Evaluating Mathematical Reasoning Chains

Tanay Dixit (tanayd2)* **Neeraja Kirtane (kirtane3)** **Rishabh Adiga (radiga2)**
Aakriti (aa117) **Akshat Sharma (akshat7)** **Omkar Gurjar (ogurjar2)**

Abstract

There has been a growing interest in teaching large language models to perform step-by-step reasoning or Chain of thought (CoT) reasoning to solve complex tasks (Chung et al., 2022; Wei et al., 2022; Wang et al., 2022), yet assessing the quality of these step-by-step reasoning chains remains a challenge. Current approaches tend to focus on the correctness of the final answer, potentially overlooking the intricate facets of genuine reasoning. Existing methods for CoT evaluation either perform poorly on math-based tasks (Prasad et al., 2023) or fail to measure the logical correctness of steps effectively since they only check math calculations (Chern et al., 2023). There is also a lack of a benchmark dataset with human annotations of CoT chains generated by open-source Large Language Models (LLMs) since existing methods focus only on GPT-3 chains (Golovneva et al., 2022a). As a result, no thorough evaluation metric has been trained yet for evaluating mathematical reasoning chains.

To bridge this gap, we plan to propose MARoC ((**Ma**thematical **R**easoning **C**hain Evaluator)), particularly focused on evaluating math-based reasoning chains. Inspired by current trends in training evaluation metrics (Qin et al., 2022; Zhong et al., 2022), we plan to make use of large-scale math datasets (Lightman et al., 2023; Yue et al., 2023a) with augmentation strategies (Khalifa et al., 2023) to fine-tune (Qin et al., 2022; Prasad et al., 2023) a pre-trained metric. We plan to do an extensive evaluation on both Golovneva et al. (2022a) benchmark and our own generated data to show the effectiveness of our metric. *

1 Introduction

The Chain of Thought (CoT) concept involves the systematic progression of ideas, establishing a logical sequence for problem-solving and concept exploration. CoT facilitates the structured breakdown of complex problems, revealing insights into underlying sub-problems. It enhances the explainability of Language Models (LLMs), thereby improving their performance in various tasks by encouraging transparent reasoning pathways (Wei et al., 2022). However, evaluating the accuracy of these reasoning steps is challenging due to the lack of reliable automatic evaluation methods or gold reference chains (Prasad et al., 2023). Previous works have mainly concentrated on evaluating the CoT based on the model's downstream performance for the given task. Focusing only on the conclusion of the reasoning chain may mix up the quality of reasoning with unrelated shortcuts or spurious steps used to arrive at the answer.

In mathematical settings, correct answers heavily rely on intermediate steps, yet their evaluation is often overlooked. Existing metrics, such as generic entailment models, tend to perform poorly on numeric-based inputs (Prasad et al., 2023). (Chern et al., 2023) propose a general evaluation framework, but it concentrates solely on assessing the correctness of mathematical operations, limiting its applicability as LLMs may make errors beyond numerical calculations (Golovneva et al., 2022a). The absence of a benchmark dataset featuring Chain of Thought (CoT) chains generated by diverse open-source Language Models (LLMs) is noteworthy. Existing pre-trained metrics predominantly focus on reasoning chains in the (Golovneva et al., 2022a) benchmark dataset, limited to the GPT-3 model Davinci. Given the varied distributions and error patterns among different LLMs, relying solely on chains from a specific model may yield biased conclusions (Peyrard, 2019a). Training metrics exclusively on GPT-3 chains introduces variance hindering generalization. This underscores the necessity for a more inclusive benchmark dataset to capture the diverse CoT characteristics across

