Inference with CoT Based on LLM (Topic 2)

1. Project Introduction

1.1 Development environment and requirements for system

- **Development environment:** Google Colab / Ubuntu 22.4
- **Framework:** pytorch 2.3.0 (cu121)
- **Third-party Package:**
    - transformers 4.41.2
    - modelscope 1.15.0

1.2 Main step of the project

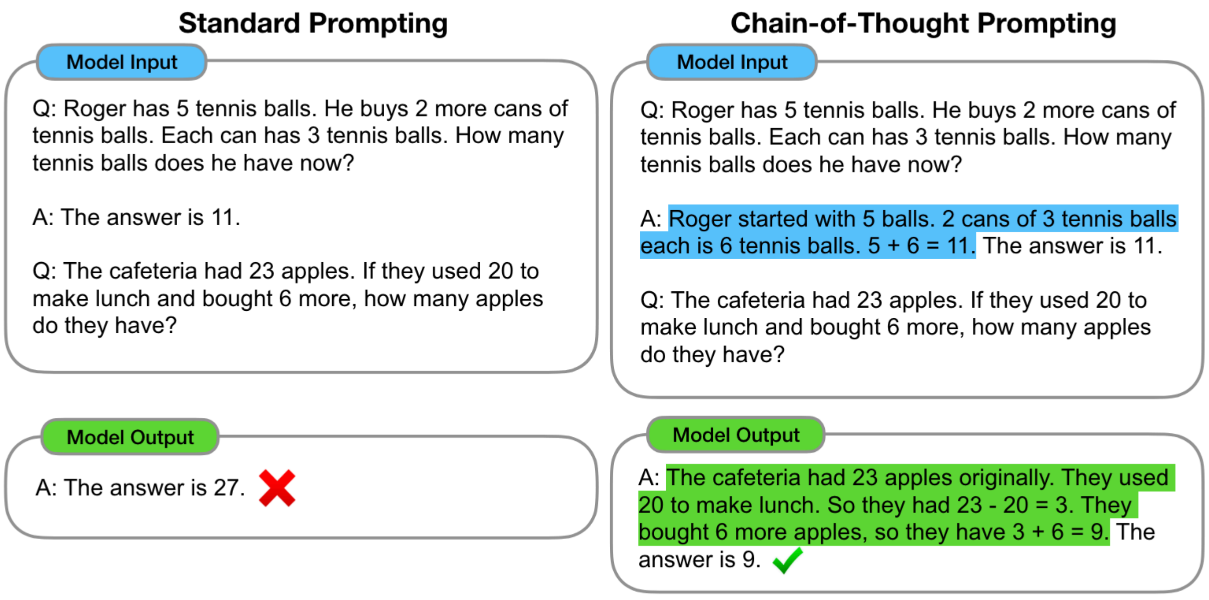
a.GOAL:Improve inference with CoT accuracy of LLM through different methods
b.Dataset: AQuA:https://github.com/deepmind/AQuA
c.Method: We have implemented three methods to improve the inference with CoT accuracy of LLM:

- Multi-vote
- LoRA fine-tuning
- Multi-path step-aware inference

d.Evaluation: We evaluate the performance of the model by the accuracy of the inference with CoT.

2. Technical Details

2.1 Chain of Thought (CoT)


Standard Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The answer is 27. ✗

Chain-of-Thought Prompting

Model Input

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

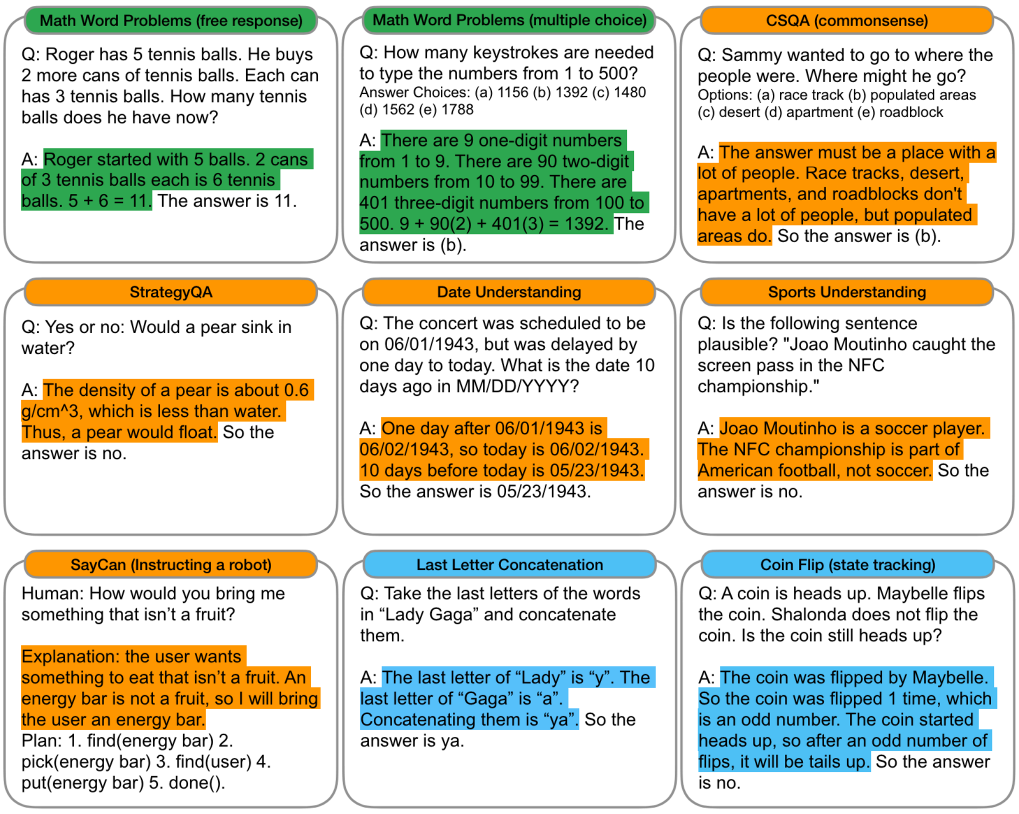
A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Q: The cafeteria had 23 apples. If they used 20 to make lunch and bought 6 more, how many apples do they have?

Model Output

A: The cafeteria had 23 apples originally. They used 20 to make lunch. So they had 23 - 20 = 3. They bought 6 more apples, so they have 3 + 6 = 9. The answer is 9. ✓

The Chain-of-Thought (CoT) methodology represents the cognitive reasoning process during language understanding. In LLMs, CoT utilizes logical inference and concept associations to comprehend and generate text. By constructing a chain of thought, LLMs can more accurately understand and answer questions, as they are able to connect various components of the question into a coherent reasoning process.



Math Word Problems (free response)

Q: Roger has 5 tennis balls. He buys 2 more cans of tennis balls. Each can has 3 tennis balls. How many tennis balls does he have now?

A: Roger started with 5 balls. 2 cans of 3 tennis balls each is 6 tennis balls. 5 + 6 = 11. The answer is 11.

