Chain-of-thought for complex reasoning using language models

Chandana Pamidi
cpamidi@umass.edu

Sreelekha Yaga
syaga@umass.edu

Sushrita Yerra
sushritayerr@umass.edu

Swetha Eppalapally
seppalapally@umass.edu

Yogeshwar Pullagurla
ypullagurla@umass.edu

1 Problem statement

Recent advancements have enhanced the reasoning capabilities of large language models (LLMs), yet these models continue to face challenges when addressing complex, multi-step reasoning tasks. Moreover, it is often the case that models performing exceptionally well on intricate reasoning tasks may have been inadvertently exposed to test data during pretraining, thereby obscuring true insights into their reasoning abilities (Zhang et al., 2024). This project is designed to assess the reasoning proficiency of models through a systematic experimental approach that incorporates rationales in various configurations. Utilizing the AQUA-RAT dataset (Ling et al., 2017), which comprises mathematical word problems necessitating reasoned solutions, we aim to explore the effect of rationales across diverse experimental paradigms, including zero-shot, few-shot, chain of thought, and fine-tuning. Our goal is to delineate the model's reasoning performance across varied in-context learning and fine-tuning scenarios.

2 Proposed vs Accomplished

In our project, we have successfully implemented and evaluated a series of experimental settings to investigate the reasoning capabilities of large language models using the AQUA-RAT (Ling et al., 2017) dataset. The experimental paradigms we successfully completed are as follows:

2.1 Accomplished

- **Zero-shot:** We evaluated the model's inherent reasoning abilities in a zero-shot context by supplying only the question and answer options as input. The model was instructed to select an answer option based solely on this information. Given that solving these mathematical questions necessitates multi-step reasoning calculations, this approach allowed us to measure the model's intrinsic reasoning capabilities.

- **Few-shot:** In this scenario, the model was given two examples, questions, and options. The aim was to evaluate the model's ability for in-context reasoning by observing how it learned from the examples to respond similarly to subsequent questions. The objective mirrored that of the zero-shot setting, where the model was expected to solely provide an answer option.

- **Few-shot with Rationale:** Expanding upon the few-shot approach, in this setup, the model was furnished with examples that included both reasoning and answer options as part of the output. The objective was for the model to generate both rationale and output similar to the provided examples, thereby comprehensively assessing its ability to reason.

- **Fine-tuning with Rationale with Arbitrary Weightage:** Here, the model was fine-tuned on the AQUA-RAT dataset, incorporating rationales into its training. It was trained to generate both the rationale and the answer option. This approach utilized a composite loss function, which concurrently assessed the rationale's quality and the accuracy of the answer.

- **Fine-tuning with Rationale and Weighted Loss:** Here, we have explored the Multi-Task Fine-Tuning with dual forward propogation. We have created two data points for each QA example during training: one for predicting the rationale underlying the answer and another for predicting the final answer option itself. Dual propagation is employed during the forward pass, entailing the separate computation of losses for rationale prediction and final answer option prediction. We have considered a fixed ratio of weightage [50, 50] between the answer and the rationale to investigate its influence on the learning process.

- **Zero-shot with Rationale as Guideline:** This setup extends the zero-shot scenario by incorporating rationales alongside the inputs. This evaluation framework aims to assess the model's capability in leveraging chain-of-thought reasoning effectively. The objective is to investigate whether providing rationales along with questions and options enables the model to comprehend, calculate, and select the correct answer option.

- **Zero-shot with Reasoning Structure generated from a Large Language model as Guideline:** We have experimented by providing the model with a reasoning structure on how to solve the problem instead of providing the rationale of the solution with the input in the earlier experiment using the methodology from the paper (Zhou et al., 2024).

2.2 Pending

- **Fine tuning with Weighted Loss - Different ratios of Rationale prediction loss and Answer option prediction loss:** Planned as an extension of the fine-tuning experiments, these are designed to explore whether training the model on rationales enhances its likelihood of selecting the correct answer option. By increasing the weightage of rationale loss, we aim to observe if there's a corresponding improvement in the model's performance in predicting the final answer. The underlying idea is that by explicitly teaching the model to reason effectively, it should demonstrate better reasoning capabilities when presented with new questions, leading to a higher probability of choosing the correct answer option. Despite having a well-thought approach, the experiement couldn't be executed due to multiple reasons.

  - **Technical Challenges with Implementation :** The proposed approach involved overwriting the training method of the SFTTrainer to accommodate two models, each focusing on a different loss objective. The core idea behind the proposed experiment was to conduct backpropagation solely on one model, leveraging a weighted loss objective and use the model's updated weights to synchronize the weights of the second model. Essentially, both models maintained identical weights throughout the training process. However, they were utilized differently during forward propagation: one model focused on predicting the answer option, while the other model emphasized rationale prediction (i.e inputs during training to each model are different). While this approach theoretically seemed viable, it required a departure from the standard training loop provided by the SFTTrainer. Implementing a custom training loop for this purpose was technically challenging, as it demanded significant time and effort to ensure proper synchronization and compatibility with other functionalities of the trainer.
  - **b) Scarcity of Computing Resources:** Running the proposed experiment demanded a substantial amount of computing resources. Given that previous fine-tuning approaches had already consumed a significant portion of available computing power, allocating resources for yet another intensive experiment became impractical. The proposed approach required nearly the same level of computational resources as previous experiments, which the team simply didn't have at their disposal.