* Team Leader
*Code available at https://github.com/neeraja1504/Maroc

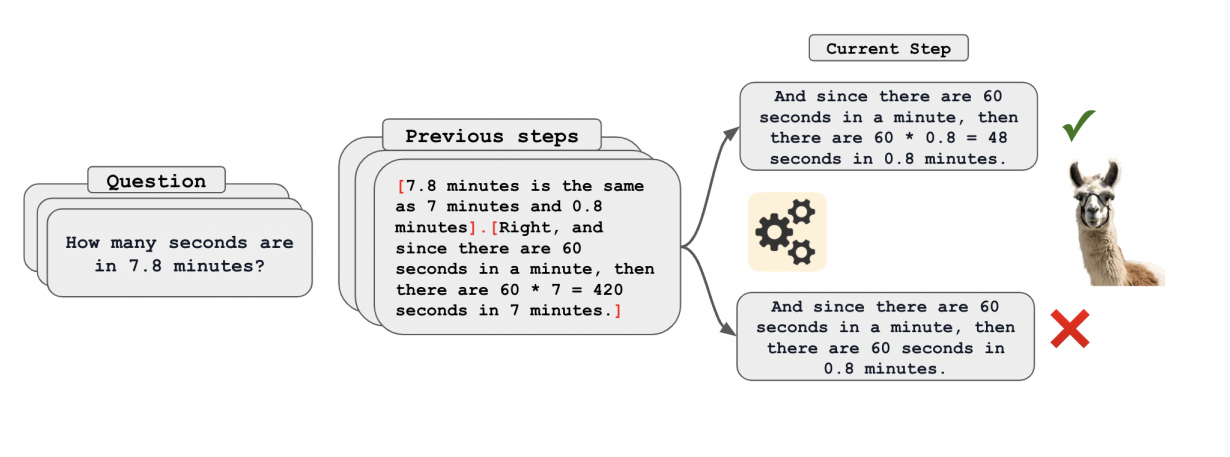

Figure 1: Overall MAROC metric fine-tuning process. For every question in the training data, we have a set of steps that are used to arrive at the final answer. For every step, the dataset contains a positive and corresponding negative step, which can be either synthetically generated or model generated. During training, the metric is trained to penalize the negative step and reward the positive step, thus learning the nuanced errors in reasoning chains.

different LLMs and a more robust pre-trained metric. We plan to train using recently released large-scale annotated datasets (Yue et al., 2023a; Lightman et al., 2023). We will leverage training techniques derived from (Khalifa et al., 2023) for fine-tuning LMs on mathematical reasoning and Direct Preference Optimization (DPO) for fine-tuning LMs to align with human preferences (Rafailov et al., 2023). We hope that by using a high-quality and large dataset as compared to Prasad et al. (2023) and a superior training technique, we can outperform the existing metrics at evaluating mathematical reasoning chains.

2 Related Work

2.1 Chain of Thought Prompting

Inspired by the human thought processing of breaking down complex reasoning problems into multiple intermediate steps, (Wei et al., 2022) shows that prompting sufficiently large language models with a few demonstrations in natural language can unlock the reasoning capabilities of LLMs. They show improvements in problem-solving performance on various math problem benchmarks (Cobbe et al., 2021; Hendrycks et al., 2021; Lightman et al., 2023; Yue et al., 2023b). Other works such as (Kojima et al., 2022) propose a zero-shot approach of prompting LLMs to solve complex reasoning tasks such as Arithmetic, Common Sense, and Symbolic reasoning. Combining the methodologies of the above two works (Zhang et al., 2022) eliminate the need for manually crafted exemplar CoT demonstrations by generating automatic demonstrations leveraging (Kojima et al., 2022)'s single prompting technique.

2.2 Evaluation of CoT

With more and more works exploring LLMs CoT capabilities, recent research has also started focusing on evaluating the quality of the generated CoT sequences. (Lightman et al., 2023) provide the first large-scale dataset for step-wise CoT evaluation (PRM800K), in which they rate each step in the CoT as positive/negative or neutral. (Golovneva et al., 2022a) proposes a suite of metrics (ROSCOE) for evaluating the reasoning steps on several dimensions including semantic alignment (factuality, hallucination, etc.), semantic similarity, logical (self and source) consistency, and language coherence. Moving towards reference-free metrics, (Prasad et al., 2023) propose an evaluation technique RECEVAL based on the correctness and informativeness of the generated CoT.

2.3 Meta Evaluation

Automatic CoT evaluation metrics, (Golovneva et al., 2022b; Prasad et al., 2023; Tang et al., 2022), are compared based on their correlations with human annotations. However, (Peyrard, 2019a) show that metrics that correlate well with human annotations can possibly correlate poorly amongst each other in the desired model output regime.

