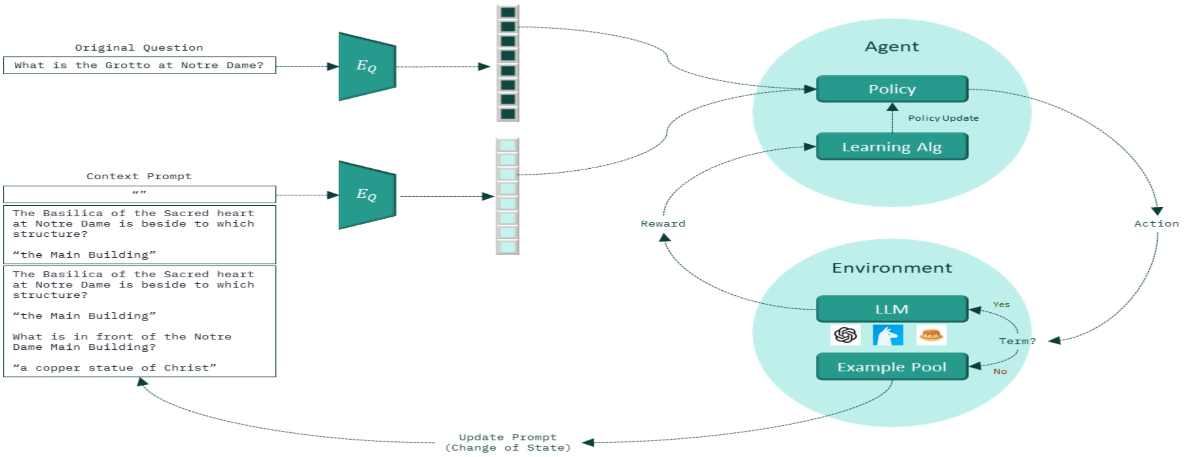
Original Question
What is the Grotto at Notre Dame?
E_Q
Context Prompt
“”
The Basilica of the Sacred heart at Notre Dame is beside to which structure?
“the Main Building”
The Basilica of the Sacred heart at Notre Dame is beside to which structure?
“the Main Building”
What is in front of the Notre Dame Main Building?
“a copper statue of Christ”
E_Q
Agent
Policy
Policy Update
Learning Alg
Reward
Action
Environment
LLM
Yes
Term?
No
Example Pool
Update Prompt
(Change of State)

Figure 1: The figure depicts the PGP architecture where a query and an initial prompt are encoded to establish the state. This state is inputted into a policy, which dictates the subsequent action: evaluating the current combination or augmenting the prompt with an additional example, thereby refining the state for the next iteration. Upon evaluating the generated answer, the policy is updated according to the received reward, fine-tuning future actions.

2 Related Work

In-context learning is a key feature of large language models (LLMs) that allows them to execute a variety of tasks based on a limited set of input-output examples, without the need for parameter adjustments or fine-tuning. This capability has been showcased in models like GPT-3 (Brown et al., 2020a), GPT-Neo (Black et al., 2021), and LLaMA (Touvron et al., 2023), drawing significant interest in the academic sphere. Research in this domain primarily revolves around deciphering the foundational mechanisms and principles behind in-context learning. For example, giving rise to the proposition that this form of learning may be akin to implicit Bayesian inference (Xie et al., 2022), or proposing that it resembles meta-optimization (Dai et al., 2023).

Another research avenue involves investigating diverse methodologies for curating and selecting in-context examples for LLMs. Recent studies (Liu et al., 2021; Rubin et al., 2022; Li et al., 2023a; Luo et al., 2023; Wang et al., 2024) have indicated that employing the BM25 algorithm or refining dense retrievers with feedback from LLMs to mine training sets can boost in-context learning efficacy. Our study aligns with this trajectory, introducing a novel training methodology for choosing in-context examples.

Dense retrieval, a prevalent technique in information retrieval, leverages dense vectors to semantically match queries and documents within a latent space (Reimers and Gurevych, 2019a; Wang et al., 2022). This method, which utilizes the robust modeling capabilities of pre-trained language models, is often preferred over sparse retrieval strategies like BM25 due to its ability to address vocabulary mismatches. Techniques such as hard negative mining (Karpukhin et al., 2020), knowledge distillation (Ren et al., 2023), and continual contrastive pretraining (Wang et al., 2022) have been developed to further refine dense retrieval's effectiveness.

