
MINI-BATCH IN-CONTEXT LEARNING FOR LLM SELF SELECTION

William Morgan, Ji Heon Kim, Jason Chen, Ishan Tripathi

Department of Computer Science
University of Maryland
College Park, MD 21532, USA

ABSTRACT

Large Language Models (LLM) like GPT3 (Tom B. Brown (2020)) demonstrate a novel emergent capability known as in-context learning. In-context learning (ICL) operates by providing LMs with demo examples of input-output pairs for a given task. Our work involves exploring the best prompt design for the LLM itself to select and summarize these in-context demo examples in mini-batches, enabling it to optimize its own output. Our experiment uses Meta’s Llama2-7b LLM and the ss2 Stanford Sentiment Treebank database, which contains 67.3k rows of movie reviews and their positive or negative labels. We asked the LLM to determine which of its given database examples are most related and relevant to comedy, changing the structure and wording of the prompt each time. Finally, we rated how effectively the LLM was able to select and summarize demo examples based on each of our specifically engineered prompts.

1 INTRODUCTION

In recent developments within the realm of Large Language Models (LLMs), a major surge in natural language processing has occurred. However, despite advancements, LLMs face major challenges that can hinder their performance. Some challenges apparent in today’s models are the risk of unintended connections and interference between entities when attempting to utilize larger amounts of in-context demo examples, as well as the difficulty in handling very long input prompts (limited maximum context tokens length). These challenges limit LLMs’ capabilities, compromising contextual understanding and lowering efficiency.

Using a mini-batch approach is one way to deal with these problems. This innovative solution minimizes the inaccuracies caused by interference, by processing in-context demo examples in subsets, smaller groups, or “batches,” hence the name. Doing so enhances the robustness and ability of LLMs to handle larger amounts of demos in natural language processing tasks.

The LLM input length limit, as encountered in models like GPT-3, refers to the inherent challenge posed by constraints on the number of tokens these models can process in a single input. This limitation becomes particularly problematic when dealing with very long input prompts with many ICL examples, as it necessitates strategies to accommodate or truncate the input to fit within the model’s token constraints. Handling excessively long inputs can compromise the model’s contextual understanding, as it may struggle to maintain coherence and grasp the full context of extended text. This limitation is not especially pertinent during inference, but also during the training phase, where longer inputs contribute to heightened computational demands and increased memory requirements. Therefore, effective input management and preprocessing strategies are crucial, involving thoughtful consideration of how to balance the need for sufficient context with the model’s inherent token limits to optimize both efficiency and contextual understanding. Although our paper does not deal with the training phase of LLMs, the mini-batch ICL technique could find use in that area as well, thus making use of mini-batches is useful in both contexts of training and especially inference.

A further problem is the issue of prepending increasing numbers of ICL demo examples to a prompt, which incurs the risk of unintended connections and interference between entities across different examples. When multiple examples are provided in the same input, the model might attempt to relate or connect entities and context from one example to another. While this inter-example connectivity

can sometimes lead to positive correlations, such as recognizing patterns or themes, it often results in negative interference and a degradation of final performance.

In leveraging Large Language Models (LLM) for contextual learning, the key lies in designing prompts to select and summarize the most relevant context and examples for addressing input questions. The objective is to capitalize on the In-Context Learning (ICL) approach (Dong et al. (2023)). Utilizing a Mini-Batch approach helps enhance this capability, enabling virtually unlimited ICL examples to be used by mini-batching them in smaller groups through the LLM. This involves partitioning all examples into distinct mini-batches, we experiment with 5-10 ICL examples per batch, thereby adopting a mini-batch in-context learning methodology. The design includes a memory mechanism to accumulate information extracted from each mini-batch of examples for a test sample, forwarding the output of the language model to subsequent steps. We label this: *summarization from last mini - batch*.

Conceptually, this process draws parallels to recurrent neural networks (RNNs), treating the LLM as an RNN cell where each output serves as the input for the next step. This sequential mechanism enables the accumulation of knowledge from all preceding mini-batches.

We demonstrate that using mini batches along with a specific prompt order results in more accurate in-context learning. Our baselines are the previous demo results and the results from arranging the prompts in different combinations of question (Q), context (E), and (P) (previous summarization/mini-batch output). Our results show that LLM's are capable of conducting mini-batches in context learning and that it does indeed alleviate many of the concerns surrounding input length, as well as overtraining and creation of unintended links between topics.

