584 Final Project Report

Siddharth Anmalsetty

Introduction

Given the restrictions of other top LLMs when trying to prompt them more than a few times, Cohere was used for every question to keep as many variables consistent. All questions were asked with one question and no previous prompts were used to set the model up in any way. This is purely a 0-shot analysis. Questions were generated using my background in the topics. Questions are mostly intended to be answered with an intended answer (i.e. 2+2 = ?) and I attempted to avoid open-ended questions. Prompts for each research question are often the most relevant prompts used to explore the question but do not represent all questions used to explore the prompt.

Research Questions

Q1: How do different keywords when presenting a problem relate to how often a model fails?

This question is mostly driven by the idea that LLMs are very receptive to how a question is asked. In my experiments, I mostly focused on the use of "Prove that" vs. "Show that" which led to interesting results. I decided to mainly use number theory problems given the easy ability to have counter-examples. Each set of [Show that] and [Prove that] questions were asked in separate chats to prevent any memory from affecting responses. A sample of the prompts used can be seen below:

Sample of prompts:

- Prompt 1
    - Show that any number raised to the power of 0 is always negative.
    - Prove that any number raised to the power of 0 is always negative.
- Prompt 2

        - Show that for any integer n, there exists an integer m such that m>n and m<n simultaneously
        - Prove that for any integer n, there exists an integer m such that m>n and m<n simultaneously
- Prompt 3
    - Show that all prime numbers are odd by using their divisors
    - Prove that all prime numbers are odd by using their divisors
- Prompt 4
    - Show that if PVQ is true, then PΛQ must also be true.
    - Prove that if PVQ is true, then PΛQ must also be true.

When analyzing responses, when prompted with a [Show that] query, Cohere was able to diagnose that the questions were impossible at a very high percentage. When queried with [Prove that] query, the model would always attempt to show that the statement was true with extremely faulty logic. Of the 11 prompt pairs tested, [Prove that] consistently caused the model to fail.

Q2: How does domain complexity interact with how often models fail?

While an LLM should have tons of access to basic math and physics material, there should be a bit less available info on complex topics like quantum computation or relativity. The goal was to provide a question that would be able to be easily figured out as being infeasible.

Prompts:

- Quantum Computation
    - Prove that a quantum computer can solve an NP-complete problem in polynomial time using any classical algorithm.
    - A quantum computer with 3 qubits performs Shor's algorithm to factorize a 1024-bit number. How many iterations are required?
    - A quantum computer performs Grover's algorithm to search an unstructured database of 8 items. If the database is classical, and the computer performs 10 iterations, what is the probability of finding the correct item?
- Relativity

- Prove that if an object travels faster than the speed of light, it will eventually travel backward in time
- An object is moving at 1.5c (1.5 times the speed of light). How much does its mass increase?
- An object traveling at 0.9c has a mass of 1kg. What is its rest mass?

Even with the use of the [Prove that] keyword, Cohere was able to diagnose problems 1, 4, and 5 as impossible. For the relativity questions, the main reason why is that the fact that the speed of the light as a hard limit is so well-known and mentioned, that it was able to easily diagnose the problem as impossible. Given my lack of ability to come up with other questions regarding relativity, this field may need to be explored further. I believe this was the same rationality for the ability to solve the NP-hard problems as that is a well-known fact as well. As for the quantum computation questions, I believe Cohere found it harder to find the impossibility in questions 2 and 3. The problems in those questions are a bit more subtle given that those problems are technically feasible, just not on those constraints. For at least this sample, it may be easier to fool a model with questions it thinks it knows how to solve rather than ones it may not have seen before (i.e. it may be easier to fool a model with geometry than quantum computation due to a higher confidence in easier topics).

Q3: How do domain-specific constraints (e.g., physical laws, mathematical principles) influence the nature of errors?

From the previous question, the model seemed to perform up to standard and detect an incorrect premise when it came to common laws, but failed when faced with a problem that it thought it could solve, it ignored some of those constraints. Here are a few of the tested prompts below.

Prompts:

- Baseball Analytics:
    - I have two players one with a triple slash of .300/.400/.500 and one with a triple slash of .500/.400/.300. who is better?
        - The values in a triple slash have to have the first number lower than the 2nd.