Retrieval-augmented LLMs represent an integration of LLMs' generative capabilities with the proficiency to fetch pertinent information from external data sources (Ram et al., 2023; Lewis et al., 2021; Shi et al., 2023). This model not only enhances the accuracy and current relevance of the generated content but also facilitates source attribution (Nakano et al., 2022). In the context of in-context learning, the aim is to bolster LLMs' performance in downstream tasks by retrieving informative examples relevant to the task (Li et al., 2023a; Rubin et al., 2022; Luo et al., 2023).

3 Preliminaries

Let us first outline the fundamental aspects of in-context example retrieval. Consider a test example x_{test} from a specified target task and a set of k in-context examples $\{(x_i, y_i)\}_{i=1}^{k}$ drawn from a predetermined example pool $\mathcal{P}$. The task involves feeding a frozen LLM with the concatenation of the selected in-context examples and x_{test}, and using it to predict an output y'_{test} via autoregressive decoding. The core goal in this process is to select k examples from $\mathcal{P}$ in such a way that the predicted output y'_{test} closely aligns with the actual ground-truth output y_{test}, as measured by certain task-specific scores. In our study, the example pool $\mathcal{P}$ corresponds to the training dataset of each task under consideration, or some transformation of it.

Additionally, policy gradient methods are a class of algorithms in reinforcement learning focused on directly optimizing the policy that an agent follows to make decisions. Unlike value-based methods, which first estimate the value of actions and then derive a policy, policy gradient methods adjust the policy directly. Two prominent policy gradient algorithms are Vanilla Policy Gradient (VPG) (Sutton et al., 1999) and Proximal Policy Optimization (PPO) (Schulman et al., 2017).

4 Methodology

This section outlines the methodology adopted in our study, focusing on the development and implementation of the PGP framework. It is depicted in Figure 1 and outlined in detail in Algorithm 1.

Datasets: We employ a diverse collection of NLP datasets, each with a designated scoring method relevant to its task, such as F1 score for sequence alignment or exact match for boolean outcomes. The datasets encapsulate the task-specific intricacies and serve as the knowledge base from which our policy selects informative prompts. Results for the following datasets will be discussed in the following section: PAWS (Zhang et al., 2019), the RTE, MNLI, MRPC and CoLA tasks from GLUE benchmark (Wang et al., 2019), SQuAD (Rajpurkar et al., 2016) and StrategyQA (Geva et al., 2021).

Environment: Central to our approach is a reinforcement learning environment which interfaces with both the datasets and the LLM. This environment, acting as a dynamic scaffold, orchestrates the policy's interaction with the LLM. It maintains a pool of examples, which could depend on the query, from which it can draw to augment the prompts, contingent upon the policy's actions. The environment is also responsible for evaluating the LLM's outputs by measuring the reward relative to the performance on the selected in-context examples.

LLM Integration: Our framework integrates a comprehensive suite of pre-trained LLMs, including GPT-2, several FLAN T5 (Chung et al., 2022), and LLaMA models, to generate responses from the policy-crafted prompts. The output from these LLMs is evaluated to derive the reward signal necessary for policy learning. For the results discussed in this paper, we particularly utilize the FLAN T5-base model due to its proven reliability and efficient inference speed that align well with our hardware resources.

Example Pool Retrieval Mechanism: Our approach utilizes two distinct mechanisms for sourcing in-context examples. In scenarios where a retrieval model is not employed, the example pool comprises the entirety of the dataset, excluding the current query. This offers the policy an exhaustive selection, albeit without any pre-filtering. Conversely, when employing a retrieval model, we utilize a Sentence-BERT model (Reimers and Gurevych, 2019b), which encodes the questions and retrieves a top-k subset of contextually relevant examples for our given query. This subset forms a more manageable and pertinent pool from which the policy can select examples to optimize the prompt's effectiveness.

State Encoding: The state representation within our reinforcement learning framework is crucial for capturing the context necessary for decision-making. We first encode the query and the context prompt separately, preserving the distinct semantics of each. These encoded representations are then concatenated to form a comprehensive state encoding. This approach ensures that the unique information of the query is maintained and meaningfully integrated with the context of the prompt. While our primary results utilize Sentence-BERT for encoding due to its balance of performance and efficiency, our framework is also compatible with other models such as BGE (Xiao et al., 2023), GTE (Li et al., 2023b) and BERT.

