

Assignment 2: Prompting LLMs to answer maths questions

Leonardo Ranaldi, Joshua Ong and Shay Cohen

Overview

While large language models (LLMs) are groundbreaking technology, they often struggle with reasoning tasks, particularly mathematical reasoning. In this assignment, you will explore prompting techniques for mathematical reasoning using simple maths word problems and analyse LLMs in this context. The work is based on the paper Chain of Mathematically Annotated Thought ([link](#)), referred to as CoMAT. Before you begin, please skim the paper to familiarise yourself with its key concepts and details. To make the assignment self-contained, we provide the important details below.

How CoMAT Works: CoMAT relies only on the predictions of the large language model to solve the question. It is a relatively straight-forward *prompt engineering technique*. Its steps are detailed below, with each step corresponding to an instruction in the prompt used to solve the question.

Symbolic conversion: The LLM is asked to formalise the natural language question Q in four intuitive steps that you might have internalised in some form in a maths class. The four steps include *identification and definition*, *structural logic translation*, *explicit factual representation*, and *question formalisation*.

Hence, given the following question MQ0:

MQ0

Taylor Swift is planning a concert tour. The venue can hold 50,000 fans. VIP tickets cost \$250 each, and regular tickets cost \$100 each. If the total revenue from ticket sales is \$6,500,000 and all tickets are sold, how many VIP tickets were sold?

s_1 . **Identification and Definition.** CoMAT identifies and defines the relevant variables and constants in MQ0.

- **Variables:** v (number of VIP tickets), r (number of regular tickets)
- **Constants:** T (total capacity of 50,000), P_v (price of VIP tickets, 250), P_r (price of regular tickets, 100), R (total revenue, 6,500,000)

s_2 . **Structural Logic Translation:** Then, CoMAT extracts the key variables and translates the problem into formal rules that define their relationships:

- $v + r = T$
- $P_v \cdot v + P_r \cdot r = R$
- $v \geq 0, r \geq 0$ (non-negative constraints)

s_3 **Explicit Factual Representation:** CoMAT then integrates all relevant facts into the logical structure:

- $T = 50,000$
- $P_v = 250$
- $P_r = 100$
- $R = 6,500,000$

s_4 **Question Formalisation:** CoMAT formalises the question based on the previous steps:

In our example, we are tasked with finding v , the number of VIP tickets:
Find $v : (v + r = T) \wedge (P_v \cdot v + P_r \cdot r = R)$

Reasoning Execution: During the second part, the problem is solved step-by-step using previous passages, as demonstrated:

Step 1: Express r in terms of v using $v + r = T$:

$$r = T - v = 50,000 - v$$

Step 2: Substitute into the revenue equation $P_v \cdot v + P_r \cdot r = R$:

$$250v + 100(50,000 - v) = 6,500,000$$

Step 3: Simplify:

$$250v + 5,000,000 - 100v = 6,500,000$$

Step 4: Solve for v :

$$v = \frac{1,500,000}{150} = 10,000$$

6. **Derivation of Final Answer:** The final answer is then derived. In this case:

The number of VIP tickets sold is 10,000.

In the next set of questions, we will explore CoMAT both programmatically and conceptually. You will learn what makes the case for such step-by-step chain-of-thought style prompting and whether it can be improved.

Assignment Questions

The questions below are designed to help you become familiar with solving math problems methodically, following an analysis of solving such problems with an LLM. You will need to input your answers on a Gradescope form; refer to the *Submission Procedure* for more details.

Q1 - Formalising questions

In the first part of the assignment, you will be required to formalise two questions yourself, as if you were the LLM. This should provide you with an idea of the kind of skill a language model needs to possess in order to derive the solution.

The two questions you will work on formalising are:

MQ1

In a neighbourhood, the number of rabbits pets is twelve less than the combined number of pet dogs and cats. If there are two cats for every dog, and the number of dogs is 60, how many pets in total are in the neighbourhood?

MQ2

Company X shipped 5 computer chips, 1 of which was defective, and Company Y shipped 4 computer chips, 2 of which were defective. One computer chip is to be chosen uniformly at random from the 9 chips shipped by the companies. If the chosen chip is found to be defective, what is the probability that the chip came from Company Y?