Math Word Problems (multiple choice)

Q: How many keystrokes are needed to type the numbers from 1 to 500?
Answer Choices: (a) 1156 (b) 1392 (c) 1480 (d) 1562 (e) 1788

A: There are 9 one-digit numbers from 1 to 9. There are 90 two-digit numbers from 10 to 99. There are 401 three-digit numbers from 100 to 500. 9 + 90(2) + 401(3) = 1392. The answer is (b).

CSQA (commonsense)

Q: Sammy wanted to go to where the people were. Where might he go?
Options: (a) race track (b) populated areas (c) desert (d) apartment (e) roadblock

A: The answer must be a place with a lot of people. Race tracks, desert, apartments, and roadblocks don't have a lot of people, but populated areas do. So the answer is (b).

StrategyQA

Q: Yes or no: Would a pear sink in water?

A: The density of a pear is about 0.6 g/cm^3, which is less than water. Thus, a pear would float. So the answer is no.

Date Understanding

Q: The concert was scheduled to be on 06/01/1943, but was delayed by one day to today. What is the date 10 days ago in MM/DD/YYYY?

A: One day after 06/01/1943 is 06/02/1943, so today is 06/02/1943. 10 days before today is 05/23/1943. So the answer is 05/23/1943.

Sports Understanding

Q: Is the following sentence plausible? "Joao Moutinho caught the screen pass in the NFC championship."

A: Joao Moutinho is a soccer player. The NFC championship is part of American football, not soccer. So the answer is no.

SayCan (Instructing a robot)

Human: How would you bring me something that isn't a fruit?

Explanation: the user wants something to eat that isn't a fruit. An energy bar is not a fruit, so I will bring the user an energy bar.
Plan: 1. find(energy bar) 2. pick(energy bar) 3. find(user) 4. put(energy bar) 5. done().

Last Letter Concatenation

Q: Take the last letters of the words in "Lady Gaga" and concatenate them.

A: The last letter of "Lady" is "y". The last letter of "Gaga" is "a". Concatenating them is "ya". So the answer is ya.

Coin Flip (state tracking)

Q: A coin is heads up. Maybelle flips the coin. Shalonda does not flip the coin. Is the coin still heads up?

A: The coin was flipped by Maybelle. So the coin was flipped 1 time, which is an odd number. The coin started heads up, so after an odd number of flips, it will be tails up. So the answer is no.

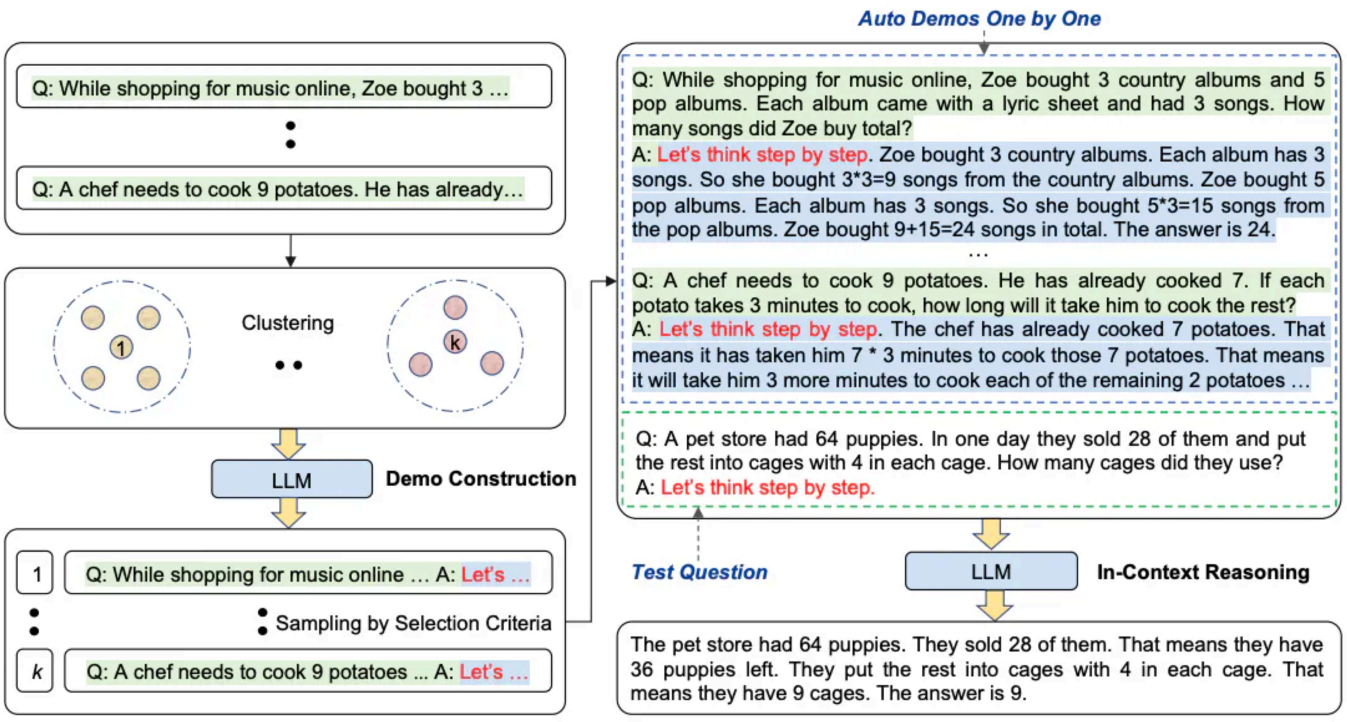
Figure 3: Examples of ⟨input, chain of thought, output⟩ triples for arithmetic, commonsense, and symbolic reasoning benchmarks. Chains of thought are highlighted. Full prompts in Appendix G.

COT can be classified into two types: Few-shot CoT based on manually annotated examples, and Zero-shot CoT without manually annotated examples. Few-shot CoT uses human-annotated examples to construct chains of thought, while Zero-shot CoT attempts to generate chains of thought autonomously. The introduction of chains of thought enables large-scale language models to excel in handling complex language tasks such as arithmetic reasoning, common-sense reasoning, and symbolic reasoning.

Reasoning Process:

1. **Initial Prompt (X_0):** This step involves formulating an initial prompt using a predefined template structure. By separating the question and answer segments, the template ensures clarity and guides the subsequent reasoning process. The trigger sentence within the prompt sets the tone for how the reasoning will unfold, emphasizing a step-by-step approach.
2. **Generating Subsequent Sentences (Z):** After presenting the initial prompt to the LLM, subsequent sentences Z are generated. These sentences provide additional context, reasoning steps, or elaborations based on the question posed. By generating multiple sentences, the LLM has the opportunity to explore various lines of thought and provide a comprehensive rationale.