Reward Model: Our framework employs a reward model tailored to the specific characteristics of each task. For tasks where sequence alignment is needed, we utilize an F1 score-based reward system, providing a reward between -1 and 1. This scoring allows for a nuanced evaluation of the generated responses in terms of precision and recall. Conversely, for tasks requiring boolean or few discrete outcomes, we adopt an exact match scoring system, where the reward is binary: 1 for a correct match and -1 for an incorrect one. Crucially, rewards are only dispensed upon prompt evaluation at the end of an episode. Immediate rewards for adding an example to a prompt are set to 0, indicating that the effectiveness of such an action is only propagated through the calculation of returns at the

Algorithm 1 Adapted Vanilla Policy Gradient for Prompt Generation

Input: Dataset $\mathcal{D}$

Output: Policy π

```
Initialize policy parameters $\theta$, baseline values $b(s)$
for iteration $= 1, 2, \ldots,$ num_batches do
    for episode $= 1, 2, \ldots,$ num_episodes_per_batch do
        Reset the environment to obtain initial state $s_0$ and empty prompt
        Retrieve initial question $q$ and ground truth answer $y$
        $P \leftarrow$ ExamplePool($q$, dataset)
        for each time step $t$ within the episode do
            Observe the current state $s_t$ consisting of $q$ and current prompt
            Choose an action $a_t$ based on policy $\pi_\theta(s_t)$
            if action $a_t$ is not terminate_action then
                $example \leftarrow P[a_t]$
                Update the prompt with $example$
            else if action $a_t$ is terminate_action or the prompt is too long then
                Concatenate the final prompt with $q$ and send to the LLM to generate a response $\hat{y}$
                $r_t \leftarrow$ Evaluate($y, \hat{y}$)
                End the episode and proceed to advantage and policy update steps
            Compute the return $G_t^{(i)}$ from time $t$ to the end of the episode
            Compute the advantage estimate $\hat{A}_t^{(i)} = G_t^{(i)} - b(s_t)$
        Re-fit the baseline to the empirical returns by minimizing $\sum_{i=1}^{m} \sum_{t=0}^{T-1} \left| b(s_t) - G_t^{(i)} \right|^2$
    Update policy parameters $\theta$ using the policy gradient estimate $\hat{g}$:
    $\hat{g} = \sum_{i=1}^{m} \sum_{t=0}^{T-1} \hat{A}_t^{(i)} \nabla_\theta \log \pi_\theta(a_t^{(i)} | s_t^{(i)})$ with an optimizer
```

episode's conclusion. This approach ensures that the policy learns the long-term impact of its actions, aligning with the overall objective of optimizing prompt selection for improved LLM responses.

Policy Modeling: Similarly to DQN (Mnih et al., 2013), we use a deep network to model our policy. Here, we use a Multi-Layer Perceptron (MLP). The MLP's architecture is designed to efficiently process the concatenated encodings of the query and the context prompt, providing a robust mechanism for policy decisions. The output of the MLP is a probability distribution over potential actions, including adding specific examples to the prompt or terminating the augmentation process.

Baseline Network: To stabilize learning and reduce variance, we employ a baseline Network that provides a baseline value for the rewards, against which the advantages of actions are calculated. This network is updated in accordance with the observed rewards. Similarly to our policy architecture, this Baseline Network is constructed as an MLP, as well, that processes the encoded state representations. It is continuously updated in alignment with the observed rewards to ensure the reliability of the advantage estimations.