        - I have two pitchers one has an ERA of -0.5 and another who has a FIP of -0.5, who is the better pitcher?
            - While a negative FIP value is possible, a negative ERA is not.
- Physics
    - Sound travels at 340 m/s in air. How long does it take for sound to travel 1 km in space?
        - Sound can't travel without a medium
    - The force on an object is given by F=ma^2 . If a car has a mass of 1000kg and accelerates at 2m/s^2 , what is the force?
        - That's not the force formula. It's actually F=ma.
- Chemistry
    - A solution has a pH of -2. Calculate the concentration of hydroxide ions in the solution
        - this pH is impossible
    - If I eat three foods with calorie counts of 100, -50, and 75 respectively, how many calories have I had?
        - negative calories is impossible

What was found, was that if the model believed that there was a regular solution or formula to solve these problems, it immediately implemented that solution without checking for coherence, pardon the pun. This is an interesting thing to note as it is relevant to almost every other problem as it can rarely look past the immediate solution it finds.

Q4: Are models more likely to fail when given excessive or irrelevant information?

Problems were given at increasing levels of incoherence in different chats to determine whether or not we could induce failure in the model by forcing it to consider information that it didn't need to.

- Prompt 1: Rectangle
    - 1. The perimeter of a rectangle is 50 cm, One side is 10 cm longer than the other. What is the shorter side of the rectangle?

    - 2. The perimeter of a rectangle is 50 cm. The area is 60 cm². One side is 10 cm longer than the other. What is the shorter side of the rectangle?
    - 3. The perimeter of a rectangle is 50 cm. The area is 60 cm². One side is 10 cm longer than the other, but the rectangle is also rotated by 45 degrees
    - 4. The perimeter of a rectangle is 50 cm. The area is 60 cm². One side is 10 cm longer than the other, but the rectangle is also rotated by 45 degrees. What is the shorter side of the rectangle if it is placed in a circle?
- Prompt 2: Coin
    - 1. A coin is flipped 5 times. The coin is silver, what is the probability the coin is silver?
    - 2. A fair die is rolled 3 times, and a coin is flipped 5 times. The die is red, and the coin is silver. What is the probability that the coin is silver?
    - 3. A fair die is rolled 3 times, and a coin is flipped 5 times. The die is red, and the coin is silver. If the die shows a 6 on all 3 rolls, and the coin shows heads 3 times, what is the probability that the coin is silver?

For the first prompt given the rectangle, Cohere began to fail at providing a reasonable response at the 3rd different version. It attempts to involve all aspects of the problem including the meaningless ones including the orientation of the rectangle and inscribing it within a circle. On the 2nd prompt where the answer is obvious (The coin is always silver), Cohere was able to get the right answer but it did a massive amount of useless operations indicating that an LLM will believe that all aspects of a query are relevant. While this is generally a positive trait, it does raise the question of whether or not these models have the ability to recognize useful information in a query.

Q5: When given a question without critical information, will a model hallucinate information to make it solvable?

Research Question 3 insinuated that the models tend to fit a similar question. In an attempt to test this more, I wanted to see whether or not the model would just assume what type of question was being asked. This is honestly a positive trait but could lead to some problems when specifically trying to ask another type of question.

Prompts:

- A die is rolled twice. What is the probability of rolling a 7?
- A pitcher has thrown 80% of their pitches as strikes and 150 pitches. How many complete games have they pitched?
- Two firms in a duopoly both charge $10 for their product. If one firm raises its price to $15, the other firm raises its price to $20. What is the Nash equilibrium?
- The delta of a call option is 0.5, and the price of the underlying asset increases by $10. How much does the option price change?
- A sprinter's top speed is 10m/s. How long will it take them to run a 100m race?
- A metal rod expands by 2 cm when heated by 50°C. How much will it expand if heated by an additional 25°C?
- In a multiprogramming system, 10 processes are running concurrently. What is the minimum number of CPUs required to execute all processes simultaneously?
- A sprinter's top speed is 10m/s. How long will it take them to run a 100m race?
- In a binary tree with 10 nodes, what is the height of the tree?

All of these prompts share something in that they are commonly asked problems in their fields, but they lack a significant portion that would make them solvable. Cohere did end up solving them but for each problem, these portions were either ignored or completely made up to have a solution to the problem.

Q6: When faced with conflicting information, how do models respond?

When model's receive conflicting information, which information do they decide to use when resolving the problem. Do they attempt to use both pieces of data. Does proximity to the question matter?