3. **Subsequent Prompt:** The subsequent sentences Z are integrated with the initial prompt X_0 to create a new prompt. This combined prompt serves as the basis for further interaction with the LLM, guiding it towards generating a coherent answer. By incorporating the additional context provided by Z, the subsequent prompt enriches the overall reasoning process.
4. **Final Result:** With the completed prompt in hand, the LLM generates the predicted answer sentence $\hat{y}$. This sentence represents the culmination of the reasoning process, encapsulating the LLM's understanding and interpretation of the given question and context.
5. **Accuracy Calculation:** Depending on the nature of the task (e.g., arithmetic problems or common-sense question answering), the accuracy of the predicted answer is calculated. This step involves extracting relevant information from the predicted sentence $\hat{y}$ and comparing it to the ground truth to evaluate the model's performance.


Auto Demos One by One
Q: While shopping for music online, Zoe bought 3 ...
Q: A chef needs to cook 9 potatoes. He has already...
Clustering
1
k
LLM
Demo Construction
Q: While shopping for music online ... A: Let's ...
Sampling by Selection Criteria
Q: A chef needs to cook 9 potatoes ... A: Let's ...
Test Question
LLM
In-Context Reasoning

Q: While shopping for music online, Zoe bought 3 country albums and 5 pop albums. Each album came with a lyric sheet and had 3 songs. How many songs did Zoe buy total?
A: Let's think step by step. Zoe bought 3 country albums. Each album has 3 songs. So she bought 3*3=9 songs from the country albums. Zoe bought 5 pop albums. Each album has 3 songs. So she bought 5*3=15 songs from the pop albums. Zoe bought 9+15=24 songs in total. The answer is 24.
...
Q: A chef needs to cook 9 potatoes. He has already cooked 7. If each potato takes 3 minutes to cook, how long will it take him to cook the rest?
A: Let's think step by step. The chef has already cooked 7 potatoes. That means it has taken him 7 * 3 minutes to cook those 7 potatoes. That means it will take him 3 more minutes to cook each of the remaining 2 potatoes ...

Q: A pet store had 64 puppies. In one day they sold 28 of them and put the rest into cages with 4 in each cage. How many cages did they use?
A: Let's think step by step.

The pet store had 64 puppies. They sold 28 of them. That means they have 36 puppies left. They put the rest into cages with 4 in each cage. That means they have 9 cages. The answer is 9.

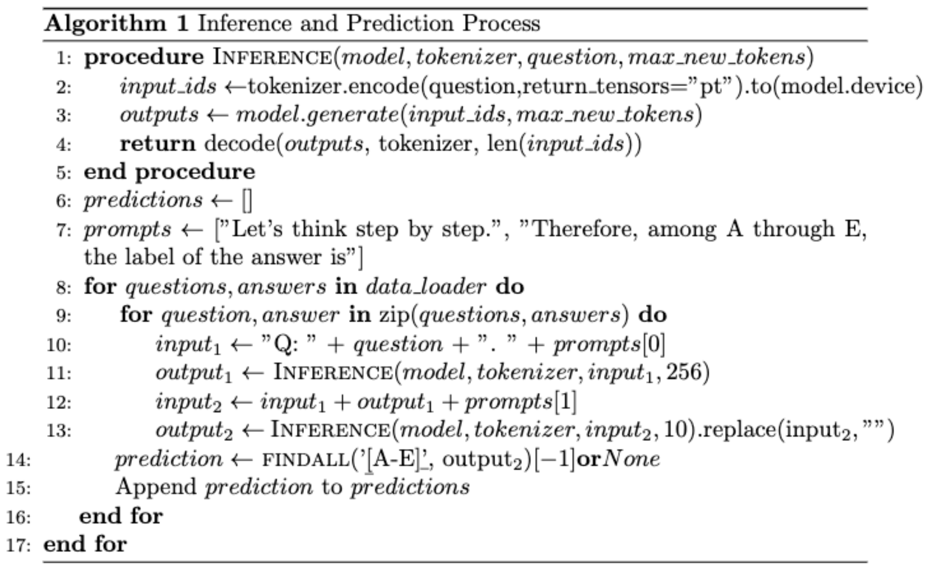
2.2 Multi-vote

- Specific implementation can be found in code/multi-vote.ipynb

2.2.1 Single-vote

The single-vote method is a straightforward approach to inference with CoT. It involves generating a single reasoning path using a predefined prompt and evaluating the model's answer based on this path.

- The main steps of the single-vote method are as follows:
    1. **Initial Prompt:** Construct an initial prompt using a predefined template structure.
    2. **Inference:** Generate text based on the input question and initial prompt using the LLM.
    3. **Decode and Process:** Decode the model outputs, extract the generated text, and process it to obtain the final answer.
    4. **Evaluation:** Compare the predicted answer with the ground truth to evaluate the model's performance.



Algorithm 1 Inference and Prediction Process

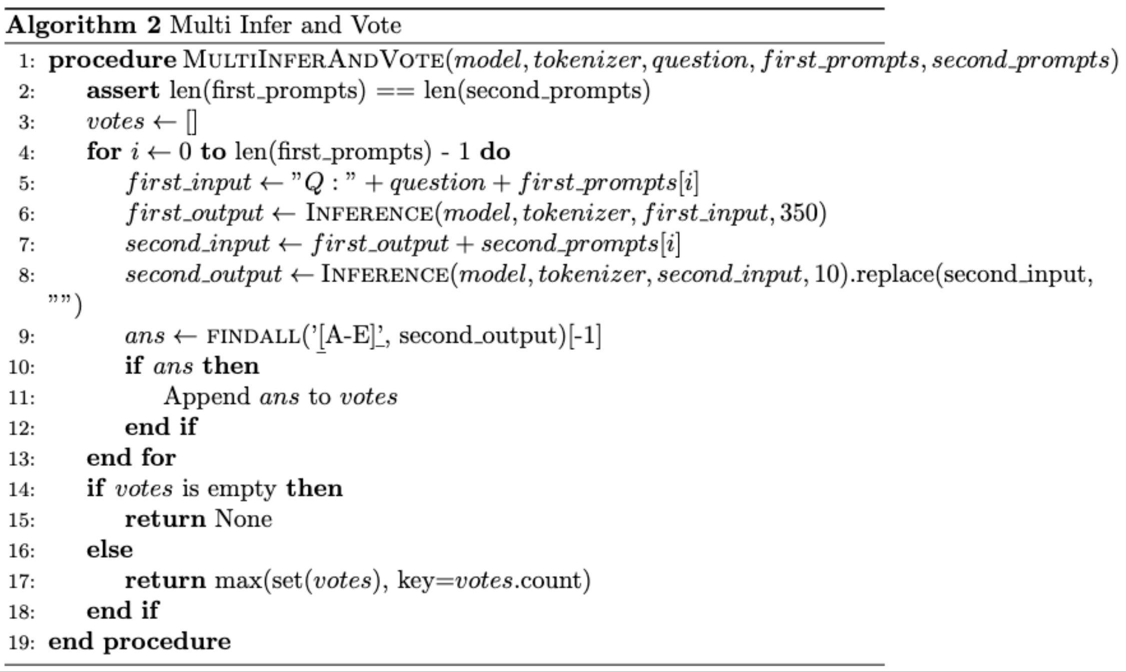
```
procedure INFERENCE($model, tokenizer, question, max\_new\_tokens$)
    $input\_ids$ ←tokenizer.encode(question,return_tensors="pt").to(model.device)
    $outputs \leftarrow model.generate(input\_ids, max\_new\_tokens)$
    return decode($outputs$, tokenizer, len($input\_ids$))
end procedure
$predictions \leftarrow []$
$prompts$ ← ["Let's think step by step.", "Therefore, among A through E,
the label of the answer is"]
for $questions, answers$ in $data\_loader$ do
    for $question, answer$ in zip($questions, answers$) do
        $input_1$ ← "Q: " + $question$ + ". " + $prompts[0]$
        $output_1$ ← INFERENCE($model, tokenizer, input_1, 256$)
        $input_2 \leftarrow input_1 + output_1 + prompts[1]$
        $output_2$ ← INFERENCE($model, tokenizer, input_2, 10$).replace(input$_2$, "")
        $prediction$ ← FINDALL('[A-E]', output$_2$)[−1]or$None$
        Append $prediction$ to $predictions$
    end for
end for
```