3 Related work

Chain-Of-Thought(COT) Chain of Thought (CoT) is a method that breaks down complex reasoning into smaller, sequential steps, enhancing the performance of large language models. By presenting a series of intermediate reasoning steps, CoT enables these models to better understand and tackle intricate problems.While large language models (LLMs) have shown significant advancements in certain domains, there is still a need for improvement in tasks such as Complex Reasoning, Relation Extraction, and Data Summarization. The introduction of chain of thought prompting, as demonstrated in(Wei et al., 2023), allows reasoning abilities to naturally emerge in LLMs. Despite the capabilities of LLMs, they currently lack formal reasoning in their processes. Even when results appear to have underlying reasons, LLMs primarily map input text or instructions (prompts) to probable continuations or responses without explicit reasoning. This limitation poses challenges for higher-level summarization.

AQUA-RAT (Ling et al., 2017), short for Algebra Question Answering with Rationales, is a dataset comprising approximately 100,000 algebraic word problems, each accompanied by detailed rationales. The dataset systematically explains the solution process for each problem in natural language, offering a valuable resource for training models in Chain of Thought (CoT) capabilities specifically tailored for mathematical contexts. (Zhang et al., 2022) conducted experiments by applying CoT to the AQuA(Ling et al., 2017) dataset, demonstrating the potential to streamline manual efforts by utilizing Large Language Models (LLMs). By employing the prompt "Let's think step by step," (Zhang et al., 2022) showed that LLMs can generate reasoning chains systematically, not only step by step but also one by one.

Large Language Model Meta AI (LLaMA) There are multiple variations of LLaMA. Variants of LLaMA include, but are not limited to 1) LoRALLaMA, as demonstrated in experiments by (Lermen et al., 2023), exhibited effective text summarization through finetuning, indicating a broad range of potential applications. 2) LLaMAReviewer introduces a framework for automating code review processes using large language models (LLMs) and implementing parameter-efficient fine-tuning (PEFT) methods. This approach achieves high performance with less than 1% of trainable parameters. (Xu et al., 2023) experiments on LLaMAReviewer demonstrated that aligning the input representation with the pretraining format enhances the utilization of LLM capabilities. 3) FactLLaMA combines instruction-following language models with external evidence retrieval to improve fact-checking performance (Cheung and Lam, 2023).

Mathematical Reasoning and Answering The process of generating solutions for math word problems, including both the final answer and step-by-step explanations, involves the machine autonomously providing the solution. Numerous researchers, such as (Gaur and Saunshi, 2023), (Srivatsa and Kochmar, 2024), (Lin et al., 2024), (Yigit and Amasyali, 2024), (Bin et al., 2023) , (Yao et al., 2023) and others, have presented diverse hybrid models that demonstrate superior performance in solving mathematical problems when compared to the existing large language models.

Parameter-efficient Fine-tuning is a technique used in (NLP) to improve the performance of pre-trained language models on specific downstream tasks. LLMs need to be fine-tuned on task-specific expert annotated data to achieve optimal performance, which can be expensive and time consuming. (Kumar et al., 2024) fine-tuned PaLM-2 with parameter efficient fine-tuning (PEFT) using noisy labels obtained from gemini-pro 1.0. fine-tuned PaLM-2 model achieves performance levels close to those obtained with human-annotated labels, showcasing the effectiveness of using LLM-generated labels for domain-specific tasks, particularly in settings where expert annotations are sparse.

Self-Discover: Large Language Models Self-Compose Reasoning Structures A Framework to generate task-specific reasoning structure to solve complex reasoning problems that are difficult to solve with existing prompting methods. From the experiments in the paper (Zhou et al., 2024), it is proved that for challenging reasoning benchmarks like BigBench-Hard, grounded agent reasoning and MATH self-discover methods perform almost 32% better than Chain-of-Thought reasoning approach.