Choices:

- a. $2/9$*
- b. $4/9$*
- c. $1/2$*
- d. $2/3$*

1. Solve MQ1 and MQ2 using any method or approach of your choice. Show your calculations clearly. *Do not use an LLM to solve MQ1 and MQ2.*

2. Now, follow the steps of Identification and Definition, Structural Logic Translation, Explicit Factual Representation and Question Formalisation for MQ1 and MQ2. Use the example at the beginning of the assignment as a basis for this formalisation. *These steps should be followed manually, rather than with an LLM. Do not use an LLM to solve this question.*
3. Now, based on the formalisation, solve the questions and get an answer (“Derivation of Final Answer”). You do not have to follow the reasoning execution steps.
4. Explain how the structured approach in 2-3 helped you arrive at an answer more mechanically, and critically assess whether it made reaching a solution easier. Discuss both the advantages and limitations of this approach, reflecting on its effectiveness in guiding your reasoning process. Note that there is no right or wrong answer; the goal is to thoughtfully evaluate the process.
5. Following subquestion 2, do you think additional formalisation steps could improve the process? Provide an example of such a step to support your argument. Alternatively, you may argue that no further formalisation steps are necessary and that the current sequence is complete. A strong answer should include specific examples and justification to support your answer.

Q2 - Analysis of steps through Shapley values

In this question, we will study the contribution of each step by analysing the different components in the step-wise procedure of formalisation. The analysis is based on the *Shapley value* of the different steps. The Shapley value defines a way to measure the contribution of a “player” to a payoff in a “game.” Formally, assume there are n players (in our case, each player corresponds to a step in CoMAT) and that there is a way to measure the level of success of a subset of these n players, $S \subseteq \{1, \dots, n\}$, in the game, denoted by $v(S)$. Let π be a permutation over the set of n players (meaning, an ordering of the n players). We define the payoff difference for player i as:

$$\Delta_i(\pi) = v(S_i \cup \{i\}) - v(S_i), \quad (1)$$

where S_i is the set of players that precede i in the ordering π . The Shapley value is then defined for each player $i \in \{1, \dots, n\}$ as:

$$\phi_i = \frac{1}{|\Pi|} \sum_{\pi \in \Pi} \Delta_i(\pi), \quad (2)$$

where Π is the set of permutations over the n players (all orderings).

In this question, we let $v(S)$ be the fraction of questions (over the full evaluation dataset) that are answered correctly when using only the steps in S (the others are omitted). This fraction is calculated on some held-out validation set of questions and answers (number of

correctly answered questions from the held-out set divided by the total number of questions in the held-out set).

To make it more concrete, if $\pi = [1, 4, 2, 3]$, and we are focusing on player 2, then $S_2 = 1, 4$. If the question at hand is answered correctly using only steps 1 and 4, then for that question, we count a correctness of 1; otherwise, 0. Similarly, if the same question is answered correctly using only steps 1, 4 and 2, then we again count it as 1 for that question, otherwise 0. We can compute $\Delta_2([1, 4, 2, 3])$ based on these values. To compute ϕ_2 , the Shapley value for step 2, we will range π over all possible permutations, each time creating S_i , which are the steps that precede step 2.

1. Fill in the template provided with this assignment, which consists of two major sections: evaluating the *mmlu-redux* dataset using GPT and calculating the Shapley value. Please follow the instructions carefully.

Code Instructions

You will complete 7 questions within the provided code files (excluding subquestions). Ensure that the completed code is included in the final document. You can grep for the questions by typing `grep Question *.py` on a Terminal. To download the required dependencies, execute the following command:

```
pip install -r requirements.txt
```

Note: Please avoid downgrading or altering the dependency versions.

Now, answer the questions based on the instructions provided below:

- (a) *Questions 1 and 2:* These are in *utils.py*, where you will create a function for GPT-based predictions. Follow the provided configuration carefully and ensure it is explicitly passed when calling the OpenAI function.
- (b) *Questions 3:* This question is located in *CoMAT_Instruction.py*, where you are instructed to fill in the prompt instructions.
- (c) *Questions 4 to 7:* These are in *shapley_value_evaluation.py*, where you will implement functions to evaluate the Shapley analysis.

Note: Helpful tips and explanations have been included in the code. Please retain them for clarity and guidance.

Please carefully follow the instructions in the code and the configurations below to answer the questions.

Configuration:

- `temperature = 0.0`
- `model = gpt-4o-mini`
- `max_tokens = 2000`

To evaluate *main.py* , execute the following command:

```
python main.py --dataset mmlu-redux-college_mathematics
               --method comat --model gpt
```

Take Note:

Note: A higher accuracy does not lead to higher marks, so please *do not* use a more advanced model than gpt-4o-mini. We ask you to use gpt-4o-mini to balance the costs, so you have the ability to evaluate the full dataset multiple times. Each student is allocated approximately \$8 in total for this assignment, and a full evaluation of the dataset costs about \$0.05. We estimate you could run the evaluation at most 100-150 times from beginning to end. Far fewer runs are actually needed. Budget your tokens!

2. Answer the following questions based on the code for prompting the LLM:
 - (a) Evaluate the model using the given configuration with temperature 0.0. What is the accuracy of the model under this setting?
 - (b) Adjust the temperature setting to 0.7 and evaluate the model again. What is the accuracy of the model this time?
 - (c) Compare the accuracy under this configuration to the original evaluation. Explain why the accuracy is higher or lower when the temperature is set to 0.7 compared to 0.0.
 - (d) What alternative configurations other than adjusting temperature could affect performance, and why? Note: Changing the model itself is not a valid answer; focus only on configuration adjustments within the same model. (Consider what other hyperparameters control the inference of the model, for example, the top-p parameter.)
3. You might ask yourself: How much does each step contribute to the success of solving a given question?

