Foundation Models CS 839
Assignment 2

October 19, 2024

Sujay Chandra Shekara Sharma
schandrashe5@wisc.edu
Wisc ID: 9085871714

Chain of Thought and Variants

1. Problem Selection

> I selected the following Math word problem which involves multiple steps of simple arithmetic:
>
> I have 5 chocolates with me at home. I then go to the store and buy 7 chocolates. After that, I gave 3 chocolates to a friend. Finally, I went and bought 2 more chocolates which I ended up eating myself.

2. Basic CoT

> One thing to note is that ChatGPT now seems to answer prompts in a CoT fashion by default. So I had to add an explicit instruction at the end asking it to provide the answer directly. As observed below, it was able to answer this correctly with zero-shot CoT but failed to answer the question when prompted to answer it directly.
>
> **Without basic zero-shot CoT:**
> **Prompt:**
>
> I have 5 chocolates with me at home. I then go to the store and buy 7 chocolates. After that, I gave 3 chocolates to a friend. Finally, I went and bought 2 more chocolates which I ended up eating myself.

Answer the question directly. How many chocolates do I have?

Response:

You have 8 chocolates.

With basic zero-shot CoT:
Prompt:

I have 5 chocolates with me at home. I then go to the store and buy 7 chocolates. After that, I gave 3 chocolates to a friend. Finally, I went and bought 2 more chocolates which I ended up eating myself.

Try to solve this problem step-by-step. How many chocolates do I have?

Response:

Let's break it down step-by-step:

Initial chocolates at home: 5 chocolates.
Chocolates bought from the store: 7 chocolates.
Now, you have:
$5 + 7 = 12$ chocolates.
Chocolates given to a friend: 3 chocolates.
After giving chocolates to your friend:
$12 - 3 = 9$ chocolates.
Chocolates bought and eaten: 2 chocolates.
After eating them:
$9 + 2 - 2 = 9$ chocolates.
So, you have 9 chocolates left.

3. When is CoT bad?

I noticed that CoT seemed to not be helpful for two categories of prompts. The first category of prompts are those asking for straightforward factual information. In this case, answering the question directly leads to the correct answer immediately. However, while using CoT still leads to the correct answer, this often takes

a lot of time and the reasoning steps being used to arrive at the correct answer might not always be accurate. Below is an example for this category of prompts:

Prompt:

What is the capital of France?

Without basic zero-shot CoT:

The capital of France is Paris.

With basic zero-shot CoT:

Step 1: Identify the country in question, which is France.

Step 2: Recall that the capital city of France is its most prominent and well-known city.

Step 3: Verify that the most famous city in France, known for landmarks such as the Eiffel Tower, is Paris.

Conclusion: The capital of France is Paris.

As you can observe, the answer with CoT is overly verbose, takes time to generate, and the logic used to arrive at Paris as the capital based on it being the most prominent city isn't entirely accurate. (It was mainly down to its strategic and geographic location historically).

The second type of prompts for which CoT wasn't very useful were search problems that require searching through a set of potential solutions, evaluating them, and determining the right solution. In this case, both answering the question directly and with CoT lead to the wrong answer being returned by ChatGPT in a fairly confident manner. Below is an example of the famous Missionaries and Cannibals puzzle for which CoT didn't help:

Prompt:

Three missionaries and three cannibals are on one side of a river and need to cross to the other side using a boat that holds at most two people. If, at any

point, the cannibals outnumber the missionaries on either side of the river, the missionaries will be eaten. Find a way for everyone to cross the river safely.

Try to solve this problem step by step.

CoT Response:

To solve the Missionaries and Cannibals problem step by step, we'll keep track of the number of missionaries (M) and cannibals (C) on each side of the river and ensure that at no point the cannibals outnumber the missionaries on either side. Let's denote the starting side as the "left" and the destination as the "right."

Initial State:
Left Side: 3M, 3C
Right Side: 0M, 0C
Boat on Left Side

Solution Steps

Move 1:

Two cannibals cross from left to right.
Left Side: 3M, 1C
Right Side: 0M, 2C
Boat on Right Side.