4 Dataset

To investigate the reasoning capabilities of large language models, we selected the AQUA-RAT (Algebra Question Answering with Rationales) dataset as detailed in (Ling et al., 2017). This dataset features a compilation of algebraic questions, each accompanied by a correct answer, a rationale for the solution, and a set of multiple-choice options—of which only one is correct. Figure 1 shows two sample problems from the dataset. It encompasses 97,975 entries, divided into 97,467 training, 254 validation, and 254 test records. This distribution heavily favors the training set, ensuring ample data to examine the effects of training on the model's reasoning abilities. The dataset provides a varied array of algebraic word problems and their corre-

Baselines			**Metrics**		
	$Accuracy$	$F1 - Score$	$Rouge1$	$Rouge - L$	$BertScore$
Zero-Shot w/oRationale	0.24015	0.1867	0.1927	0.2603	0.7868
Few-Shot with Rationale	0.2322	0.1622	0.1877	0.2468	0.7816
Few-Shot w/o Rationale	0.2086	0.2005	N/A	N/A	N/A
Zero-Shot with Rationale as Guidance	0.6496	0.5315	N/A	N/A	N/A

Table 1: Base line prediction results

Problem 1:
Question: Two trains running in opposite directions cross a man standing on the platform in 27 seconds and 17 seconds respectively and they cross each other in 23 seconds. The ratio of their speeds is:
Options: A) 3/7 B) 3/2 C) 3/88 D) 3/8 E) 2/2
Rationale: Let the speeds of the two trains be x m/sec and y m/sec respectively. Then, length of the first train = 27x meters, and length of the second train = 17 y meters. (27x + 17y) / (x + y) = 23 → 27x + 17y = 23x + 23y → 4x = 6y → x/y = 3/2.
Correct Option: B

Problem 2:
Question: From a pack of 52 cards, two cards are drawn together at random. What is the probability of both the cards being kings?
Options: A) 2/1223 B) 1/122 C) 1/221 D) 3/1253 E) 2/153
Rationale: Let s be the sample space.
Then n(s) = 52C2 = 1326
E = event of getting 2 kings out of 4
n(E) = 4C2 = 6
P(E) = 6/1326 = 1/221
Answer is C
Correct Option: C

Figure 1: Examples from AQUA-RAT dataset.

sponding rationales, enabling the model to learn different reasoning strategies for various types of problems. By aligning the model's generated reasoning steps with the given rationales, our goal is to evaluate the model's ability to replicate the cognitive processes involved in solving these problems.

4.1 Data preprocessing

The AQUA-RAT dataset exhibited outstanding quality, eliminating the need for additional modifications. Our preprocessing approach was customized to align with the specific requirements of our experiments, facilitating necessary modifications including the inclusion or exclusion of certain data elements. A significant aspect of our preprocessing involved the extraction of answers from the rationales, a common feature in the dataset, particularly for our zero-shot and few-shot experiments. This process required identifying and removing various string patterns that represented the answers within the rationales. This key preprocessing step was undertaken to bolster the model's capacity to independently generate explanations, thereby not depending on the answer embedded within the rationale. This method was intended to prompt the model to cultivate a more profound comprehension of the context and reasoning underpinning the provided data.

5 Baselines

5.1 Existing baselines:

Role-playing, enables LLMs to embody not only human characters but also non-human entities. This versatility allows LLMs to simulate complex humanlike interactions and behaviors within various contexts, as well as to emulate specific objects or systems.(Kong et al., 2024) designed a zero-shot role-play methodology and assessed its performance under the zero-shot setting across twelve diverse reasoning benchmarks. Here the prompts are designed and sampled manually which is time consuming and does not always guarantee optimal results.

5.2 Our baselines:

This section outlines the baseline configurations for this project using the CodeLlama Instruct 7B model. Our choice of the 7B variant was influenced by the balance between the available computational resources and the model's capacity for following instructions. The decision to use the 7B size of Llama was driven by our intent to investigate the effects of fine-tuning; opting for a larger model would necessitate more resources than are currently available. Specifically, we selected the instruct variant of CodeLlama due to its enhanced instruction-following capabilities. We also experimented with several other models, including the Llama 7B pretrained, instruction-tuned Llama 7B, and the CodeLlama pretrained. The analysis of these models was conducted manually using a set of predetermined prompts for a subset of the dataset.

To establish our baseline results studying the model's reasoning capabilities we have done the following set of experiments.

- **Zero-shot without rationale** This experiment evaluates the model's ability to navigate complex reasoning tasks, such as those found in the AQUA-RAT algebraic word problem dataset. Here, the model receives the question and associated options as inputs and is expected to produce an answer by reasoning through the presented problem. Table 2 llustrates a sample input provided to the model and the corresponding expected output. Notably, the sample output reveals that the model may have undergone pre-training on similar tasks, as it generates a rationale even though the prompt specifically requests only an answer.

- **Few-shot without rationale** This baseline too is similar to the above zero-shot rationale, without one notable difference being, that the input to the model consists of question, options and a set of similar question, option, and answer pairs and expecting the model to generate only the answer without the rationale as output for the current question. The aim of this baseline is to study the model's ability to effectively translate the reasoning capabilities from similar problems in the context and applying them to the current problem. Table 2 illustrates a sample input provided to the model and the corresponding expected output. When provided with two examples, the model generates answer options without rationale, successfully following instructions, unlike in zero-shot scenarios.