To do this, calculate the Shapley value of each step, with total of 4 steps based on the code you have written. Base your calculations on the file *evaluation_with_steps.csv*, which includes 0/1 values indicating correctness for different configurations of the steps. Do not answer the question based the execution of the different steps yourself. Your calculations will be done by completing the code in *shapley_value_evaluation.py*, which you are required to submit.

- (a) What are the Shapley Values for each step s_1, s_2, s_3, s_4 ? Include the contribution of each step.
(**Note:** Negative Δ values do exist, so please do not assume that the result cannot contain negative Δ values when evaluating the Shapley Value.)
- (b) Which step receives the highest Shapley value? Argue why this might be the case.
- (c) Order the steps by their Shapley value. What do you notice between the relationship of the Shapley values to this order? Argue why this is the case.
- (d) Based on Appendix D in the CoMAT paper, removing both Steps 1 and 2 results in a smaller accuracy drop than removing Step 1 alone (4.48% higher accuracy). Argue what this tells about the dependence between the steps.

Q3 - Analysing a case in which CoMAT does not work

In this question, you will identify a question for which the steps in CoMAT fail to provide the correct answer. You will have to investigate the dataset provided and analyse a few questions, possibly manually. Since CoMAT is far from being perfect, it should not take you too long to find a case in which it fails. Hence, after finding the question Q :

1. Write down the full set of steps CoMAT provides through the large language model for a question in which the final answer is incorrect.
2. Analyse the steps CoMAT follows and identify where it fails.
3. Can you now suggest a step that would help fix this failure? Try it out with an LLM. Did it indeed fix the wrong answer?

Q4 (extra credit) - Using LLMs to prove an identity about the Shapley value

Prove that Shapley value with permutations (as in Equation 2) will give the same answer if you use the formula below with sets rather than permutations:

$$\phi_i = \sum_{S \subseteq [n] \setminus \{i\}} [v(S \cup \{i\}) - v(S)] \cdot \frac{1}{n \cdot \binom{n-1}{|S|}}. \quad (3)$$

You may use a language model to help you solve this problem. Briefly report about your experience using the LLM to solve the question. Where do you think LLMs can improve? How did the LLM help you solve the above?

Hint: How many permutations become equivalent given a specific i when they are reduced to a set?

Submission Procedure

Timeline We recommend starting early to work on the assignment. The deadline is March 21, 2025 (Friday) at 12:00.

Usage of Edinburgh’s Access to Language Models As part of this assignment, you were provided an OpenAI key to use with ELM. You will need to use this key as part of your code. Note that you should NOT use this key for any other reasons. In addition, note that you should not run your code through the whole dataset more than 100 times or so (which should be plenty to complete the code). Your budget is \$8. If in doubt, run the code on a small part of the dataset until you have a stable version of the code. You should ONLY use the gpt-4o-mini model. If you have not yet received your key (search for an email with the string “Coursework 2” in the subject from scohen@inf.ed.ac.uk), please contact us as soon as possible.

Submission entry point Your group’s solution to the assignment should be submitted on Gradescope. The Gradescope submission entry is a *form* where you will need to put your answers in the specific rubric as detailed in the above script. We highly recommend to:

- Write down your answers on a scratch pad or in Overleaf, and then copy paste your answers into Gradescope (possibly formatting them if needed) when you feel ready. Gradescope does have a “Save answer” button, where you can enter your answers gradually, but keeping a separate document could save your group a lot of trouble.
- Not leave the submission of the assignment to the last minute! Make sure you are familiar with the form beforehand, and that you have enough time to enter your solution properly.

If you used maths equations in your answer, and used Overleaf or L^AT_EX for that, you may need to convert them into Gradescope. The most likely thing is that you would need to change \$ to \$\$. See more information here: <https://guides.gradescope.com/hc/en-us/articles/21591530268429-Can-I-use-LaTeX-on-Gradescope>.

Good scholarly practice As with other assignments in this course, this assignment is intended to be done in pairs. You may freely discuss and share work within your pair, but may not share or make available solutions or code outside your group. As usual, please use care when posting questions to ensure you don’t give away parts of the solution. In

addition, you are responsible for following the University academic misconduct policies regarding all assessed work for credit. Details and advice about this can be found at <http://web.inf.ed.ac.uk/infweb/admin/policies/academic-misconduct>.

We are aware that some of the code required for this assignment may be available online. You are expected to write the code on your own – and there are tools to identify code similarity that indicate a suspect copying of a code compared to a baseline code.