Training Procedure: At the core of PGP lies an adapted Vanilla Policy Gradient algorithm[1]. Our training approach focuses on optimizing the policy network through backpropagation using the ADAM optimizer (Kingma and Ba, 2017). We leverage the calculated advantages and the gradients of the policy during this process. Each training episode represents a complete cycle of prompt generation and evaluation, enabling the policy to learn from the outcomes of its actions. The objective is to train the network to predict actions (example selections) that maximize rewards, thereby iteratively refining the quality of prompts. This process unfolds over numerous episodes, with the cumulative reward across these episodes serving as the primary indicator of policy performance and improvement. To facilitate effective learning, we employ exploration versus exploitation strategies, including ϵ-greedy and temperature-based methods, where the selection randomness is adjusted over time (both ϵ and temperature values decay over time). These heuristics are crucial for balancing the trade-off between exploring new possibilities and exploiting known rewarding strategies, enabling the policy to converge to optimal actions. Empirically, the ϵ-greedy approach showed better results for our project, and therefore, we will present our findings using this method. Notable hyperparameters that were fine-tuned during training include the initial learning rate for the optimizer, the number of layers in the policy and baseline networks, the initial ϵ value, and the top K retrieved examples (when using a retriever).

5 Experimental Results

In this section, we present the experimental outcomes of applying the PGP method across various tasks. The scores in the table represent the average rewards obtained when testing according to the task-specific reward model, reflecting the efficiency of each prompting strategy in generating accurate responses.

- **Zero-Shot:** This strategy involves prompting the LLM without any in-context examples, relying solely on the query.
- **Random Few-Shot:** This approach selects a few examples at random from the dataset to form the prompt, without considering their relevance to the query.
- **Top-1 and Top-3:** These strategies use the SBERT model to select the top 1 or top 3 most similar examples from the dataset, respectively, based on semantic similarity to the query, and add them to the prompt.

As mentioned, we evaluate two variants of our approach: **PGP-No-Retrieval** and **PGP-Top-K-Retrieval**. The results, as shown in Table 1, highlight the efficacy of PGP in enhancing the quality of in-context learning with large language models.

Furthermore, during our experiments with the SQuAD and StrategyQA datasets, we encountered a notable phenomenon that, while not leading to a significant improvement in accuracy, offers valuable insights into the behavior of our policy model. Specifically, when provided with the questions and their added context from SQuAD or the questions and their added facts from StrategyQA, the policy learned a rather intuitive strategy. It discerned that the most effective action in these cases was to terminate the episode immediately and forward the query directly to the LLM. Although this did not translate into a marked improvement in accuracy, since it is effectively the same as the zero-shot set-up, it is a compelling instance of the policy adapting to the nuanced requirements of different datasets and optimizing the prompt-selection process accordingly.

6 Analysis and Conclusions

The experimental results underscore the effectiveness of PGP across various NLP tasks, showcasing its nuanced advantage in in-context learning.

PGP vs. Conventional Strategies: PGP generally outperforms the Zero-Shot strategy across the above mentioned tasks, highlighting the importance of contextually relevant examples in enhancing LLM performance. The significant improvements observed in the GLUE-MNLI and GLUE-MRPC tasks exemplify PGP's capability in handling complex and diverse prompts. PGP also displays superior results to both the Top-1 and Top-3 retrieval methods. This observation suggests that selecting the semantically closest examples is not always the most effective strategy. The PGP approach, by understanding and exploiting the underlying statistics of the data and the retrieval model, can discern more contextually appropriate examples to enhance the prompt, rather than solely relying on semantic proximity.

Robustness of PGP: In tasks like GLUE-CoLA, where adding additional prompts appears to be detrimental, PGP demonstrates a unique robustness. Unlike other methods where adding prompts consistently hampers performance, PGP minimizes this negative impact. This indicates that PGP, while integrating additional context, does so in a more discerning and effective manner, leading to less harm compared to other prompt-adding methods.

No-Retrieval vs. Top-K-Retrieval: The comparison between PGP with and without retrieval illustrates the adaptability of our method. The choice between these two variants offers flexibility depending on task-specific requirements and available computational resources. In the PGP-Top-K-Retrieval variant, our policy learns to choose from the top 50 closest neighbors, as this number yielded the best results in our experiments. The policy adeptly selects the best candidates from these neighbors based on the input query, indicating a nuanced understanding that some queries may benefit from more similar candidates while others may need less similar ones. While still providing effective results, this task appears to be more challenging, and as indicated in Table 1, generally the No-Retrieval approach is preferred. This preference underscores the complexity of utilizing a retrieval model within the prompting process in such a way.