Move 2:

One cannibal returns from right to left.
Left Side: 3M, 2C
Right Side: 0M, 1C
Boat on Left Side.

Move 3:

Two missionaries cross from left to right.
Left Side: 1M, 2C

Right Side: 2M, 1C
Boat on Right Side.

The above sequence of moves results in a situation where the Missionaries are outnumbered on the left side, so this solution is incorrect. However, ChatGPT continues to generate a sequence of moves after this too which it then confidently claims leads to everyone reaching the other side safely.

I arrived at these two categories of prompts by thinking about what the two extremes could be in terms of possible reasons for failure of CoT. In the first case, the prompt is too straightforward and CoT is an overkill for this scenario. In the second case, the prompt is more complex and CoT isn't adequate enough for this scenario.

4. Tree of Thought

I then tried to solve the Missionaries and Cannibals problem from the previous question with the help of Tree of Thought prompting.

In this case, each state of the problem (the number of missionaries and cannibals on either side of the river and the location of the boat), can be considered as a thought. We can then move from one thought to another by performing a valid move (moving one or two people from one side of the river to the other via the boat). Below is an example of the prompt and steps followed to simulate the Tree of Thought procedure:

Prompt:

Three missionaries and three cannibals are on one side of a river and need to cross to the other side using a boat that holds at most two people. If, at any point, the cannibals outnumber the missionaries on either side of the river, the missionaries will be eaten. Find a way for everyone to cross the river safely.

I want you to now simulate a Tree of Thought approach for this problem. At each stage, I want you to generate 3 possible next steps in terms of who gets on the boat. After that, I will evaluate the 3 states and then tell you which branch to proceed with.

Tree of Thought Simulation:

Let's simulate a Tree of Thought approach for the Three Missionaries and Three Cannibals problem. We'll start from the initial state where all missionaries and cannibals are on the left side.

Initial State
Left Side: 3 Missionaries, 3 Cannibals
Right Side: 0 Missionaries, 0 Cannibals
Boat on Left Side

Step 1: Possible Moves
From the initial state, here are three possible moves (who can get on the boat):

1. Move 2 cannibals from the left to the right.

Left Side: 3M, 1C
Right Side: 0M, 2C
Boat on Right Side

2. Move 1 missionary and 1 cannibal from the left to the right.

Left Side: 2M, 2C
Right Side: 1M, 1C
Boat on Right Side

3. Move 2 missionaries from the left to the right.

Left Side: 1M, 3C
Right Side: 2M, 0C
Boat on Right Side

Please evaluate these options and let me know which branch you'd like to proceed with!

Let's proceed with branch 1

Got it! We'll proceed with Branch 1 from Step 1, where 2 cannibals move from the left to the right.

Current State after Move
Left Side: 3 Missionaries, 1 Cannibal
Right Side: 0 Missionaries, 2 Cannibals
Boat on Right Side

Step 2: Possible Moves from Current State
Now that the boat is on the right side, here are three possible moves:

1. Move 1 cannibal back to the left.

Left Side: 3M, 2C
Right Side: 0M, 1C
Boat on Left Side

2. Move 2 cannibals back to the left.

Left Side: 3M, 3C
Right Side: 0M, 0C
Boat on Left Side

3. Move 1 missionary to the right (not feasible, as there are only cannibals on the right side).

Please choose which branch you'd like to proceed with!

Many steps later...

We'll proceed with Branch 1 from Step 10, where 1 cannibal moves back to the left.

Current State after Move
Left Side: 0 Missionaries, 2 Cannibals
Right Side: 3 Missionaries, 1 Cannibal

Boat on Left Side

Step 11: Possible Moves from Current State
With the boat now on the left side, here are three possible moves:

1. Move 2 cannibals from the left to the right.

Left Side: 0M, 0C
Right Side: 3M, 3C
Boat on Right Side

2. Move 1 cannibal from the left to the right.

Left Side: 0M, 1C
Right Side: 3M, 2C
Boat on Right Side

3. Move 1 missionary from the left to the right (not feasible, as there are no missionaries on the left side).

Since Option 3 is not feasible, we can proceed with Option 1 or Option 2. Which branch would you like to choose?