- **Few-shot with rationale** This baseline is similar to the above few-shot example with the notable difference being that the output also contains the rationale. This baseline is intended to study the model's abilities to reason in-context along with its pre-trained learning capabilities using the few-shot examples. Table 2 illustrates a sample input provided to the model and the corresponding expected output. When provided with two examples containing rationale in the prompt , the model successfully generates the rationale.

- **Zero-shot with rationale as guideline** This baseline is similar to the above zero-shot rationale, with one notable difference being, that the input to the model consists of question, options and the rationale and expecting the model to generate the output. The aim of this baseline is to study the model's ability to incorporate the provided rationale into it's reasoning capabilities to solve the problem. Table 2 illustrates a sample input provided to the model and the corresponding expected output.

6 Approach

This section describes a series of experiments designed to enhance our baseline model. These experiments focus on augmenting the model's reasoning skills by fine-tuning the pre-trained model using a combination of problem sets and associated rationales. We adjust the significance of rationales within the learning process by manipulating the weighting in the custom loss function. This adjustment allows us to explore the extent to which understanding rationales contributes to the model's reasoning abilities. Our methods integrate contextual reasoning, answer generation, and multi-task fine-tuning, aiming to improve the Large Language Model's (LLM's) ability to generalize solutions and address typical difficulties encountered in mathematical question-answering scenarios.

For each method, we assess the model's performance in both answer generation and rationale generation. We employ BertScore and RougeScore to gauge the model's effectiveness at generating rationales and use Accuracy and F1 score to evaluate its performance in generating answers.

Model	Prompt Description

Zero-Shot
with Rationale
	You are an mathematical assistant that helps users with the Algebra questions. You are asked a multi-choice question along with the answer options - A, B, C, D, E and rationale. Solve the provided question and choose the correct option based on the rationale provided. Follow the below instructions while responding.
	1) Do not provide intermediate calculation steps.
	2) Share only correct option and always provide response in the format : The correct answer option is ()
	Question: {question}
	Options: {options}
	Rationale: {rationale}

Zero-Shot w/o Rationale	You are an mathematical assistant that helps users with the Algebra questions. You are asked a multi-choice question along with the answer options - A, B, C, D, E. Solve the provided question and choose the correct option. Follow the below instructions while responding. 1) Do not provide intermediate calculation steps. 2) Share only correct option and always provide response in the format : The correct answer option is () Question: {question} Options: {options}
Few-Shot with Rationale	You are an mathematical assistant that helps users with the Algebra questions. You are asked a multi-choice question along with the answer options - A, B, C, D, E. Look at the below sample examples provided (question, answer options, expected output), focus on the expected output and its format (i.e reasoning, conclusion and final answer option selection). Example Question 1 : 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? Options for Example Question 1: [’A)21’, ’B)21.5’, ’C)22’, ’D)22.5’, ’E)23’] Expected Output for Example Question 1: If Q complete x kilometers, then P completes 1.15x kilometers. x + 1.15x = 43, 2.15x=43, x = 43/2.15 = 20. Then P will have have walked 1.15*20=23 km. The correct answer option is E) 23 Example Question 2 : Three birds are flying at a fast rate of 900 kilometers per hour. What is their speed in miles per minute? [1km = 0.6 miles] Options for Example Question 2: [’A)32400’, ’B)6000’, ’C)600’, ’D)60000’, ’E)10’] Expected Output for Example Question 2: To calculate the equivalent of miles in a kilometer. 0.6 kilometers = 1 mile, 900 kilometers = (0.6)*900 = 540 miles. In 1 hour there are 60 minutes. Speed in miles/minutes = 60 * 540 = 32400. The correct answer option is A) 32400 Now, Solve the below provided question and choose the correct option. Provide the reasoning and final answer option similar to the sample questions shared. Provide the output in a format that is same as the sample questions outputs. Provide the answer option response in the format : The correct answer option is () Question: {question} Options: {options}

Model	Prompt Description
Few-Shot	
w/o Rationale	You are an mathematical assistant that helps users with the Algebra questions. You are asked a multi-choice question along with the answer options - A, B, C, D, E. Look at the below sample examples provided (question, answer options, expected output), focus on the expected output and its format (i.e reasoning, conclusion and final answer option selection). Example Question 1 : 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? Options for Example Question 1: [’A)21’, ’B)21.5’, ’C)22’, ’D)22.5’, ’E)23’] Expected Output for Example Question 1: If Q complete x kilometers, then P completes 1.15x kilometers. x + 1.15x = 43, 2.15x=43, x = 43/2.15 = 20. Then P will have have walked 1.15*20=23 km. The correct answer option is E) 23 Example Question 2 : Three birds are flying at a fast rate of 900 kilometers per hour. What is their speed in miles per minute? [1km = 0.6 miles] Options for Example Question 2: [’A)32400’, ’B)6000’, ’C)600’, ’D)60000’, ’E)10’] Expected Output for Example Question 2: To calculate the equivalent of miles in a kilometer. 0.6 kilometers = 1 mile, 900 kilometers = (0.6)*900 = 540 miles. In 1 hour there are 60 minutes. Speed in miles/minutes = 60 * 540 = 32400. The correct answer option is A) 32400 Now, Solve the below provided question and choose the correct option. Provide the reasoning and final answer option similar to the sample questions shared. Provide the output in a format that is same as the sample questions outputs. Additionally, follow the below instructions 1) Do not provide intermediate calculation steps or explanation or reasoning for choosing a correct option. You’ll be penalized if you do so. 2) Provide the answer option response in the format : "The correct answer option is ()" 3) Do not follow the answer selection by any reasoning or explanation. You’ll be penalized if you do so. Question: ’question’ Options: {options}