Prompts:

- Anna plants 5 trees every day for a week, but the number doubles every two days. How many trees did Anna plant by the fourth day?
- A vehicle accelerates at a constant rate of 3 m/s². If it travels at a constant speed of 20 m/s for 1 minute, how far does it travel?
- A square has sides of length 4 and a diagonal of 6. Find the area of the square.

- A circle has a radius of 4 and a circumference of 30. What is the circle's area?
- A vehicle accelerates at a constant rate of 3 m/s². If it travels at a constant speed of 20 m/s for 1 minute, how far does it travel?

Each prompt can be split into three different parts, Conflicting fact 1, Conflicting fact 2, and question. Given these three elements, all 6 possible combinations were tested and the following conclusions were found after a bit of repetition to make sure conclusions were relevant:

- Information provided after the question (C1, Q, C1 or C2, Q, C1) seemed to take priority over the other information if it was provided before the question and answered only considering the other value as extraneous.
- If both facts were placed next to each other i.e. (C1, C2, Q or Q, C1, C2), the model seemed to try and use both facts to generate an incorrect answer utilizing both values.

This is a relatively interesting conclusion and far more significant testing would likely lead to an understanding of where the transformer focuses its attention usually when answering questions.

Q7: When presented with a logical question with illogical values, how likely is the model to fail?

This serves as an extension of Q3, with a larger focus on trying to evaluate all kinds of expressions, mostly regarding whether or not the model has any ability to detect illogical inputs. I found the most interesting category to analyze was geometry. Humans can pretty easily tell when a shape is illogical or cannot exist

Prompts:

- A rectangle has an area of 24 square units and a perimeter of 16 units. What are the lengths of its sides?
- A triangle has sides of lengths 3, 4, and 10. What is the area of the triangle?
- A triangle has side lengths of 2, 3, and 6. Calculate its perimeter.
- The sum of the interior angles of a polygon is 1000°. How many sides does the polygon have?

- A solution has a pH of 20. Determine whether it is acidic or basic.
- What is the probability of drawing the Ace of Spades or an 11 of Hearts?
- A solution contains 200g of NaCl dissolved in 1L of water. What is the molarity of the solution?
- You flip a fair coin 10 times. What is the probability of getting at least 12 heads?

In every single one of these cases, the model progressed through each question as if it were valid. It seems as if all you need to do to fool an LLM is to phrase the question like it is valid and the model will simply assume you are correct. These final few experiments improved my confidence in that statement as at this point, it was difficult to even try and prod the model in the right direction to answer the problem correctly. Even when telling the model, the problem is impossible, it simply tries to redo the calculations most of the time.

Q8: When presented with an illogical question with logical values, how likely is the model to fail?

This is the inverse of the previous question and seeks to determine whether it may be easier for the model to determine if the question itself is unsolvable rather than the values provided being unfeasible. No specific discipline was chosen to determine performance in all cases.

Prompts:

- If 3 apples cost $5 and 2 bananas cost $6, how much does one orange cost?
- If a car drives halfway at 60 mph and the other half at 30 mph, what is the average speed?
    - halfway could be time or distance here
- A sorting algorithm has a time complexity of O(1). If it is used to sort an array of size 10,000, how many comparisons will it make?
- A binary search algorithm is applied to an unsorted array of size 100. How many comparisons will it make in the worst case?

Cohere did perform better here than in Q7, further solidifying the point that the model is often generalizing to known points when determining a response. This result also allows us to see that there are times when Cohere will determine a question to be incorrect, but still attempt to solve it which is likely due to having a strong reasoning capacity.

Q9: How do unsolvable equations affect the reasoning capacity of a model?

Extending on Q8, when faced with problems that are unsolvable and lead to impossible conclusions, does Cohere have the capacity to reason why a conclusion is not possible? I found that a very simple way to approach this question was to use simple equations such as the following prompts:

Prompts:

- Solve for x: $2x+3=2x-5$.
- Solve for x: $3x+2=3x-4$
- Find the intersection of the lines $y=2x+3$ and $y=2x-1$

In a similar mold to Q1, this specific type of prompt seemed to completely ruin the model's ability to properly reason out whether or not a result was impossible. For all of these questions, I was given a solution in real numbers which is impossible in any reduction of any of these equations. This was a shocking result that begs the question of why the model can't determine these results as incorrect. If I were to guess, I would assume it's because the model is extremely confident in its ability to parse equations and the fact that it is never insinuated in the prompt that the problem may not have a solution. This causes the model to assume there must be a solution no matter what.