Enhancing Large Language Model Teaching Ability Via Directed Acylic Graphs and Insights

Michael Li
Troy High School
Troy, MI
mli4648@gmail.com

Matthew Li
Troy High School
Troy, MI
mattqlf25@gmail.com

Sanchi Deshpande
Adlai E. Stevenson High School
Lincolnshire, IL
sanchidesh5@gmail.com

Eesh Pant
Monte Vista High School
Danville, CA
eesh.pant@gmail.com

Abstract

Large language models are widely explored in their capabilities in education. However, they experience difficulty in engaging in dialogue that is truly helpful and relevant because the user may not have the necessary prerequisite knowledge and may not be able to apply what he/she has learned to more complex and difficult but similar situations. We propose two new techniques to ameliorate the quality of dialogue between two models. 1.) During a knowledge distillation process, the "teacher" model generates a Directed Acyclic Graph to record the student's knowledge 2.) An insight generator model provides the "student" model with important hints to apply to solving a problem. Surprisingly, adding more models and features to the problem-solving process results to worse performance than the baseline because the models were unable to strictly follow the prompts and instructions.

1 Introduction

As Large Language Models continue to be explored for education as there has been a growing interest in using models for additional assistance with math, physics, and many other subjects. Models have access to a more diverse and deep swath of information which can enable them to instruct students more effectively given the right training.

We consider educational conversations in the context of mathematical problems. Specifically, we use entries of the MATH dataset that a teacher model can solve but the student model cannot. Math problems, especially those in competitions, are complex for two reasons: 1) a student may not have the necessary background knowledge to solve them. 2) even if the student has the necessary background, there may non be intuitive steps or complex reasoning that the student may struggle with.

In this paper, we exhibit a series of advancements for knowledge distillation with dialogue in an educational context. Our contributions are as follows:

1.) We introduce the application of Directed Acyclic Graphs (DAGs) in constructing a graph of the “student” model’s prerequisite knowledge. Each node in the DAG represents a concept, while edges represent dependencies. This knowledge conversation is used to address the first source of difficulty by clearing up any prior misconceptions that the student may have. To generate these DAGs, we use a DAG LLM (GPT-4o) along with a carefully crafted prompt. The responsibility of the DAG model

Preprint. Under review.

is to construct a fixed graph which is passed into the teacher model, which is then prompted to teach the topics in the DAG, respecting its structure. A 3-turn conversation is initiated between the teacher and student model. The DAG is very intuitive and reminiscent of how humans learn , which is why we chose this idea.

2.) We also generate a list of crucial "insights", which are key connections that a student must draw in order to solve the problem. These closely resemble hints that a real teacher would give to students to assist without providing direct answers. These insights are generated by an insight model (GPT-4o), again with a specific prompt to do so.

2 Related Works

General bot conversations have been done before in ways that not even artificial intelligence could detect. GPT-4 and GPT-3.5 have been shown to be exceptional at generating conversation (Duan et al. 23), which we took into account as we were deciding our teacher and student models.

The use of pedagogical domain knowledge in dialogue based teaching has been implemented before as well (Nye et al. 23). The best teachers are the ones that can understand not only the subject matter but also the misconceptions that are present (Nye et al. 23). This means that the teacher's questions to the student help the student think critically about the problem rather than revealing the answer. The paper - Knowledge Graphs(KGs) as Context Sources for LLM-Based Explanations of Learning Recommendations contributed to the development of idea on DAGs. The authors extract relevant context from KGs to supplement the recommendations made by LLMs and leverage the contextual information from KGs to generate understandable explanations for the recommendations provided by LLMs. (Rasheed et al. 24) DAGs are similar as they understand a student's background's knowledge, graph it, and extract from it. Their results were exceptional as they reduced risk of imprecise information compared to a mere GPT model.

We borrowed from the MINT paper (Xingyao et al. 23) on the the evaluation of LLMs in multi-turn interactions. The authors used an LLM to simulate providing feedback to the LLM solving the problem. We branched off this idea to use for our teacher model in order to provide the student with specific feedback.

3 Methods

3.1 Obtaining Data

We selected GPT-4o as the teacher model as it was most accurate on problems from the MATH dataset. For the student, we wanted to pick a model with the least accuracy and attempted models like Phi-1.5b. Unfortunately, they were unable to provide a proper response and produced irrelevant dialogue. Instead, we settled on GPT-3.5-Turbo, a model with decent instruction-following capability, but also left room for possible improvements in MATH accuracy from our experiments.

