SWER – Project Progress Report Studiability

Bowen Yang UNI: by2365

April 2024

1 Overview

To develop a system capable of using GPT-3.5 to address math word problems effectively, I embarked on a thoughtful journey, detailed as follows:

1. Initially, I explored a Retrieval-Augmented Generation (RAG)[2]-based concept. Considering that math questions at the primary school level are generally straightforward and exhibit limited patterns, I speculated that it might be feasible to classify all problems and leverage similar questions as few-shot prompts to enhance the system's efficiency. However, I quickly realized that achieving a detailed categorization is as challenging as solving the problems directly, rendering this approach impractical.

2. Subsequently, I considered an alternative where the model could learn from its errors without the need for reinforcement learning. By compiling a repository of instances where the model faltered and using these as cues in solving new problems, I hoped to improve accuracy. However, this strategy still demands precise categorization to match relevant tips with current questions, thus proving unfeasible. Nonetheless, employing few-shot examples that include both incorrect and correct solutions might enhance performance. I plan to investigate further if I have enough time.

3. After trying other methods, I started focusing on directing the thought process of language models by using specific prompts. But, from initial tests, I found two big issues. First, it's challenging to create a logical structure that works universally for all types of questions. Second, machines process information differently than humans. Trying to make them mimic human thinking often doesn't enhance their performance and can sometimes hinder it.

Even though many of my ideas didn't work out, I discovered some strategies that could help models do better on math word problems. So, I suggested a new system that could enhance how GPT-3.5 tackles these problems. This system would use several agents, each taking different few-shot examples as input, and then produce a solution that goes through a refining process. This process further polishes the solution before it's sent to a comparison pool. There, all the solutions are evaluated against each other, and the best one is chosen as the final answer.

2 Research Questions

1. What abilities does GPT-3.5 lack that hinder its efficiency in solving math word problems?
2. What factors affect the reasoning abilities of LLMs in general?
3. How can we enhance the proficiency of LLMs in solving math word problems?

3 Value to User Community

Training powerful large language models (LLMs) like GPT-4.0 needs a lot of resources, making them quite costly. For instance, using GPT-4.0 for API calls is about 45 times more expensive than using GPT-3.5-turbo. You can also use GPT-3.5-turbo for free in web applications while need to pay for subscription for GPT-4.0. It's expected that this cost difference will continue with future versions, such as GPT-5.0, which will be significantly larger and likely more expensive than GPT-4.0. Boosting GPT-3.5's ability to solve math problems can significantly benefit society, especially since it's much more cost-effective compared to GPT-4. This affordability makes it an accessible tool for a wider audience, including researchers and students who often operate under budget constraints. By enhancing its mathematical capabilities, GPT-3.5 can serve as a valuable resource in educational settings, offering personalized tutoring and assistance with complex math problems, thereby democratizing access to quality education. In research, it can aid in tasks like data analysis, speeding up scientific discoveries and innovation.

4 Demo

I'll demonstrate a Python application that I can interact with, which will answer the questions I've set. This application will display its thought process in the terminal.

I'd like to develop an Android app if I have enough time.

5 Delivery

The project will be made available through a public repository. However, a portion of the GSM8K data will need to be included in the repo since it's necessary for few-shot prompting. I'll make sure to cite these in the references section.

6 Extra

I will use GSM8k[1] as the major dataset. Popular frameworks such as RAG[2] and ReAct[3] are not that useful in this scenario.

7 Reference

1. Cobbe, K., Kosaraju, V., Bavarian, M., Hilton, J., Nakano, R., Hesse, C., & Schulman, J. (2021). Training Verifiers to Solve Math Word Problems. arXiv. https://arxiv.org/abs/2110.14168

2. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.-t., Rocktäschel, T., Riedel, S., & Kiela, D. (2020). Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks. arXiv. https://ar5iv.org/abs/2005.11401

3. Liu, Z., Shen, Z., Savvides, M., & Cheng, K.-T. (2020, March 7). ReActNet: Towards Precise Binary Neural Network with Generalized Activation Functions. DeepAI. https://deepai.org/publication/reactnet-towards-precise-binary-neural-network-with-generalized-activation-functions