Insights from SQuAD and StrategyQA: Our analysis of PGP's performance on the SQuAD and StrategyQA datasets unveiled a noteworthy pattern. Contrary to other datasets, PGP chose not to augment the prompts with additional examples. It relied instead on the rich context inherently accompanying each question in these datasets. This decision stems from the realization that the provided context or facts within these datasets are sufficient for the LLM to generate informed responses, rendering the addition of further examples redundant, and potentially harmful, as it dilutes the context of the target query. In stark contrast, our observations in the CoLA dataset showed that PGP added prompts due to the absence of such rich contextual information. However, this strategy often led to

	Zero-Shot	Random-Few-Shot	Top-1	Top-3	PGP-No-Retrieval	PGP-Top-K-Retrieval
GLUE-MNLI	-0.818	-0.078	-0.7	-0.618	**0.29**	-0.056
GLUE-MRPC	0.409	0.458	0.407	0.421	**0.505**	0.464
GLUE-CoLA	**0.263**	0.016	0.142	0.157	0.254	0.125
GLUE-RTE	0.097	0.085	0.084	0.087	0.094	**0.153**
PAWS	0.777	0.815	0.63	0.63	**0.818**	0.791

Table 1: Comparison of performance across different prompting strategies for various NLP tasks.

a decline in performance, indicating that prompt augmentation may not always be beneficial. We conjecture that PGP still chose to add prompts in the case of CoLA, since it wasn't as clear cut. That is, adding prompts did in fact increase the reward for some of the episodes. These findings underscore PGP's ability to adapt to the specific needs and nuances of different datasets.

PPO Performance: Our experiments included attempts with PPO, as well, but it yielded suboptimal results in this context. One plausible explanation lies in PPO's clipped advantage mechanism. Designed to prevent large policy updates, this mechanism might have limited the algorithm's ability to adapt effectively in our setting. Given that optimal prompt selection might require notable shifts in policy, the conservative nature of PPO's updates may not be fully compatible with the dynamic requirements of our task.

Testing on Stronger Models: Due to time constraints, our exploration with larger models was limited. However, preliminary tests with more powerful models - namely the large, XL, and XXL variants of FLAN T5 - indicated similar trends of improvement as observed with smaller models. Notably, these larger models inherently demonstrated better baseline performances, attributed to their advanced capabilities. While comprehensive testing with these models was beyond our current scope, initial results suggest that the observed trends are consistent across different model scales.

7 Future Work

The research presented in this paper opens several avenues for future exploration and development. Key areas of interest include:

- **Training Custom Encoders:** Instead of utilizing pre-trained, frozen encoders, future work could involve training custom encoders specifically tailored to the task at hand. This approach would allow for more flexibility and potentially greater effectiveness in prompt selection. Additionally, considering separate encoders for processing the query and the prompt could further optimize the state representation.

- **Multi-Task Training:** Another promising direction is the simultaneous training of the policy model across multiple tasks. This multitask approach could enhance the generalizability and robustness of the policy, enabling it to adapt to a wider range of scenarios and datasets with varying characteristics.

- **Employing Stronger LLMs:** While our preliminary tests with more powerful LLMs have shown promising trends, a more extensive exploration with these models could yield significant insights. Future research could involve a comprehensive evaluation using state-of-the-art LLMs, which could potentially amplify the benefits observed in our current study.

Pursuing these avenues presents exciting and interesting research opportunities, potentially yielding substantial improvements in the application of PGP.

References

Sid Black, Leo Gao, Phil Wang, Connor Leahy, and Stella Biderman. 2021. GPT-Neo: Large Scale Autoregressive Language Modeling with Mesh-Tensorflow.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D 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 Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020a. Language models are few-shot learners. In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc.

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

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, Albert Webson, Shixiang Shane Gu, Zhuyun Dai, Mirac Suzgun, Xinyun Chen, Aakanksha Chowdhery, Sharan Narang, Gaurav Mishra, Adams Yu, Vincent Zhao, Yanping Huang, Andrew Dai, Hongkun Yu, Slav Petrov, Ed H. Chi, Jeff Dean, Jacob Devlin, Adam Roberts, Denny Zhou, Quoc V. Le, and Jason Wei. 2022. Scaling instruction-finetuned language models.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can gpt learn in-context? language models implicitly perform gradient descent as meta-optimizers.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. Bert: Pre-training of deep bidirectional transformers for language understanding.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. Did aristotle use a laptop? a question answering benchmark with implicit reasoning strategies.