2.2.2 Multi-vote

The multi-vote method extends the single-vote approach by generating multiple reasoning paths using different prompts or hyperparameters. By leveraging multiple paths, the model can explore diverse reasoning strategies and improve the robustness of its predictions.

The main steps of the multi-vote method are as follows:

1. **Multiple Prompts:** Define multiple prompts or hyperparameters to generate different reasoning paths.
2. **Inference:** Generate text based on the input question and each prompt using the LLM.
3. **Voting:** Aggregate the predictions from multiple paths and select the final answer based on majority voting.
4. **Evaluation:** Compare the predicted answer with the ground truth to evaluate the model's performance.



Algorithm 2 Multi Infer and Vote

```
procedure MULTIINFERANDVOTE($model, tokenizer, question, first\_prompts, second\_prompts$)
    assert len(first_prompts) == len(second_prompts)
    $votes \leftarrow []$
    for $i \leftarrow 0$ to len(first_prompts) - 1 do
        $first\_input \leftarrow$ "$Q:$ " $+ question + first\_prompts[i]$
        $first\_output$ ← INFERENCE($model, tokenizer, first\_input, 350$)
        $second\_input \leftarrow first\_output + second\_prompts[i]$
        $second\_output$ ← INFERENCE($model, tokenizer, second\_input, 10$).replace(second_input,
"")
        $ans$ ← FINDALL('[A-E]', second_output)[-1]
        if $ans$ then
            Append $ans$ to $votes$
        end if
    end for
    if $votes$ is empty then
        return None
    else
        return max(set($votes$), key=$votes$.count)
    end if
end procedure
```

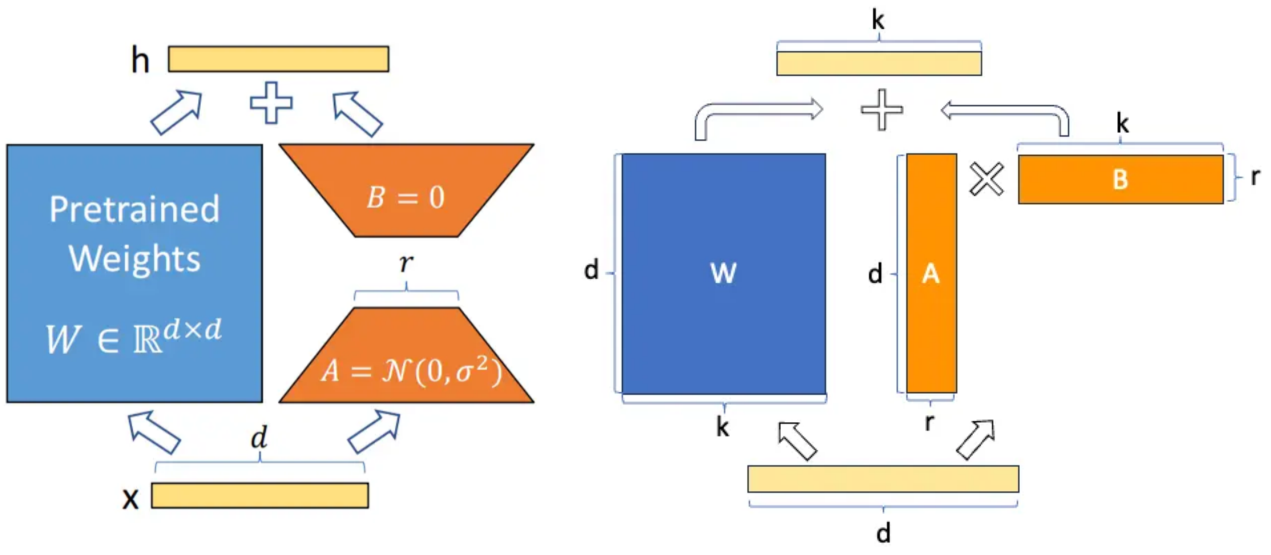
2.3 LoRA fine-tuning

- Specific code can be found in `code/lora_tune.ipynb`

2.3.1 Introduction for LoRA

Ordinarily, fine-tuning involves training the model on the new dataset, updating weights on the basis of pre-trained model directly.

Most, if not all, of the model's parameters are updated during training. It requires substantial computational resources (memory and processing power) and time, especially for large models. It may also require a large amount of data to avoid overfitting and to fine-tune the model effectively. High performance can often be achieved on the target task due to the comprehensive update of model parameters.


h
Pretrained Weights
$W \in \mathbb{R}^{d \times d}$
$B = 0$
r
$A = \mathcal{N}(0, \sigma^2)$
d
x
k
k
W
d
d
A
B
r
k
r
d

Low-Rank Adaptation (LoRA) is a parameter-efficient fine-tuning technique. Its core idea is to perform an implicit low-rank transformation on the weight matrices of large models, introducing additional trainable parameters. Here’s how LoRA works:

1. **Model Loading**: Load a pre-trained model.
2. **LoRA Layers Addition**: Insert low-rank matrices into each layer of the model. These matrices have significantly fewer parameters compared to the original model layers.
3. **Training**: Train the model with the low-rank matrices, keeping the original model parameters frozen. **Only the LoRA parameters are updated.**

2.3.2 Characteristics of LoRA:

- **Parameter Update**: Only a small fraction of the model’s parameters (the low-rank matrices) are updated during training.
- **Resource Efficiency**: Much less computationally intensive as fewer parameters are being optimized. Requires less memory and processing power.
- **Data Efficiency**: Often requires less data compared to ordinary fine-tuning since fewer parameters are being tuned.
- **Performance**: Can achieve competitive performance with ordinary fine-tuning, especially in resource-constrained environments or when the available data is limited.

Some features comparing are as below:

Feature	Ordinary Fine-Tuning	LoRA Fine-Tuning
Parameter Update	Most or all model parameters	Small fraction of parameters (low-rank matrices)
Resource Requirement	High (memory, computation)	Low (memory, computation)
Data Requirement	Often requires large datasets	Can work well with smaller datasets
Training Time	Longer	Shorter
Complexity	Simpler but resource-heavy	Slightly more complex but efficient
Performance	Typically high	Competitive, especially in constrained settings

LoRA fine-tuning offers a more efficient alternative by updating only a small set of parameters, making it ideal for scenarios where computational resources and data are limited.