2 RELATED WORK

One related paper is What Makes Good In-Context Examples for GPT-3 (Jiachang Liu (2021)). This paper addresses the challenge of optimizing GPT-3's performance without resorting to computationally expensive fine-tuning. Instead, the authors aim to enhance the model's few-shot capabilities by strategically selecting in-context examples, emphasizing the importance of targeted choices over random sampling. Like our approach, this paper seeks to avoid fine-tuning, and use targeted examples instead of using random samples.

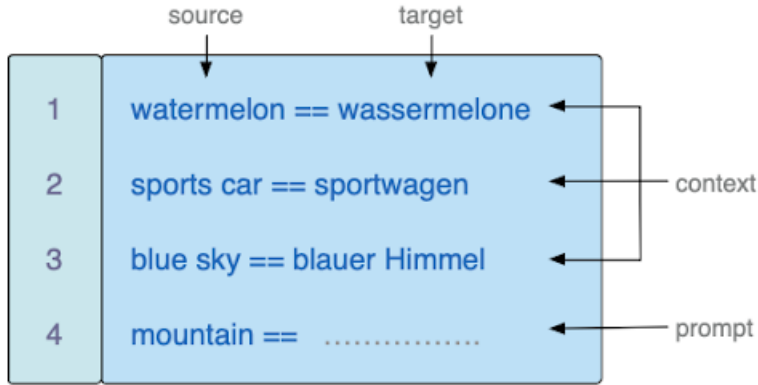


Figure 1: Context has sources and targets which gives instructions to our LLM for predicting the outcome of the task given by the prompt

However, our goal is to have the model remember useful context from each mini batch. Additionally, this paper uses a k nearest neighbor approach so it will just draw parallels to the closest prompts it was trained with, whereas we are employing a genuine language model.

3 APPROACH

Our research experiments and enhances the in-context learning (ICL) capabilities of large language models (LLMs) such as GPT-3, Bing, Bard, etc. The inherent functionality of ICL may enable these models to learn from provided inputs such as demonstrations, which can be used for its responses to be optimized for specific tasks. During the process of optimizing with specific tasks, we first apply context selection and summarization ordering to the prompts that we provide for our LLM in order to get the best context output.

Our approach centers around improving the innate ability of LLMs to comprehend and summarize contextual information. To get more effective context selection and summarization, we experiment with mini-batches on top of using different prompt structures. Mini-batches help the LLM to select and memorize useful or related contexts within each mini-batch. This approach not only mitigates the limitation of the maximum input length of transformer models but also allows for a better understanding of context relevance.

Two main factors for creating our prompts are: Contextual Prompt: Creating prompts that explicitly ask the model to utilize context and summarization capabilities. Prompts are structured to inquire about related contexts from previous mini-batches and specific summarization of context relevant to the input query. Prompt Variations: Exploring different prompt variations by altering the order of information and utilizing diverse vocabularies and phrasing. This exploration helps us to mitigate positional biases and estimate the impact of language on the model’s contextual selection.

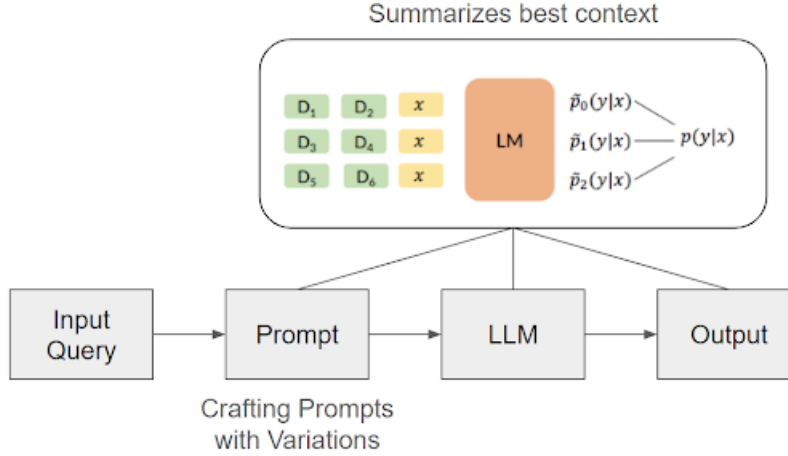


Figure 2: Creates contextual prompts with variations and then send them with mini-batches to our LLM to generate the best context output(s).