Table 2: Prompts Used for Baselines

Fine-tuning methodologies:

6.1 Task specific fine-tuning

This process required preprocessing the dataset to enhance the answers with additional contextual information or supporting evidence, thereby refining the dataset. During the fine-tuning phase, the model was trained using both the questions and these enriched answers. The goal was to enhance the accuracy of the responses by incorporating pertinent contextual clues. To address issues during inference, where the model might generate both the rationale and the answer, we investigated methods to selectively extract only the answer. This included developing post-processing steps designed to ensure that the final output consisted exclusively of the relevant answer.

6.2 Custom weighted Loss fine-tuning

Although the initially idea was to implement custom weighted fine-tuning to vary the ratio of rationale loss and answer option prediction loss and run multiple experiments, we ran into the challenges outlined in the above “Pending“ section. Hence, as an alternative to run experiment for 50-50 ratio, we pre-processed the training data. For each input, we created two distinct inputs, each augmented with supplementary information to distinguish between tasks. One input included the question, options, and the statement “The rationale is,“ followed by the rationale. The other input comprised the question, options, and the statement “The correct answer option is,“ followed by the answer option. We utilized dual propagation to compute the loss for both the rationale and answer options, amalgamating them in a 50-50 ratio. This methodology allowed us to assess the efficacy of each component—rationale generation and answer option—in enhancing the model’s reasoning capabilities.

6.3 Self-Discovery of Reasoning Structure

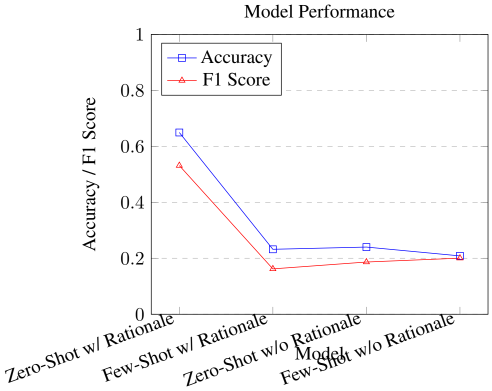
This method involves generating the steps to solve the task before solving it. Two phases are involved in this method to solve the problem, Discover the Reasoning Structure on Task-Level and Solve Problems Using the Discovered Structure on Instance-Level. For the first phase, we have used a larger model LLama 3 70B. The primary reason to choose this model is that we need a model that knows how to solve the problem to give a better reasoning structure.

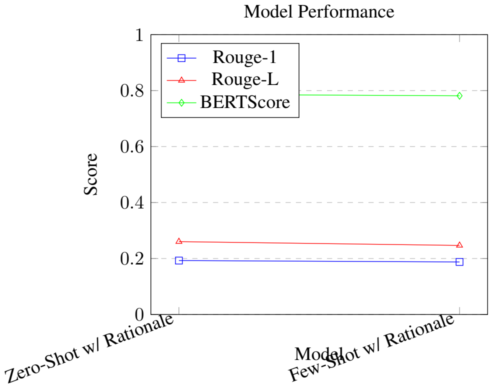
There are three steps involved in generating the reasoning structure for the problem. In the first step, from a list of reasoning modules that can be applied to various types of reasoning problems the modules which can be used for the task are chosen. The selected reasoning modules are then adapted to the task, in this step the reasoning modules are rephrased. And in the final step, the reasoning modules are converted into a structured actionable plan using which the problem can be solved.

SELECT
ADAPT
APPLY

7 Results and Reasoning

7.1 Base line Results





We chose these baselines over other models for several reasons:

- **Model Capability**: The LLaMA 2 7B model, with its extensive pre-training on 2 trillion tokens, exhibits a strong understanding of natural language and mathematical concepts, making it suitable for our task.
- **Resource Efficiency**: Leveraging pre-trained models like LLaMA 2 7B allows us to avoid the computational cost of training models from scratch, making our approach more resource-efficient.

- **Versatility**: By exploring various configurations (zero-shot vs. few-shot, with vs. without rationale), we aim to understand the impact of different factors on model performance and generalize our findings to a wider range of scenarios.