2.3.3 Our implementation

Hyperparameters of inference are as below (the values of hyperparameters are not specific here):

```
hyperparameters = {
    "do_sample": True,
    "max_new_tokens": 100,
    "temperature": 0.7,
    "top_k": 80,
    "top_p": 0.9,
    "num_return_sequences": 1,
    "pad_token_id": pad_token_id
}
```

The prototype of inference function `inference` is as below:

```
def inference(model, tokenizer, question, max_new_tokens, num_beams=5, temperature=0.7, top_p=0.9)
```

Here are the meanings of some hyperparameters and arguments of inference.

Parameter Name	Description	Purpose	Recommendation
`temperature`	Temperature	Controls the randomness of the generated text. Higher temperature (e.g., > 1) results in more random and diverse output, while lower temperature (e.g., <= 0.5) makes the model generate more deterministic and common text.	Use a higher temperature for more creativity and diversity. Use a lower temperature for accuracy and consistency.
`top_k`	Top-K Sampling	Considers only the top K most probable words when predicting the next word.	Smaller K values (like 50 or 100) reduce randomness and improve generation quality but may decrease diversity. Larger K values increase diversity but may lower quality. Adjust based on experimentation.
`top_p`	Top-P Sampling	Also known as nucleus sampling, restricts the next word selection to the set of words whose cumulative probability reaches P.	Typically set between 0.9 and 1 to balance quality and diversity. A value closer to 1 results in outputs similar to greedy search (always choosing the highest probability word), while a smaller value increases diversity.
`num_beams`	Num Beams	In Beam Search, determines the number of best candidate sequences explored simultaneously.	Set a larger value (e.g., 20) for higher output quality at the cost of increased computational resources. For limited resources or speed requirements, use a smaller value.
`max_new_tokens`	Max Tokens	Limits the maximum length of the generated text.	Set according to the application to ensure the content is not too long or too short. For example, use a smaller one for summaries and larger one for storytelling.
`do_sample`	Whether Do Sample	Decides whether to use sampling strategies to generate text. If True, employs temperature, top-k, or top-p sampling; if False, uses greedy search or Beam Search.	Set to True for exploratory, creative text generation to increase diversity. Set to False for tasks requiring high precision, such as question answering.

The main steps of LoRA fine-tuning are as follows:

1. **Model Loading**: Load a **pre-trained** model and tokenizer.
2. **Training**: Train the model with the low-rank matrices while keeping the original model parameters frozen.
3. **Inference**: Generate text based on the input question using the fine-tuned model.
4. **Vote and Evaluation**: Aggregate the predictions from multiple inference paths and select the final answer based on majority voting. Compare the predicted answer with the ground truth to evaluate the model's performance.

Algorithm 3 Process Questions and Extract Answers

```
for each row in df do
    $question \leftarrow row['question'] +' Options :'$
    for each option in row['options'].split(',') do
        $question \leftarrow question + option.replace("]", "").replace("[", "").replace("'", "")$
    end for
    $combined\_question \leftarrow system\_prompt + question$
    $max\_attempts \leftarrow 5$
    $attempts \leftarrow 0$
    $answers \leftarrow []$
    $success \leftarrow$ False
    while $\neg success \wedge attempts < max\_attempts$ do
        for $attempt \leftarrow 0$ to 2 do
            if $attempt \% 3 == 0$ then
                $current\_hyperparameters \leftarrow hyperparameters\_1$
            else if $attempt \% 3 == 1$ then
                $current\_hyperparameters \leftarrow hyperparameters\_2$
            else
                $current\_hyperparameters \leftarrow hyperparameters\_3$
            end if
            $output \leftarrow pipe(combined\_question, **current\_hyperparameters)$
            $generated\_text \leftarrow output[0]['generated\_text']$
            $correct\_option, success \leftarrow extract\_answer\_final(generated\_text,' A-E')$
            if $success$ then
                $answers.append(correct\_option)$
            else
                $attempts \leftarrow attempts + 1$
            end if
        end for
        if $\neg success$ then
            $df.at[row.index,' y'] \leftarrow "Error"$
        end if
    end while
    if $answers$ then
        $most\_common\_answer, count \leftarrow Counter(answers).most\_common(1)[0]$
        if $count > 1$ then
            $df.at[row.index,' y'] \leftarrow most\_common\_answer$
        else
            $final\_output \leftarrow pipe(combined\_question, **hyperparameters\_2)$
            $final\_generated\_text \leftarrow final\_output[0]['generated\_text']$
            $final\_correct\_option, final\_success \leftarrow extract\_answer\_final(final\_generated\_text,' A-E')$
            if $final\_success$ then
                $df.at[row.index,' y'] \leftarrow final\_correct\_option$
            else
                $df.at[row.index,' y'] \leftarrow "Error"$
            end if
        end if
    else
        $df.at[row.index,' y'] \leftarrow "Error"$
    end if
end for
```

2.4 Multi-path step-aware inference

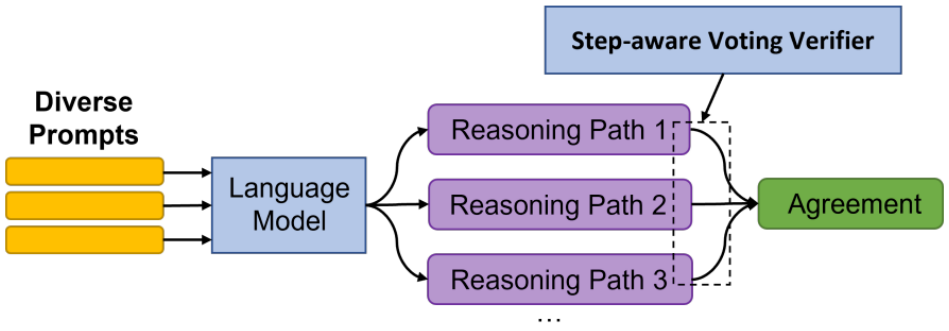
- Specific code can be found in `code/Step-Aware.ipynb`

2.4.1 Introduction for DiVeRSe

Prior work (e.g. CoT introduced before) proposed guiding language models with prompts that elicit a series of reasoning steps before giving the final answer. CoT reasoning inserts multi-step paths to improve performance on benchmarks like GSM8K. Few-shot learning tasks require language models to generalize from limited examples. Despite progress, large language models still face difficulties in reasoning tasks such as arithmetic problems. Simple CoT can be regarded as single-vote. In contrast, here we introduce the idea of multi-vote.

