Research Study: CS 421

EchoPrompt: Instructing the Model to Rephrase Queries for Improved In-context Learning - Reproducibility Study

Davide Salonico[1]
[1]University of Illinois Chicago

Reproducibility Summary

Scope of Reproducibility – This study reproduces key claims from the EchoPrompt paper, which introduces a novel prompting strategy to enhance the performance of large language models (LLMs) in various reasoning tasks. The primary claims under investigation are:

- EchoPrompt improves accuracy in zero-shot settings for arithmetic reasoning tasks (e.g., MultiArith).
- EchoPrompt enhances performance in reading comprehension tasks (e.g., DROP subsets) when used with few-shot or chain-of-thought (CoT) prompting strategies.
- **Performance improvements result from the synergy between the original query and its rephrased version, which provides additional contextual clarity.**

Methodology – Experiments were conducted using GPT-3.5-turbo, accessed via OpenAI's API, with prompts and methodologies based on descriptions provided in the original paper. The study focused on two datasets: MultiArith for numerical reasoning and DROP_break for reading comprehension. All code and implementation details were adapted from the original paper's GitHub repository. Due to computational and budgetary constraints, systematic hyperparameter tuning and tests with deprecated models (e.g., code-davinci-002) were not performed. The total cost of API usage was kept minimal by limiting maximum tokens and experimenting with a streamlined setup.

Results – The results of this study closely match the original paper's findings:

- For MultiArith, EchoPrompt consistently improved accuracy in zero-shot, few-shot, and CoT settings, with results differing in a neglectabe way from the original paper's reported values.
- For DROP_break, EchoPrompt achieved similar accuracy gains, confirming its effectiveness in reading comprehension tasks involving discrete reasoning.
- Minor discrepancies in results are attributed to differences in model versions, as the original GPT-3.5-turbo weights are no longer available.

Overall, the findings support the original paper's claims about EchoPrompt's effectiveness.

What was easy – Implementing EchoPrompt was straightforward due to the availability of clear examples, prompt templates, and code in the original paper's GitHub repository.

What was difficult – Challenges included the inability to access deprecated models and budgetary constraints.

1 Introduction

The introduction of EchoPrompt[1] represents a significant step forward in enhancing in-context learning capabilities for large language models (LLMs). By prompting models to rephrase queries before solving them, the technique addresses common issues such as logical errors and omitted reasoning steps that frequently occur in standard and chain-of-thought (CoT) prompting methods. The authors demonstrated substantial performance improvements across diverse reasoning tasks, including but not limited to numerical reasoning (e.g., MultiArith[2], GSM8K), reading comprehension (e.g., DROP[3]), and logical reasoning.
In this reproducibility study, I focused on validating the performance improvements reported in the original paper by conducting experiments on two subsets: the MultiArith dataset (numerical reasoning) and the DROP_break dataset (reading comprehension). Given the constraints of computational budget and the deprecation of certain models (e.g., code-davinci-002), my study exclusively utilizes the GPT-3.5-turbo model and it focuses on two datasets in order to evaluate the model accuracy in different domains. The findings aim to establish whether EchoPrompt's effectiveness generalizes to these subsets under my experimental setup.

2 Scope of reproducibility

The original paper makes several key claims about the efficacy of EchoPrompt in improving model performance:

- **Claim 1:** EchoPrompt improves accuracy in zero-shot settings for arithmetic reasoning tasks, including datasets like MultiArith.
- **Claim 2:** EchoPrompt enhances performance in reading comprehension tasks, particularly on DROP subsets, when used with few-shot or chain-of-thought prompting strategies.
- **Claim 3: The performance improvements stem from the synergy between the original query and its rephrased version, which together provide additional contextual clarity for the model.**

The scope of this study is limited to testing these claims on the MultiArith and DROP_break subsets, using GPT-3.5-turbo, to assess the replicability of the reported improvements.