To obtain problems that would be the most effective when combined with our pedagogy methods, we filtered the MATH dataset according to the criteria that each problem had to be solved by our teacher model while also being solved incorrectly by the student model. This was implemented because the teacher model needed to have sufficient knowledge of the ground truth to properly guide the student from an incorrect to a correct answer. Furthermore, to increase the discrepancy between the teacher and student, we used a 3-shot prompt, along with chain-of-thought (CoT) reasoning before giving the teacher the actual problem, maximizing the teacher's accuracy. However, the student received CoT prompting but it did not receive a 3-shot prompt, as this method is more consistent to the environment that the student will be learning in during the actual dialogue (where only CoT is present). Of the 12500 problems in the MATH dataset, we condensed down to 6600 usable problems to experiment on.

3.2 Training Pipeline

There will be two models interacting through dialogue. The teacher model (GPT-4o) will know the ground truth with the problem's answer and solution, as well as the train of thought to inform the student of correctness. The student model (GPT-3.5-Turbo), will be trained on sets of problems from

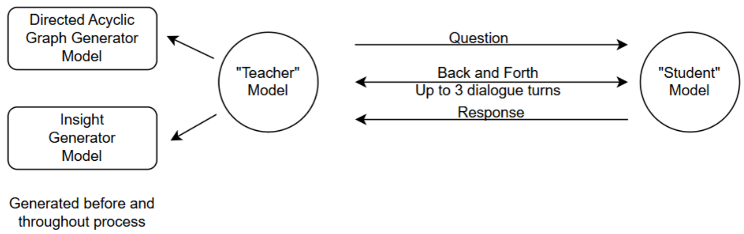

Figure 1: Models Architecture

the MATH dataset (Hendrycks et al. 21) which it cannot solve on its own. Through our pipeline, we hoped that the student can learn the prerequisite steps to solve a problem.

We employ a two-pronged approach to the problem, consisting of insights and a directed acylic graph (DAG) to direct the teacher. We first generate a DAG (directed acyclic graph) of prerequisites to the problem which will direct the teacher model (GPT-4o) to ask appropriate questions to the student model to most effectively guide it to producing the correct answer. Next, we update the DAG as needed. If a student has sufficient understanding of the current layer of prerequisites, the teacher model will move on to the next higher layer of prerequisites if it exists. Otherwise, the teacher will repeat through the previous steps by generating new, layers to the prerequisite DAG, at a lower level. Once the student has mastered all the prerequisite topics, the teacher will ask the original question, to which the student will attempt to provide the correct answer.

3.3 Insights

We use GPT-4o as the insight model which will generate insights for each problem given the answer and solution (ground truth). Then, to increase the odds of getting an effective list of insights, we apply k-shot prompting with manually crafted positive examples. We take this insight list and feed it into to the teacher, along with a specialized prompt which will drop insights one at a time as needed by the student, checking to see if the student can finish the problem at that point or dropping new insights if needed. We attempt to prevent "leakage" of the answer (where the teacher reveals the answer to the student without the student arriving there on its own), by explicitly stating this in the prompt. We also filtered out conversations in which the teacher still revealed the answer prematurely despite telling it not to , resulting the student automatically getting the correct answer without proper reasoning.

4 Results

After running our experiment, our method, which includes the combination of a Directed Acyclic Graph and insights from the teacher model, performed worse than the baseline model. The baseline model achieved an accuracy of about 79.11, compared to our method which achieved an accuracy of 47.57%. The above results are because the prompts were too difficult for the models to follow. GPT was unable to handle the prompting due to the strictness of the extensive prompts.
Here's an example experiment conversation:
Problem: Find the distance between the points $(1, 1)$ and $(4, 7)$. Express your answer in simplest radical form.
Solution (ground truth): The horizontal separation between $(1, 1)$ and $(4, 7)$ is $4 - 1 = 3$ units. The vertical separation between the points is $7 - 1 = 6$ units. Therefore, the segment whose endpoints are (1,1) and (4,7) is the hypotenuse of a right triangle whose legs measure 3 units and 6 units. By the Pythagorean, the length of this segment is $\sqrt{3^2 + 6^2} = 3\sqrt{1^2 + 2^2} = 3\sqrt{5}$.
The generated DAG : add_node("Distance Formula") | add_node("Pythagorean Theorem")| add_node("Concept of Coordinates") | add_node("Simplifying Radicals") | add_edge("Pythagorean Theorem", "Distance Formula") | add_edge("Concept of Coordinates", "Distance Formula")| add_edge("Simplifying Radicals", "Distance Formula") |
Insight List:
1. Recognize that this is a distance problem in the plane, which suggests using the distance formula or Pythagorean theorem.