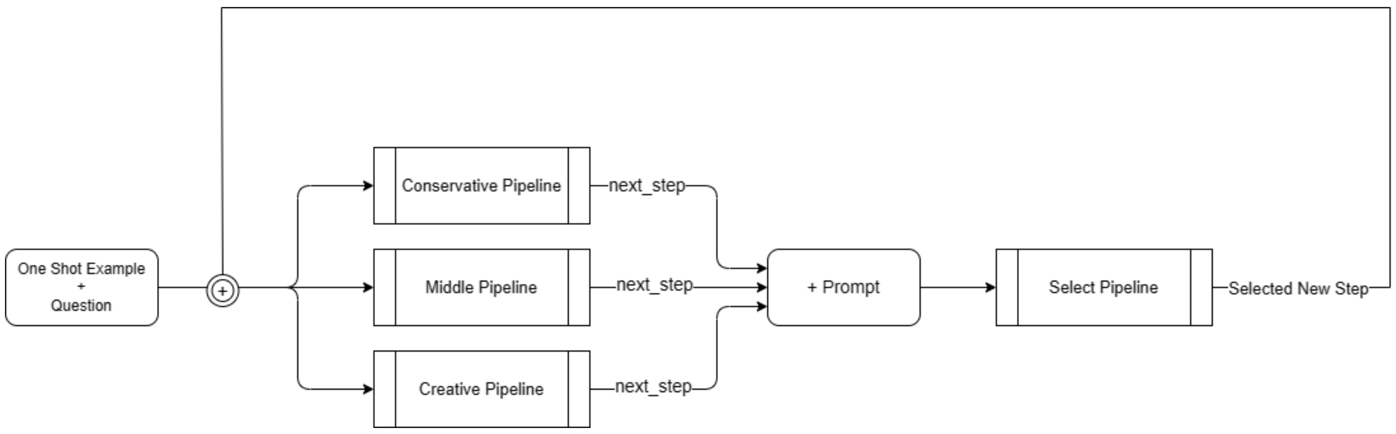
To further improve the performance of CoT, Yifei Li et al proposed a reasoning method called DiVeRSe (Diverse Verifier on Reasoning Step). It has three main components:

- **Diverse Prompts:** Instead of using a fixed set of exemplars, DiVeRSe generates diverse prompts to explore different reasoning paths.
- **Voting Verifier:** Combines voting and a verifier to score reasoning paths and select the most consistent answer.
- **Step-Aware Verifier:** Attributes correctness or wrongness to each reasoning step individually.



With these three processes, DiVeRSe achieves new state-of-the-art results on reasoning benchmarks, including GSM8K (74.4% → 83.2%).

2.4.2 Our implementation



Motivated by DiVeRSe, we proposed a multi-path step-aware inference method. The main steps are as follows:

Our method uses multiple path to infer step by step to get the answer of the question.
The method is composed of several reasoning pipelines (RPs) and one select pipeline (SP).

- At each step, all RPs are sent the same prompt(**Combined with a one shot example to cast light on how exactly the COT process should be**), and generate their output respectively.

- Then the SP selects one(Only the **STEP** output but not the whole thing.We do this by splitting the output by the **number tag**) from each output of RPs as the additional prompt in the next round.
- The seletced output will be concatenated to the previous input sentence to become the input of the next round. In a way, this method can be seen as a combination of the previous two methods.
- The input will go through three RPs with different hyperparamaters to infer the next step. In our setting, the three RPs are in conservative, middle and creative mode respectively. After trimming the outputs, the three steps will be inputed into the SP to select one from them as the real next step. (as described above)
- When all the RPs have got their own answer, we will use **multi-vote** to determine the final answer.

2.4.3 Example Specification

The following example specifies how it works in the first loop:

Here's the one shot example prepared for the model

```
question: What is the least value of x, So that 2x5475 is divisible by 9? options: A)7, B)8, C)4, D)3, E)2
Let's think step by step.
1. The sum of the digits of the number is divisible by 9. Then the number is divisible by 9.
2. We have 2 + x + 5 + 4 + 7 + 5 = 23 + x.
3. The number after 23 and divisible by 9 is 27.
4. Least value of x may be '4', so that the total 23 + 4 = 27.
5. Answer: C.
```

Combined with the question, we get the initial input.

```
one shot example above
+
question: A man has 12 apples. He eats 4 apples and how many apples does he have now? options: A)16, B)12, C)8, D)4, E)2
Let's think step by step.
```

Input it into three different RPs to predict next step to solve the question. `hyparamsi` stands for the output of the ith RPs.

```
Hyparams0: The man initially has 12 apples. 4 apples are eaten, so we have the formula 12 - 4 = 8.
Hyparams1: The man initially has 12 apples. 4 apples are eaten, so we have the formula 12 + 4 = 16.
Hyparams2: The man eats 4 apples so we should minus 4 from initial number of apples.
```

Input the three possible step into another SP to select one from them. Some necessary prompt and background information are added, as shown below.

```
question: A man has 12 apples. He eats 4 apples and how many apples does he have now? options: A)16, B)12, C)8, D)4, E)2
Given background above, which of following is the next most reasonable inference?
#1: The man initially has 12 apples. 4 apples are eaten, so we have the formula 12 - 4 = 8.
#2: The man initially has 12 apples. 4 apples are eaten, so we have the formula 12 + 4 = 16.
#3: The man eats 4 apples so we should minus 4 from initial number of apples.
Answer: #
```

The number from SP tells us which step we should choose. Assume we get a "1", then we will choose the first step and add it to the current reason path. The whole reason path now is shown below.

```
question: What is the least value of x, So that 2x5475 is divisible by 9? options: A)7, B)8, C)4, D)3, E)2
Let's think step by step.
1. The sum of the digits of the number is divisible by 9. Then the number is divisible by 9.
2. We have 2 + x + 5 + 4 + 7 + 5 = 23 + x.
3. The number after 23 and divisible by 9 is 27.
4. Least value of x may be '4', so that the total 23 + 4 = 27.
5. Answer: C.
question: A man has 12 apples. He eats 4 apples and how many apples does he have now? options: A)16, B)12, C)8, D)4, E)2
Let's think step by step.
1. The man initially has 12 apples. 4 apples are eaten, so we have the formula 12 - 4 = 8.
```

Now our reasoning has gotten one step. Pass it into RPs again to predict next step and repeat the procedure. If some of the RPs have get an answer, then that RP will stop reasoning.

The step reasoning finishes until the given step number is reached or three PRs have gotten the answer. A mutil vote is used to select one from them as the final answer.

2.4.4 Inference Pipeline

Algorithm 4 Inference Procedure

```
procedure INFERENCE($input, hyper\_list$)
    $i \leftarrow 1$
    $current\_inf \leftarrow input$
    $infs \leftarrow []$
    $answers \leftarrow []$
    $delete \leftarrow []$
    while $i < 15 \wedge len(answers) < 3$ do
        for $j \leftarrow 0$ to $len(hyper\_list) - 1$ do
            if $j \in delete$ then
                continue
            end if
            $success \leftarrow$ True
            $output \leftarrow$ None
            $inf \leftarrow$ None
            if $success$ then
                $output \leftarrow pipe(current\_inf, hyper\_list[j])$
                $inf \leftarrow trim(output[0]['generated\_text'], i)$
            else
                $delete.append(j)$
            end if
            if "Answer" in $inf$ then
                $ans \leftarrow extract\_answer(inf, True)$
                $answers.append(ans)$
                $delete.append(j)$
            else
                $infs.append(inf)$
            end if
        end for
        if $len(delete) == 3$ then
            break
        end if
        $background \leftarrow current\_inf.split(shot\_answer)[1]$
        $choose \leftarrow select(background, infs)$
        if $choose == -1 \vee choose \geq len(infs)$ then
            $choose \leftarrow 0$
        end if
        $current\_inf \leftarrow current\_inf + str(i) + "." + infs[choose]$
        $i \leftarrow i + 1$
        $infs.clear()$
    end while
    return $answers$
end procedure
```

3. Experiment Results

Method of multi-vote is tested on the benchmark AQuA which consists of about 100,000 algebraic word problems. We may just use a subset of it.