We aim to optimize the LLM’s ability to select and summarize context relevant to the input query, thereby improving the overall performance of context-driven tasks. In order for our LLM to produce efficient and accurate outputs, we design a set of diverse prompts and apply them within controlled experimental settings across various language tasks. With these crafted prompts, we use mini-batches to help the LLM learn to assess the relevance of different example and contexts within the training data. By reviewing the outputs generated by the language model in response to these prompts, we aim to quantify and analyze the efficacy of different prompt structures in generating more useful and efficient outputs with this approach.

4 EVALUATION

4.1 GOALS OF EVALUATION

In this section, we outline the specific goals of our evaluation. Our primary claim is that LLMs are capable and effective for selecting and summarizing in-context examples, rivaling other selection or

retrieval methods, such as a BM25 function (Ram et al., 2023). We designed our experiment to prove this and to discover the prompt input that most effectively selects and summarizes examples for a large language model. We claim there is a possibility that the LLM is well-suited to the task of self-selection. Its own selections may produce its own best outputs, a mechanism possibly explaining this potential phenomenon is the LLM making connections from the ICL demo examples to its own knowledge base.

4.2 EXPERIMENT DESIGN

To explore this possibility, we have designed a comprehensive set of experiments. We perform several demos with Meta’s Llama2 LLM on the sst2 dataset. In each demo, we implement our approach, varying the wording and ordering of the prompts. We then analyze the resulting output via human scoring, to see what kind of prompts make the output better in humans’ eyes.

Baselines: Initial prompt outputs, without much thought into prompt engineering

Datasets: sst2, Stanford Sentiment Treebank (Pang (2005))

Evaluation Metrics: Human ranking of prompt output on a scale from 0-10

We started with this template prompt idea:

My question will be: question, and here are some examples/context: last mini-batch output and examples, summarize the examples/context you think is related to my question.

Altering it for Llama2 thusly:

<s>[INST] «SYS» You are a precise AI model. You select and return the demo examples that are most related and relevant to the task «/SYS» Given the examples, classify the review with a single word positive or negative. Here are the examples: examples Here is the review which you need to classify with a single word positive or negative: question Positive or negative: This review is [/INST]

After iterating on our prompt, tweaking the wording, vocabulary used, and ordering, we ended our experiment with this prompt, which performed with the best results out of 12 different iterations.

<s>[INST] «SYS» You are a precise, succinct AI model. You select and return the demo examples that are most related and relevant to comedy. «/SYS» Here are the demo examples: "memo", "examples", "Select and return only the above demo examples that are most related and relevant to comedy. Match the same formatting the examples were given in, using Review: and Positive or negative: ." [/INST]

We chose to question the LLM about comedy specifically, to generate selections and summarizations from the sst2 dataset. (Keywords “Comedy,” “Funny,” and “Humor,” make up around 4-5 percent of the sst2 dataset)

Our first mini-batch baseline output, which we ranked with an average score of 2/10.

Output:

- 1. Review: deliciously mean-spirited and wryly observant. Positive or negative: positive*
- 2. Review: cute factor. Positive or negative: positive*
- 3. Review: again that the era of the intelligent, well-made b movie is long gone. Positive or negative: negative*

4.3 EXPERIMENT RESULTS

Our experiment results have revealed key ideas and concepts in prompt tuning/engineering for LLM self-selection of ICL examples. With the best prompt structure and mini-batch configurations, the prompt 6h-summ-before provided the best results.

6h-summ-before

<s>[INST] «SYS» You are a precise, succinct AI model. You select and return the demo examples that are most related and relevant to comedy. «/SYS» Here are the demo examples: "memo", "examples", "Select and return only the above demo examples that are most related and relevant

to comedy. Match the same formatting the examples were given in, using Review: and Positive or negative: ."[/INST]

Output

1. Review: *deliciously mean-spirited and wryly observant*

Positive or negative: *positive*

2. Review: *a romantic comedy plotline straight from the ages*

Positive or negative: *positive*

3. Review: *to hold onto what 's left of his passe ' chopsocky glory*

Positive or negative: *negative*

4. Review: *'ll swear that you 've seen it all before , even if you 've never come within a mile of the longest yard .*

Positive or negative: *negative*

5. Review: *, colorful , semimusical*

Positive or negative: *positive*

6. Review: *evacuations , and none-too-funny commentary on the cultural distinctions between americans and brits*

Positive or negative: *negative*

These examples use terms associated with comedic writing, such as "mean-spirited," "wryly observant," "romantic comedy," "passe," "chopsocky," "seen it all before," "colorful," and "semimusical."