3 Methodology

3.1 Model descriptions

GPT-3.5-turbo, an advanced version of OpenAI's GPT-3 series, was chosen as the model for this study. With approximately 175 billion parameters, GPT-3.5-turbo is a powerful language model known for its strong in-context learning capabilities. The model is accessed via OpenAI's API[4], and all experiments adhere to the default configurations provided by the API, with minimal tuning.

3.2 Datasets

Two datasets from the original study were selected for reproducibility:

- **MultiArith:** A subset of arithmetic word problems requiring multi-step reasoning to solve. It evaluates the model's ability to handle numerical reasoning tasks effectively. The dataset contains problems with well-defined numerical operations. To help the reader having an idea of the content of this dataset, I show an example for each dataset:

question: ”The school cafeteria ordered 42 red apples and 7 green apples for students lunches. But, if only 9 students wanted fruit, how many extra did the cafeteria end up with?”,

answer: 40.0,

golden_answer: 40.0,

id : 3,

- **DROP_break:** A subset of the DROP (Discrete Reasoning Over Paragraphs) dataset, focusing on questions requiring discrete reasoning and often involving numerical calculations or logical deductions. Again, I report an example:

main_question: ”How many of the personnel were not officers?”,

question: ”The total number of active military personnel in the Croatian Armed Forces stands at 14,506 and 6,000 reserves working in various service branches of the armed forces. In May 2016, Armed Forces had 16,019 members, of which 14,506 were active military personnel and 1,513 civil servants. Of the 14,506 active military personnel, 3,183 were officers, 5,389 non-commissioned officers, 5,393 soldiers, 520 military specialists, 337 civil servants and 1,176 other employees. How many of the personnel were not officers?”,

passage: ”The total number of active military personnel in the Croatian Armed Forces stands at 14,506 and 6,000 reserves working in various service branches of the armed forces. In May 2016, Armed Forces had 16,019 members, of which 14,506 were active military personnel and 1,513 civil servants. Of the 14,506 active military personnel, 3,183 were officers, 5,389 non-commissioned officers, 5,393 soldiers, 520 military specialists, 337 civil servants and 1,176 other employees.”

answer: [[”11323”, 2], [”21932”, 1], [”1513”, 1]],

length: 44,

n_steps: 5

The datasets were processed as described in the original paper, ensuring consistency with their evaluation setup. Statistics such as the number of examples, label distributions, and train/test splits were directly obtained from publicly available resources.

3.3 Hyperparameters

No systematic hyperparameter tuning was performed, as the study aimed to replicate the results under standard conditions. Moreover, since the paper leverages API calls (which have an economic cost) it was not feasible to perform an exhaustive hyperparameter search. Additionally, the number of maximum tokens were limited in order to limit economic cost of the study. Despite this is not an hyperparameter per se I think it was worth mentioning.

3.4 Experimental setup and code

The experiments used Python scripts to interact with the OpenAI API. Prompts were implemented based on the examples and formats provided in the original paper. EchoPrompt experiments included:

- Generating rephrased queries using a task-agnostic prompt (e.g., “Let’s repeat the question and also think step by step.”).

- Combining the original and rephrased queries for final reasoning and answer generation.
- Rearranging the prompt structure (e.g. moving the question at the beginning of the sentence)
- Repeating the prompt more than once

All code is available for reproducibility on the original paper implementation at EchoPrompt. I'll provide an additional README to better explain how to run the code.

3.5 Computational requirements

Experiments were conducted on a cloud-based computational setup of which specification are not relevant since the most computationally complex operations are performed on the server. The average runtime per experiment was approximately 30 minutes per training and learning mode (a specific way of processing the queries), with a total of around 10 hours required for all experiments.

4 Results

It follows a table showing results of the reproducibilty study againt original paper scores.