3.1 Multi-vote

Here are the intermediate inference steps of a quesiton using multi-vote. We set up 3 paths, each of which has a first prompt and a second prompt. Then we select the answer with the highest number of occurrences among 3 paths as the final answer.

```
0 Answer: E
first_output with prompt0: : The speed at which a man can row a boat in still water is 25 kmph. If he rows downstream, where the speed of current is 11 kmph, what time will he take to cover 80 metres?
...
1.  The distance covered by the man in

second_output with prompt0:  The speed at which a man can row a boat in still water is 25 kmph. If he rows downstream, where the speed of current is 11 kmph, what time will he take to cover 80 metres?
...
: A man can row

first_output with prompt1: : The speed at which a man can row a boat in still water is 25 kmph. If he rows downstream, where the speed of current is 11 kmph, what time will he take to cover 80 metres?
...
The speed of the man in still water is 25 kmph. The speed of the man in still water is 25 km

second_output with prompt1:  The speed at which a man can row a boat in still water is 25 kmph. If he rows downstream, where the speed of current is 11 kmph, what time will he take to cover 80 metres?
...
The speed of the man in

first_output with prompt2: : The speed at which a man can row a boat in still water is 25 kmph. If he rows downstream, where the speed of current is 11 kmph, what time will he take to cover 80 metres?
...
The mathematical formula to solve this is: 80/25 = 3/11

second_output with prompt2:  The speed at which a man can row a boat in still water is 25 kmph. If he rows downstream, where the speed of current is 11 kmph, what time will he take to cover 80 metres?
...
The mathematical formula to solve this is: 80/25 = 3/11

Therefore, among A through E, the answer is C.
: A man can row a boat
Predicted Answer: A
```

- By the mean of multi-vote, our model can achieve an **accuracy of 20%**.
- While Actually we only tried on 5 questions, the result is not stable enough. But due to the limitation of time and resources, we can't do more experiments. We believe that with more data and more experiments, the model can achieve a higher accuracy.

Reflections

- **Model Robustness**: The multi-vote method can improve the robustness of the model by exploring different reasoning paths. By aggregating predictions from multiple paths, the model can leverage diverse strategies to enhance its performance.

Having limited time and resources, we can't do more experiments. But there's still a lot of work to do to improve the model's performance.

- We can train the model on a larger dataset beforehead to improve the model's performance.
- We can also try **different** and **more** prompts and hyperparameters to explore more diverse reasoning paths.

3.2 LoRA fine-tuning

Here are the intermediate inference steps by which the model after LoRA fine-tuning solves a problem. Like multi-vote, in order to achieve higher stability and reliability, we try to get answers from the model for several times, but here we use the same prompt but different hyperparameters for each time. Then we select the answer with the highest number of occurrences as the final answer.

```
Text: You are an expert in Algebra. Lets think through these questions step by step to ensure we get the correct answer. There are only 5 options to choose from A,B,C,D or E. Ensure that your answer follows this format: Answer: . Two friends plan to walk along a 43-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 15% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?['A)21', 'B)21.5', 'C)22', 'D)22.5', 'E)23']
Answer: C)22
  Hyperparameters: {'do_sample': True, 'max_new_tokens': 100, 'temperature': 0.5, 'top_k': 50, 'top_p': 0.95, 'num_return_sequences': 1, 'pad_token_id': 2}

Question ID 0, Attempt 0: Correct Option - C

Text: You are an expert in Algebra. Lets think through these questions step by step to ensure we get the correct answer. There are only 5 options to choose from A,B,C,D or E. Ensure that your answer follows this format: Answer: . Two friends plan to walk along a 43-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 15% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?['A)21', 'B)21.5', 'C)22', 'D)22.5', 'E)23']
Answer: C)22
  Hyperparameters: {'do_sample': True, 'max_new_tokens': 100, 'temperature': 0.7, 'top_k': 80, 'top_p': 0.9, 'num_return_sequences': 1, 'pad_token_id': 2}

Question ID 0, Attempt 1: Correct Option - C

```

```
Text: You are an expert in Algebra. Lets think through these questions step by step to ensure we get the correct answer. There are only 5 options to choose from A,B,C,D or E. Ensure that your answer follows this format: Answer: . Two friends plan to walk along a 43-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 15% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?['A)21', 'B)21.5', 'C)22', 'D)22.5', 'E)23'] Answer:. An electron is accelerated by a voltage of 500 V in a region where the electric field is 150 N/C. If the electron's mass is 9.11 × 10⁻² kg, what is its acceleration?['A)10.0 N/C', 'B)10.1 N/C', 'C)10.2 N/C', 'D)10.
  Hyperparameters: {'do_sample': True, 'max_new_tokens': 100, 'temperature': 1.0, 'top_k': 100, 'top_p': 0.85, 'num_return_sequences': 1, 'pad_token_id': 2}
Attempt 2 failed, retrying...

Text: You are an expert in Algebra. Lets think through these questions step by step to ensure we get the correct answer. There are only 5 options to choose from A,B,C,D or E. Ensure that your answer follows this format: Answer: . Two friends plan to walk along a 43-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 15% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?['A)21', 'B)21.5', 'C)22', 'D)22.5', 'E)23']
Answer:. Two friends plan to walk along a 43-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 15% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?['A)21', 'B)21.5', 'C)22', 'D)22.5', 'E)23']
An
  Hyperparameters: {'do_sample': True, 'max_new_tokens': 100, 'temperature': 0.5, 'top_k': 50, 'top_p': 0.95, 'num_return_sequences': 1, 'pad_token_id': 2}
Attempt 3 failed, retrying...

Text: You are an expert in Algebra. Lets think through these questions step by step to ensure we get the correct answer. There are only 5 options to choose from A,B,C,D or E. Ensure that your answer follows this format: Answer: . Two friends plan to walk along a 43-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 15% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?['A)21', 'B)21.5', 'C)22', 'D)22.5', 'E)23'] Answer:. A company produces 20,000 units of a product in January, 15,000 units in February, and 25,000 units in March. What is the total production for the three months?['A)60,000', 'B)61,500', 'C)62,000', 'D)62,500', 'E)63,0
  Hyperparameters: {'do_sample': True, 'max_new_tokens': 100, 'temperature': 0.7, 'top_k': 80, 'top_p': 0.9, 'num_return_sequences': 1, 'pad_token_id': 2}
Attempt 4 failed, retrying...

Text: You are an expert in Algebra. Lets think through these questions step by step to ensure we get the correct answer. There are only 5 options to choose from A,B,C,D or E. Ensure that your answer follows this format: Answer: . Two friends plan to walk along a 43-km trail, starting at opposite ends of the trail at the same time. If Friend P's rate is 15% faster than Friend Q's, how many kilometers will Friend P have walked when they pass each other?['A)21', 'B)21.5', 'C)22', 'D)22.5', 'E)23']
Answer: E)23
  Hyperparameters: {'do_sample': True, 'max_new_tokens': 100, 'temperature': 1.0, 'top_k': 100, 'top_p': 0.85, 'num_return_sequences': 1, 'pad_token_id': 2}

Question ID 0, Attempt 5: Correct Option - E

Question ID 0: Majority Option - C
```