Vladimir Karpukhin, Barlas Oğuz, Sewon Min, Patrick Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen tau Yih. 2020. Dense passage retrieval for open-domain question answering.

Diederik P. Kingma and Jimmy Ba. 2017. Adam: A method for stochastic optimization.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Xiaonan Li, Kai Lv, Hang Yan, Tianyang Lin, Wei Zhu, Yuan Ni, Guotong Xie, Xiaoling Wang, and Xipeng Qiu. 2023a. Unified demonstration retriever for in-context learning.

Zehan Li, Xin Zhang, Yanzhao Zhang, Dingkun Long, Pengjun Xie, and Meishan Zhang. 2023b. Towards general text embeddings with multi-stage contrastive learning.

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

Man Luo, Xin Xu, Zhuyun Dai, Panupong Pasupat, Mehran Kazemi, Chitta Baral, Vaiva Imbrasaite, and Vincent Y Zhao. 2023. Dr.icl: Demonstration-retrieved in-context learning.

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

Volodymyr Mnih, Koray Kavukcuoglu, David Silver, Alex Graves, Ioannis Antonoglou, Daan Wierstra, and Martin Riedmiller. 2013. Playing atari with deep reinforcement learning.

Reiichiro Nakano, Jacob Hilton, Suchir Balaji, Jeff Wu, Long Ouyang, Christina Kim, Christopher Hesse, Shantanu Jain, Vineet Kosaraju, William Saunders, Xu Jiang, Karl Cobbe, Tyna Eloundou, Gretchen Krueger, Kevin Button, Matthew Knight, Benjamin Chess, and John Schulman. 2022. Webgpt: Browser-assisted question-answering with human feedback.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

Pranav Rajpurkar, Jian Zhang, Konstantin Lopyrev, and Percy Liang. 2016. Squad: 100,000+ questions for machine comprehension of text.

Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. 2023. In-context retrieval-augmented language models.

Nils Reimers and Iryna Gurevych. 2019a. Sentence-bert: Sentence embeddings using siamese bert-networks.

Nils Reimers and Iryna Gurevych. 2019b. Sentence-BERT: Sentence embeddings using Siamese BERT-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3982–3992, Hong Kong, China. Association for Computational Linguistics.

Ruiyang Ren, Yingqi Qu, Jing Liu, Wayne Xin Zhao, Qiaoqiao She, Hua Wu, Haifeng Wang, and Ji-Rong Wen. 2023. Rocketqav2: A joint training method for dense passage retrieval and passage re-ranking.

Ohad Rubin, Jonathan Herzig, and Jonathan Berant. 2022. Learning to retrieve prompts for in-context learning.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen tau Yih. 2023. Replug: Retrieval-augmented black-box language models.

Richard S. Sutton, David McAllester, Satinder Singh, and Yishay Mansour. 1999. Policy gradient methods for reinforcement learning with function approximation. In *Proceedings of the 12th International Conference on Neural Information Processing Systems*, NIPS'99, page 1057–1063, Cambridge, MA, USA. MIT Press.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. Llama: Open and efficient foundation language models.

Alex Wang, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel R. Bowman. 2019. Glue: A multi-task benchmark and analysis platform for natural language understanding.

Liang Wang, Nan Yang, Xiaolong Huang, Binxing Jiao, Linjun Yang, Daxin Jiang, Rangan Majumder, and Furu Wei. 2022. Text embeddings by weakly-supervised contrastive pre-training.

Liang Wang, Nan Yang, and Furu Wei. 2024. Learning to retrieve in-context examples for large language models.

Shitao Xiao, Zheng Liu, Peitian Zhang, and Niklas Muennighoff. 2023. C-pack: Packaged resources to advance general chinese embedding.

Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference.

Gokul Yenduri, Ramalingam M, Chemmalar Selvi G, Supriya Y, Gautam Srivastava, Praveen Kumar Reddy Maddikunta, Deepti Raj G, Rutvij H Jhaveri, Prabadevi B, Weizheng Wang, Athanasios V. Vasilakos, and Thippa Reddy Gadekallu. 2023. Generative pre-trained transformer: A comprehensive review on enabling technologies, potential applications, emerging challenges, and future directions.

Yuan Zhang, Jason Baldridge, and Luheng He. 2019. Paws: Paraphrase adversaries from word scrambling.