Dataset	Data source	Zero shot Standard	Zero shot CoT	Few shots Standard	Few shots CoT
MultiArith	Original Paper	31.0/48.5(+17.5)	76.0/78.7(+2.7)	44.0/53.8(+9.8)	96.1 97.8(+1.7)
MultiArith	Reproducibility Study	31.2/47.5(+16.3)	75.9/78.8(+2.9)	44.0/53.5(+9.5)	96.1/97.8(+1.7)
DROP_break	Original Paper	43.7/55.8(+12.1)	38.2/51.2(+13.0)	55.5/63.1(+7.6)	65.3/69.6(+4.3)
DROP_break	Reproducibility Study	44.1/55.4(+11.3)	40.0/51.3(+11.3)	55.6/63.4(+7.8)	65.2/69.7(+4.5)

Table 1. Results comparison.
For every cell it's reported: result without EchoPrompt / result with EchoPrompt (difference)

It's possible to notice how the results[1] closely match the ones claimed in the paper. This is because the setting of the two experiments were very similar. A difference that probably impacted in a moderate way the results is that the model called from the API is slightly different (updated weights) and the legacy model is not accessible anymore. However, the results still confirm the reproducibility of the original paper.
It's important to remember that this is only a fraction of the many experiments reported in the paper and it involves only two dataset and one LLM (GPT-3.5-turbo).

4.1 Results reproducing original paper

My experiments on MultiArith and DROP_break confirmed the following:

- **MultiArith:** EchoPrompt improved over all the different settings compared to baseline performance. The improvement aligns with the original claim that EchoPrompt enhances numerical reasoning tasks.
- **DROP_break:** Accuracy always increased when using EchoPrompt, confirming that EchoPrompt provides significant benefits in reading comprehension tasks involving discrete reasoning.

4.2 Results beyond original paper

No additional datasets or experimental configurations were explored due to computational and time constraints. On the contrary, it would be extremely interesting to asses if the results are confirmed also for the other datasets mentioned in the paper.

5 Discussion

5.1 Do the results support the original claims?

The results observed in our experiments strongly support the original claims:

- The consistent improvements in accuracy validate the claim that EchoPrompt is effective for enhancing both numerical reasoning and reading comprehension tasks.
- The observed synergy between the original and rephrased queries corroborates the hypothesis that EchoPrompt's performance gains are rooted in this combination.

5.2 What was easy?

Implementing EchoPrompt was straightforward, thanks to the detailed examples and prompt templates provided in the original paper. The availability of pretrained models via the OpenAI API further simplified the process.

5.3 What was difficult?

Some challenges encountered include:

- Even though the original paper is relatively recent, some models (like code-davinci-002) are already deprecated, making the reproducibility way more complex. This makes the inability to test deprecated models such as code-davinci-002 or the original version of GPT-3.5-turbo used in the original work the main challenge of the study.
- Budgetary constraints limited the scope of experimentation and prevented an extensive hyperparameter search. I manteined total budget under 10$.
- Evaluating multi-step reasoning tasks occasionally required manual inspection to ensure prompt alignment with the original experimental setup.

5.4 Communication with original authors

No direct communication was initiated with the authors of the original paper. All reproducibility steps were based solely on the content of the paper and its supplementary materials. However, some issues (and their resepective answers) opened in the github repository base were extremely helpful.

6 Conclusion

My reproducibility study confirms that EchoPrompt substantially enhances the performance of GPT-3.5-turbo on the tested subsets of MultiArith and DROP_break. The improvements align closely with the original claims, demonstrating EchoPrompt's robustness and utility as a prompting strategy.
Future work could involve expanding the study to include additional datasets and models, as well as investigating the impact of systematic hyperparameter tuning on EchoPrompt's performance.

References

1. R. R. Mekala, Y. Razeghi, and S. Singh. **EchoPrompt: Instructing the Model to Rephrase Queries for Improved In-context Learning**. 2024. arXiv: 2309.10687 [cs.CL].
2. S. Roy and D. Roth. "Solving General Arithmetic Word Problems." In: Jan. 2015, pp. 1743–1752.
3. D. Dua, Y. Wang, P. Dasigi, G. Stanovsky, S. Singh, and M. Gardner. **DROP: A Reading Comprehension Benchmark Requiring Discrete Reasoning Over Paragraphs**. 2019. arXiv: 1903.00161 [cs.CL].
4. OpenAI. **OpenAI API Documentation**. Accessed: 2024-12-04. 2024.