7.2 Task Specific Finetuning results

We performed finetuning on the base model for next-word prediction with the AQuA-RAT dataset of 97,467 training examples for 10 epochs. And after finetuning we observe the model giving better rationales and answers in comparison to the base model.

The training for 10 epochs took 33 hours on a single Nvidia A100 GPU with a batch size of 10 and grad accumulation of 4.

In comparison to the base model, finetuning brought an improvement from 24.01% to 26.38% in accuracy and from 18.67% to 28.58% in F1-Score. There is major improvement seen in Rouge scores with Rouge1 increasing from 0.1927 to 0.7296 and Rouge-L increasing from 0.2603 to 0.6650.

Because of the fine-tuning, the model is more confined to the format used in the AQuA-RAT dataset, and hence the rationale generated by the model is more aligned with the rationales in the dataset. This is the reason for the increase in the Rouge scores.

Even when the answers generated by the model are wrong, the model generates the rationale in a way that the Rouge scores are high. This is a limitation of the Rouge score as it doesn't consider the correctness of the numbers in the rationale.

The results of the finetuning experiments are shown in Table 3.

7.3 Custom weighted fine-tuning results

Performed fine-tuning with AQuA-RAT dataset of 194,934 training examples for 5 epochs, which took 30 hours of runtime with an A100 GPU. The dataset was pre-processed so that only 50% of the training samples included rationale, while the other 50% included only the correct answer. This pre-processing was done to implement our idea of a 50-50 weighted loss with rationale. We modified the dataset so that the model could only see 50% of the rationale examples.

The goal of this approach was to investigate how the model balances learning from rationale versus answers and to explore the impact on the model's reasoning capabilities. By introducing this 50-50 split, we aimed to discern whether the model can effectively internalize reasoning processes from limited rationale exposure and to compare its performance against a standard training approach.

The fine-tuning with the AQuA-RAT dataset resulted in an accuracy of 0.2992 and an F1 score of 0.2971. The model's performance was further evaluated using Rouge scores, achieving 0.8323 for Rouge-1, 0.8257 for Rouge-2, and 0.8317 for Rouge-L. Additionally, the BLEU score was 0.6486, and the BERTscore was 0.7167. The performance of this fine-tuning is comparatively better than the task-specific fine-tuning. This supports the idea that focusing on improving the rationale of a model can indeed have a direct correlation with its ability to predict answer accurately.

7.4 Self-Discovery Reasoning structure Results

The reasoning structures generated using the self-discovery method looked promising. They had broken down the problem into smaller parts and identified the parts that had to be solved. Refer to Figure 3 for an example of the generated reasoning structures. To evaluate the model's performance when using the generated reasoning structures, We provided the model with rationale from the AQuA-RAT dataset and asked it to choose the correct answer from the options. The model was able to generate the correct answer for 64.96% of the questions. The F1-Score for the model was 53.15%. The results of these experiments are shown in Figure 2. However, the model was not able to achieve the same level of performance when using the generated reasoning structures. It was not able to reach the performance of even the baseline zero-shot without rationale results.

8 Error analysis

8.1 Self-Discovery Reasoning structure Error Analysis

The self-discovery method failed drastically even in comparison to the zero-shot without rationale baseline. The model was not able to generate the correct answer for most of the questions with the accuracy being as low as 5.12% and the F1-Score of 6.45% refer Figure 2

The model failed because of the following reasons:

Experiments	**Metrics**				
	$Accuracy$	$F1-Score$	$Rouge1$	$Rouge-L$	$BertScore$
Zero-Shot Base Model	0.24015	0.1867	0.1927	0.2603	0.7868
Task Specific Finetuning	0.2638	0.2858	0.7296	0.6650	0.7085
Custom-Weighted Finetuning	0.2992	0.2971	0.8322	0.8317	0.7167

Table 3: Experiments and results

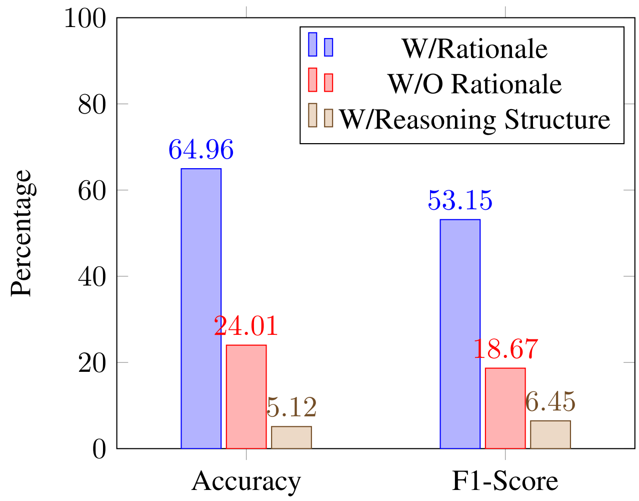


Figure 2: Self-discovery results.