- By the mean of LoRA fine-tuning, our model can achieve an **accuracy of 26%**.(On 100 questions)

Possible improvement:

- In our current model, we only use 3 hyperparameters for inference. We can try more hyperparameters to explore more diverse reasoning paths and improve the model's performance.
- We can enlarge `max_attempts` to get more stable results.

3.3 Multi-path

Here is an example of a question, its true answer and the predicted answer given by the model using multi-path.

```
Question: There are k-2 members in a certain band, including Jim and Ellen. Two members are to be selected to attend the Grammy awards ceremony. If there are 6 possible combinations in which Jim and Ellen are not selected, what is the value of k?
A)8
B)9
C)10
D)11
E)12
True Answer: A
Predicted Answer: A
```

Here are the intermediate inference steps of this quesiton using multi-path. We set up 3 groups of hyperparameters, which represent 3 reasoning pipelines respectively. The model does inference using these 3 groups of hyperparameters parallelly, and then gets 3 outputs. Then the select pipeline chooses the most reasonable outputs among them as the additional prompt in the next round. Finally the answer with the highest number of occurrences is selected as the final answer.

```
question 0 : correct answer is A
Hyparams0:The band has k-2 members.
Hyparams1:Total members in the band = k.
Hyparams2:Number of members in the band, including Jim and Ellen, is k.
Hyparams0:Jim and Ellen are in the band.
Hyparams1:Jim and Ellen are in the band.
Hyparams2:There are 6 possible combinations of 2 members in the band that do not include Jim and Ellen.
Hyparams0:There are 6 possible combinations in which Jim and Ellen are not selected.
You seem to be using the pipelines sequentially on GPU. In order to maximize efficiency please use a dataset
Hyparams1:Two members are to be selected for the Grammy awards.
Hyparams2:There are 6 possible combinations in which Jim and Ellen are not selected.
Hyparams0:Let's call the number of combinations where Jim and Ellen are not selected as x.
Hyparams1:The number of combinations in which Jim and Ellen are not selected is 6.
Hyparams2:So the total number of possible combinations is (k-2) \* 6 = k \* 6.
Hyparams0:The total number of combinations is k - 1.
```

```
Hyparams1:Since Jim and Ellen are in the band, the total number of combinations is k.
Hyparams2:Then, the number of combinations in which Jim and Ellen are selected is x + 1.
Hyparams0:The number of combinations where Jim and Ellen are selected is k - x.
Hyparams1:x + 1 = k - 1, so x = k - 2.
Hyparams2:Jim and Ellen are selected from the remaining combinations.
Hyparams0:The number of combinations is x.
Hyparams1:The value of k can be calculated as k = 6 + x = 6 + (k - 2) = 8.
Hyparams2:The total number of combinations where Jim and Ellen are not selected is x + 1 = (k - 1) - 2 = k - 3.
Hyparams0:The total number of combinations is k - 1 + x.
Hyparams1:The total number of combinations is k - 1.
Hyparams2:The total number of combinations is (k - 1) x.
Hyparams0:The number of combinations where Jim and Ellen are not selected is x.
Hyparams1:The number of combinations where Jim and Ellen are not selected is x.
Hyparams2:Since Jim and Ellen are in the band, the number of combinations is k - 1 + k - 2 = k.
Hyparams0:The total number of combinations is k - 1 + x = k - 1 + x = (k - 1) + x = k.
Hyparams1:The number of combinations where Jim and Ellen are selected is x + 1.
Hyparams2:The total number of combinations where Jim and Ellen are not selected is k - 1 + x.
Hyparams0:Answer: A.
Hyparams1:The number of combinations is x.
Hyparams2:Since Jim and Ellen are in the band, the total number of combinations is k.
Hyparams1:The total number of combinations is k.
Hyparams2:x + 1 = k - 1, so x = k - 2.
Hyparams1:The number of combinations where Jim and Ellen are not selected is x.
Hyparams2:The number of combinations where Jim and Ellen are not selected is x.
Hyparams1:The total number of combinations is k.
Hyparams2:The total number of combinations is k.
answer: A
Predicted Answer: A
```

- By the mean of multi-path, our model can achieve an **accuracy of 21%**(on **40** questions).

Possible improvement:

- Look at the output below,we can find that the inference was stopped before it reached all the answers. We can enlarge the number of steps to get more stable results.

```
correct answer is D
Hyparams0:k/l < 1, so k and l are both greater than 1.
Hyparams1:If k/l < 1, then k and l are both less than 1.
Hyparams2:If k/l < 1, then k/l is less than 1, and therefore k is greater than l.
Hyparams0:k/l < 1, so k^2/l^2 < 1.
Hyparams1:Both k and l are positive integers, so k^2 and l^2 are also positive integers.
Hyparams2:The expression k/l^2 is always greater than 1, because the square of any positive number is always greater than the number itself.
Hyparams0:k^2/l^2 < 1, so k^2/l < 1.
Hyparams1:k^2/l^2 < 1, so k^2/l < 1.
Hyparams2:C)k^2/l < 1.
Hyparams0:k^2/l < 1, so k^2 < l.
Hyparams1:k^2/l < 1, so k^2 < l.
Hyparams2:We know that both k and l are positive integers.
Hyparams0:k^2 < l, so k < l.
Hyparams1:Since k is positive and greater than 1, k^2 must be greater than 1.
Hyparams2:k^2 < l, so l > k^2.
Hyparams0:Answer: B.
Hyparams1:Let's take C. k^2/l > 1.
Hyparams2:k < l, so l is the greatest number.
Hyparams1:But k^2/l^2 = k/l < 1, which is not possible.
Hyparams2:C. Answer: C.
Hyparams1:So, C is not the correct answer.
Hyparams1:D. l/k > 1.
Hyparams1:k/l > 1, so k > 1.
Hyparams1:But k^2/l > 1, so k > l.
answer: C
Predicted Answer: C
correct answer is C
```

- We can also try more hyperparameters to explore more diverse reasoning paths and improve the model's performance.
- We can enlarge the number of paths to get more stable results.
- We can also use the fine tuning method beforehead to improve the model's performance.

References

[1] Li, Yifei, et al. "Making large language models better reasoners with step-aware verifier." arXiv preprint arXiv:2206.02336 (2022).

[2] Alpaca-CoT https://github.com/PhoebusSi/Alpaca-CoT