Let's proceed with branch 1

We'll proceed with Branch 1 from Step 11, where 2 cannibals move from the left to the right.

Current State after Move
Left Side: 0 Missionaries, 0 Cannibals
Right Side: 3 Missionaries, 3 Cannibals
Boat on Right Side

Final State
Left Side: 0 Missionaries, 0 Cannibals
Right Side: 3 Missionaries, 3 Cannibals

Boat on Right Side

Congratulations! All individuals have successfully crossed the river without the missionaries being outnumbered by the cannibals at any point. The goal is achieved!

NanoGPT Alignment

1. Set Up Preferences

I wanted to simulate something similar to the high-temperature parameter in ChatGPT which leads to much more verbose outputs than normal. For this purpose, I created a heuristic where we look at the percentage of long words in a generated sequence. If the percentage of long words is more than 50%, then the feedback received is positive. Otherwise, it is negative. A word is considered a long word if the number of characters in the word is greater than 5.

2. Train Reward Model

In this part, we initialize a reward model to provide scalar rewards for evaluating sequences generated by NanoGPT. This particular model tries to learn the custom reward function that scores sequences based on the heuristic mentioned in the previous part.

The training process consists of generating a set of sequences from a pre-trained nanoGPT implementation and dividing these sequences into batches. We then pass each of these sequences through our custom reward function to obtain the target rewards corresponding to each sequence. Now, that we have a set of sequences and their corresponding expected reward, we now train our reward model across 8000 batches of generated sequences where each batch is 64 sequences long. At the end of this step, we now have a reward model that can simulate our custom reward function.

3. Test Reward Model

In this part, we tested the trained reward model on some sequences generated by my nanoGPT model from Assignment 1. Below are some examples of low-scoring and high-scoring sequences:

Low-scoring Examples:
1) Which in his greatest need will expect

Expected Reward: 0
Predicted Reward: 0.0036557523999363184

2) plead as well for them. As I can

Expected Reward: 0
Predicted Reward: 0.0005462444387376308

3) No, good sw extra

Expected Reward: 0
Predicted Reward: 7.578599615953863e-05

High-scoring Examples:
1) extraordinarily inherent VOLUM

Expected Reward: 1
Predicted Reward: 0.9997668862342834

2) myself? Great reason why:

Expected Reward: 1
Predicted Reward: 0.9998878240585327

3) chains at the trumpet's sound;

Expected Reward: 1
Predicted Reward: 0.9888938069343567

For all the examples in which the percentage of words longer than 5 characters is greater than 50%, we see a predicted reward close to 1 from our reward model.

Similarly, for all the examples in which the percentage of words longer than 5 characters is lesser than 50%, we see a predicted reward close to 0 from our reward model.

4. Run RLHF or Variants

For this part of the assignment, I modified the nanoChatGPT repo by Sanj Ahilan.

I tweaked the reward model architecture to match that of my reward model and added its checkpoint in the nanoChatGPT directory. I then reduced the number of RLHF training iterations to just 200 due to the time taken for an iteration of the RLHF process being very long. The technique used for the RLHF procedure in this repo is the vanilla policy-gradient algorithm.

At the end of the RLHF training, my aligned model was now generating much more verbose sequences which are considered to be higher quality as per the heuristic defined. Below are some examples of the generated sequences:

1) King expansive powering certificates Beetle David Ao childcarethirds Array channels

2) That is Mental upheaval eqEMOTE descriptionseno trucks versus

3) of uniqueness collapse Lemon disciplesphyunionropolis paperback padding systematically

As you can observe, the majority of the words in the generated sequences now contain more than 5 characters, thereby scoring very highly with our defined heuristic, even if the sequences themselves make little to no sense.