These terms can be used to create comedic effects, such as satire, irony, or exaggeration.

Prompt #	Will	Jiheon	Jason	Ishan	Total	Avg
1	2	1	2	3	8	2
2	4	2	4	5	15	3.75
3	6	4	3	4	17	4.25
4	8	7	6	8	29	7.25
5-summ-before-ex1	9	8	8	8	33	8.25
5-summ-before-ex2	8	6	6	8	28	7
6	5	6	6	5	22	5.5
6b	7	4	6	7	24	6
6e	1	0	1	1	3	0.75
6f	6	5	6	6	23	5.75
6g	8	8	8	9	33	8.25
6h-summ-before	10	10	10	10	40	10

Figure 3: Group Rating Results.

4.4 ANALYSIS

This formulation of the most successful prompt puts conditionals and formatting requests first, and puts the more important task (select and return), later on in the prompt. This adheres to the recency challenge with LLMs, that they put more focus on later tokens in the prompt due to positional bias. Putting the question/task last helps the LLM stay on that given task. Our final conclusion for the

best ordering of the prompts, based on our experimental outputs (rated by humans 0-10) should be: *summarization*, then *examples*, then *question/task* at the end.

5 CONCLUSION

Our work seeks to discover how to formulate the prompt input that most effectively selects and summarizes ICL examples for a large language model. Using these formulated prompts, we predicted three main issues or challenges to overcome in our experiment:

1. The prompt input limit would make it difficult to include more in-context examples.
2. The LLM might observe incorrect connections/patterns across different examples.
3. Positional encoding bias may occur where the LLM observes and learns from more recent examples instead of all of them.

However, by the use of mini-batches, we were able to break down contexts and allow the LLM to process and learn from a large dataset, which provided us a wider range of the dataset. This did not only mitigate the limitation of prompt input, it also improved its ability to receive better examples that are relevant to the task. By applying variations to the structures of prompts and choosing the best prompt structure, we were able to mitigate the behavior of our LLM to learn incorrect connections/patterns from the examples. From the analyzation of different outputs with various prompt structures and mini-batch configurations, we chose the best prompt and set the most suitable mini-batch configuration for the LLM to produce the most relevant responses. In our case, the best prompt structure is 6h-summ-before. Determining the best way to prompt a LLM for effective in-context learning has significant value and potential. It is an easy and simple technique that can improve LLM's usability for their millions of daily users. It significantly improves their accuracy and quality and teaches users to utilize their unique in-context learning capabilities.

5.1 CONTRIBUTIONS

Each member was equally committed to the research project, and because of the clarity of communication between the group members, everybody was able to contribute in an equitable fashion. All in all, each of the four members contributed 25 percent.

REFERENCES

- Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Zhiyong Wu, Baobao Chang, Xu Sun, Jingjing Xu, Lei Li, and Zhifang Sui. A survey on in-context learning, 2023.
- Yizhe Zhang Bill Dolan Lawrence Carin Weizhu Chen Jiachang Liu, Dinghan Shen. What makes good in-context examples for gpt-3. 2021. URL <https://arxiv.org/pdf/2308.08780.pdf>.
- Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. Exploring demonstration ensembling for in-context learning, 2023.
- Lee Pang. ss2 dataset, 2005. URL <https://huggingface.co/datasets/ss2>.
- Ori Ram, Yoav Levine, Itay Dalmedigos, Dor Muhlgay, Amnon Shashua, Kevin Leyton-Brown, and Yoav Shoham. In-context retrieval-augmented language models, 2023.
- Sewon Min Sang Michael Xie. How does in-context learning work? a framework for understanding the differences from traditional supervised learning. *The Stanford AI Lab Blog*, 2022. URL <https://ai.stanford.edu/blog/understanding-incontext/>.
- 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 Dario Amodei Tom B. Brown, Benjamin Mann. Language models are few-shot learners. 2020. URL <https://arxiv.org/pdf/2005.14165.pdf>.

Jiachang Liu (2021) Ram et al. (2023) Dong et al. (2023) Sang Michael Xie (2022) Khalifa et al. (2023) Tom B. Brown (2020)