- **Lack of reasoning modules**: The reasonable modules used in the self-discovery method were not able to solve the problem. The modules were abstract and did not help when handling algebraic problems. If the model had more specific reasoning modules for algebraic problems, it would have performed better.

  Examples:

  - "How could I devise an experiment to help solve that problem?"
  - "Make a list of ideas for solving this problem, and apply them one by one to the problem to see if any progress can be made."
  - "How could I measure progress on this problem?"
  - ...

- **Lack of solving numerical operations**: The model was not able to solve the numerical operations involved in the problem.

  Example : Model's response to the problem in table 3:

  " Step 1: Calculate the discounted price after the 22% discount The discounted price after the 22% discount is: \$100 x 0.78 = \$78 Step 2: Calculate the price after applying the \$20-off coupon ...
  \$58 + \$1.90 = \$78
  Solving for the original price, we get:
  \$78 - \$1.90 = \$58
  \$58 = \$78 - \$1.90
  \$58 = \$78 - \$1.90
  \$58 = \$61
  The correct answer option is (A) \$61. "
  Whereas the correct answer is \$78.20

8.2 Finetuning Error Analysis

- **Difficulty in generating the output in a specific pattern**: Despite training the model with pre-processed training data with specified formatting using (95k) records in task-specific finetuning and (190k) records in custom weighted loss finetuning, the model failed to generate the output in similar fashion during inference. Example during training is of the format - Question, Options, "The correct answer option is ", answer option. During inference, when the Question and Options were provided as input, the expectation was that the model would start with "The correct answer option is ", however the output is quite random. This could signify two things a) Either the training data was not enough to surpass the knowledge gained during pre-training in the next word prediction b) Or it required ex-

Question:

The original price of an item is discounted 22%. A customer buys the item at this discounted price using a $20-off coupon. There is no tax on the item, and this was the only item the customer bought. If the customer paid $1.90 more than half the original price of the item, what was the original price of the item?

Selecting reasoning modules:

- How can I simplify the problem so that it is easier to solve?
- How can I break down this problem into smaller, more manageable parts?
- Critical Thinking: This style involves analyzing the problem from different ...
- ...

Adapted reasoning modules:

- Simplification: Can I simplify the problem by identifying the key elements: original price, discount percentage, coupon value, and the final price paid by the customer?
- Breakdown: Can I break down the problem into smaller parts:
  - Calculate the discounted price after the 22% discount
  - Calculate the price after applying the $20-off coupon
  - Set up an equation to relate the final price paid to the original price
- Critical Thinking: Are there any assumptions I need to question? Is the problem correctly stated? ...
- ...

Implemented reasoning modules:

- Simplification: Let's identify the key elements:
  - Original price (unknown)
  - Discount percentage: 22%
  - Coupon value: $20
  - Final price paid by the customer: $1.90 more than half the original price
- Breakdown: Let's break down the problem into smaller parts:
  - Calculate the discounted price after the 22% discount
  - Calculate the price after applying the $20-off coupon
  - Set up an equation to relate the final price paid to the original price
- Critical Thinking: Are there any assumptions I need to question? Is the problem correctly stated? Are there any ambiguities in the problem description?
  - The problem seems clear, but I'll keep an eye out for any potential issues. ...
- ...

Figure 3: Shortened example from Self-discovery generated reasoning structures.

perimentation with multiple hyper parameters which required more computing than we could resource.

- **Difficulty in choosing the correct answer option**: Another interesting observation is that even though the rationale generated was correct and the answer derived in the rationale is correct, the model couldn't map it to the correct answer option. This could be primarly because the answers associated with the options have overlapping meanings (ex - range, upper or lower bound etc) or unclear language or potentially other option values very close to the correct option.

9 Contributions of group members

List what each member of the group contributed to this project here. For example:

- Chandana: Data-preprocessing and Finetuning for Custom Weighted Loss Approach, Report writing.
- Sreelekha: Experiments with Zero-shot, Few-shot prompts, Report Writing
- Sushrita: Data-preprocessing and Task specific Finetuning, Report Writing, Error Analysis
- Swetha: Experiments with Zero-shot, Few-shot prompts, Data-preprocessing and Finetuning for Custom weighted loss approach, Report Writing, Error Analysis
- Yogeshwar: Running Task-Specific Finetuning, Self-discovery approach, Report Writing, Error Analysis

10 Conclusion

In this project, we explored the effectiveness of fine-tuning a large language model on a mathematical question-answering dataset. We implemented two different fine-tuning methodologies, including task-specific fine-tuning and custom weighted loss fine-tuning, and experimented with a prompting framework to self-discover reasoning structure. Our results showed that task-specific fine-tuning and custom weighted loss fine-tuning significantly improved the model's performance, while the self-discovery method failed to achieve the desired results.

The performance of the model is still not at the level where it can be used in a real-world scenario. The model is not able to generate the correct answer for most of the questions and failed mainly in solving numerical operations while generating the answer.