2. Identify the horizontal separation as $4 - 1 = 3$ units.
3. Identify the vertical separation as $7 - 1 = 6$ units.
4. Realize that these separations form the legs of a right triangle.
5. Apply the Pythagorean theorem to determine the length of the hypotenuse.
6. Simplify the resulting expression to obtain the final answer in simplest radical form.

Conversation:

Teacher: Let's work through this problem step by step to determine the distance between the points $(1, 1)$ and $(4, 7)$. Firstly, we need to recognize that this is a distance problem in the plane. We can solve it using the distance formula or by interpreting it as finding the hypotenuse of a right triangle using the Pythagorean theorem. Let's start by identifying the horizontal separation. Can you calculate the horizontal distance between the points $(1, 1)$ and $(4, 7)$? **Student:** Yes, let's calculate the horizontal distance between the points $(1, 1)$ and $(4, 7)$. The horizontal distance is the difference between the x-coordinates of the two points. So, we have: Horizontal distance = $4 - 1 = 3$ Now, let's move on to finding the vertical separation between the points. /newline **Teacher:** That's correct! The horizontal distance is 3 units. Next, let's identify the vertical separation between the points $(1, 1)$ and $(4, 7)$. Can you calculate the vertical distance by finding the difference between the y-coordinates of the two points?/newline **Student:** Yes, to find the vertical distance between the points $(1, 1)$ and $(4, 7)$, we calculate the difference between their y-coordinates. Vertical distance = $7 - 1 = 6$ units Now that we have the horizontal and vertical distances, we can apply the Pythagorean theorem to find the total distance between the two points. **Teacher:** Excellent! You've correctly identified the vertical distance as 6 units. Now, we understand that the points $(1, 1)$ We attempt to prevent "leakage" of the answer (where the teacher reveals the answer to the student without the student arriving there on its own), by explicitly stating this in the prompt. and $(4, 7)$ form a right triangle with legs measuring 3 units and 6 units. Let's use the Pythagorean theorem to determine the length of the hypotenuse, which represents the distance between the points. The Pythagorean theorem states that $a^2 + b^2 = c^2$, where a and b are the legs of the triangle, and c is the hypotenuse. Can you substitute the values we found into this equation and solve for c? **Student:** Certainly! We can substitute the values of the legs into the Pythagorean theorem to find the hypotenuse (c).
Using $a = 3$ and $b = 6$, we have:

$$3^2 + 6^2 = c^2$$
$$9 + 36 = c^2$$
$$45 = c^2$$

To simplify, we take the square root of both sides to solve for c:

$$c = \sqrt{45}$$
$$c = \sqrt{9 \times 5}$$
$$c = 3\sqrt{5}$$

Therefore, the distance between the points $(1, 1)$ and $(4, 7)$ is $3\sqrt{5}$.

5 Future Experimentation

Conducting ablations to test out how the two models – DAG and Insights – impact dialogue separately may improve performance. Since the teacher and student model got overwhelmed, testing the models with only the DAG and only the insights each could help the teacher and student understand their tasks better. More prompt testing should be done to streamline prompts without confusing the models. Reducing student model degeneration is another key aspect of this method and better prompting and fine-tuning could help the student stay in character as a student without attempting to become the teacher. Although not actually implemented due lack of student compliance, we originally planned to implement a RAG based "memory" system, where previous responses from the student in various problems would be stored in a vector dictionary. Then, we planned to use FAISS cosine similarity search to look for similar responses to a current problem, and provide the student with more information that it had previously learned. The student model could access these "memories" of prior conversations to aid them in solving future problems. If properly implemented, this idea could have significant improvements on effeciency of output.

References

Nye, Benjamin, et al. Generative Large Language Models for Dialog-Based Tutoring: An Early Consideration of Opportunities and Concerns, University of Southern California Institute for Creative Technologies, 20 Sept. 2023, ceur-ws.org/Vol-3487/paper4.pdf. Accessed 20 June 2024.

Duan, Haodong, et al. "BotChat: Evaluating LLMs' Capabilities of Having Multi-Turn Dialogues." *arXiv preprint arXiv:2310.13650* (2023). Accessed 20 June 2024.

Wang, Xingyao, et al. "Mint: Evaluating llms in multi-turn interaction with tools and language feedback." *arXiv preprint arXiv:2309.10691* (2023). Accessed 20 June 2024.

Hendrycks, Dan. "Hendrycks/Competition Math · Datasets at Hugging Face." Hendrycks - Competition Math, huggingface.co/datasets/hendrycks/competition$_Math.Accessed20June2024$.

Rasheed, Hasan, et al. *Knowledge Graphs as Context Sources for LLM-Based Explanations of Learning Recommendations* . arxiv.org/pdf/2403.03008. Accessed 28 June 2024.