Few observations which are notable from the experiments are:

- Despite providing the model with a structured actionable plan, it was not able to solve the problem.
- The base model does not always choose the correct answer, even when the rationale is provided in the input. It was only doing so with 64.96% accuracy.
- Finetuning the model with a custom weighted loss function significantly improves the model's performance over simple task-specific finetuning.
- A numeric reasoning module is required for the model to solve the problems effectively.
- Evaluation metrics like BertScore and RougeScore are not the best metrics for evaluating the model's performance in generating rationales.

In the future, we would like to explore the following directions:

- Experiment with different hyperparameters and training strategies to improve the model's performance.
- Explore the custom weighted loss finetuning for different ratios of Rationale loss vs Answer prediction loss
- Implement a numeric reasoning module to help the model solve the numerical operations involved in the problems.
- Experiment with different evaluation metrics to better evaluate the model's performance.
- Explore other large language models and compare their performance with the LLaMA 2 7B model.

11 AI Disclosure

- Did you use any AI assistance to complete this proposal? If so, please also specify what AI you used.
  - Yes, ChatGPT

If you answered yes to the above question, please complete the following as well:

- If you used a large language model to assist you, please paste *all* of the prompts that you used below. Add a separate bullet for each prompt, and specify which part of the proposal is associated with which prompt.
  - Paraphrase the following with better wording
  - How to add latex tables which has multiple columns
  - How to add graphs in latex
- **Free response:** For each section or paragraph for which you used assistance, describe your overall experience with the AI. How helpful was it? Did it just directly give you a good output, or did you have to edit it? Was its output ever obviously wrong or irrelevant? Did you use it to generate new text, check your own ideas, or rewrite text?
  - For rewriting my introduction for any language issues, the response was pretty good. It cited some things that are already known and some unknown. It also gave a response editing my writing with it's own suggestions which was almost really good. I only had to make a few tweaks based on my style preference.
  - Related work: I have used AI assistance for paraphrasing and to get detailed information/explanation about a topic which I felt difficult to understand while reading publications.
  - ChatGPT is used in Dataset,Baselines and data preprocessing for better formatting and wording,It did the task as expected.
  - For generating the tables and graphs it has given basic ones and If I again asked for better looking tables/graphs it has given ideas but most of them have errors

- For the Approach section, when asked ChatGPT to improve the wording, has used vocabulary that was not cohesive with the rest of the document. I had to reword it myself going through every word.

References

Bin, Y., Shi, W., Ding, Y., Yang, Y., and Ng, S.-K. (2023). Solving math word problems with reexamination.

Cheung, T.-H. and Lam, K.-M. (2023). Factllama: Optimizing instruction-following language models with external knowledge for automated fact-checking.

Gaur, V. and Saunshi, N. (2023). Reasoning in large language models through symbolic math word problems.

Kong, A., Zhao, S., Chen, H., Li, Q., Qin, Y., Sun, R., Zhou, X., Wang, E., and Dong, X. (2024). Better zero-shot reasoning with role-play prompting.

Kumar, B., Amar, J., Yang, E., Li, N., and Jia, Y. (2024). Selective fine-tuning on llm-labeled data may reduce reliance on human annotation: A case study using schedule-of-event table detection.

Lermen, S., Rogers-Smith, C., and Ladish, J. (2023). Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b.

Lin, Q., Xu, B., Huang, Z., and Cai, R. (2024). From large to tiny: Distilling and refining mathematical expertise for math word problems with weakly supervision.

Ling, W., Yogatama, D., Dyer, C., and Blunsom, P. (2017). Program induction by rationale generation : Learning to solve and explain algebraic word problems.

Srivatsa, K. A. and Kochmar, E. (2024). What makes math word problems challenging for llms?

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E., Le, Q., and Zhou, D. (2023). Chain-of-thought prompting elicits reasoning in large language models.

Xu, L., Xie, H., Qin, S.-Z. J., Tao, X., and Wang, F. L. (2023). Parameter-efficient fine-tuning methods for pretrained language models: A critical review and assessment.

Yao, J., Zhou, Z., and Wang, Q. (2023). Solving math word problem with problem type classification.

Yigit, G. and Amasyali, M. F. (2024). Data augmentation with in-context learning and comparative evaluation in math word problem solving. *SN Computer Science*, 5(5).

Zhang, H., Da, J., Lee, D., Robinson, V., Wu, C., Song, W., Zhao, T., Raja, P., Slack, D., Lyu, Q., Hendryx, S., Kaplan, R., Lunati, M., and Yue, S. (2024). A careful examination of large language model performance on grade school arithmetic.

Zhang, Z., Zhang, A., Li, M., and Smola, A. (2022). Automatic chain of thought prompting in large language models.

Zhou, P., Pujara, J., Ren, X., Chen, X., Cheng, H.-T., Le, Q. V., Chi, E. H., Zhou, D., Mishra, S., and Zheng, H. S. (2024). Self-discover: Large language models self-compose reasoning structures.