3 Methodology

In this section, we highlight the key aspects we adopt for training an evaluation metric and accu-

Error Type	Definition
Grammar	Faulty, unconventional, or controversial grammar usage
Factuality	Information about an object (i.e. quantity, characteristics) or a named entity doesn't match with the input.
Hallucination	Information is not provided in the problem statement and is irrelevant or wrong
Redundancy	Explanation contains redundant information, even though might be factual, is not required to answer the question
Repetition	Step paraphrases information already mentioned in previous reasoning steps
Missing step	The content of the reasoning is incomplete and lacks the required information to produce the correct answer.
Coherency	Steps contradict each other or do not follow a cohesive story
Commonsense	Model lacks relations that should be known from general world (e.g., "all ducks are birds")
Arithmetic	Error in math calculations

Table 1: Taxonomy of Step-by-Step Reasoning Errors.

Model	questions with at least one error	Total questions
ChatGPT	11 (38%)	29
Llama 7B	20 (87%)	23
FlanT5 (large)	22 (88%)	25
Vicuna	23 (85%)	27

Table 2: Number of questions with at least one error in the generated steps for each LLM.

rately measuring its effectiveness. To begin with, we identify certain shortcomings with the existing meta-evaluation datasets, i.e., ROSCOE. We address it by expanding the dataset to represent several new LLMs that have been introduced (§3.1). We then train our own evaluation metric by making use of large-scale datasets (Lightman et al., 2023) (§3.2).

3.1 Benchmark Expansion

ROSCOE is the only existing dataset for meta-evaluating reasoning chains. It consists of annotated chains generated **only** by GPT-3 (Brown et al., 2020) (*175b_verification*) model. This lack of outputs from various LLMs is concerning as, Peyrard (2019b) shows that evaluation metrics that behave similarly on a skewed dataset can strongly disagree when used to evaluate different types of errors. Hence putting into question the true performance of those metrics in evaluating the reasoning chains of open-sourced LLMs, which are known to depict different characteristics. This same issue has been observed in meta-evaluation of summarization systems (Fabbri et al., 2021), hence learning from these previous mistakes, the first step we adopt is to expand the given ROSCOE dataset to represent outputs from various LLMs.

Taxonomy. We adopt a similar annotation setup as Golovneva et al. (2022a). The list of error categories and the definitions are present in Table 1. More details are in §4.

3.2 MaRoC: Mathematical Reasoning Chain Evaluator

Problem Formulation Our goal is to score a step-by-step reasoning chain generated by a language model. We assume that the model is given a *question* q and is prompted to generate step-by-step reasoning (Nye et al., 2021). We refer to this as a *hypothesis* $h = \{h_1, \cdots, h_N\}$ of N-steps, including a final answer as the last step. We do not assume the availability of gold step-by-step reasoning *references* $r = \{r_1, \cdots, r_K\}$ of K-steps as these are not available for majority datasets (Prasad et al., 2023). We train a metric M that learns to give a score s_i to h_i given q and $h_{<i}$. Formally

$$s_i = M(h_i|q, h_{<i}) \tag{1}$$

Although we score each step, we can combine the step-level scores in order to determine the overall quality of a reasoning chain. Following the scoring setup in (Golovneva et al., 2022a), we consider a reasoning chain to be only as good as its least correct step. Therefore, given a reasoning chain and an evaluation metric, we aggregate step-level scores using a `min` operation.

3.3 Training

For training the metric, we plan to make datasets that contain positive and negative reasoning chains for every given question. This paired data can be either generated with the help of human annotations (Lightman et al., 2023) or synthetically generated (Golovneva et al., 2022a). Given this paired data, we can explore training techniques like Supervised Fine-tuning and Direct Preference Optimization, which would help train the metric to differentiate between right and wrong reasoning steps.

Metric	QUAL	COH	COM	FACT	HALL	RED	REP	LOGIC	MATH
ROSCOE-SA	0.20	0.19	0.19	0.08	0.22	0.39	0.79	0.18	**0.44**
-SS	0.20	0.17	0.17	0.14	0.25	**0.51**	0.87	0.15	0.23
-LI	0.28	0.26	0.18	0.34	0.22	0.35	**0.98**	0.22	0.09
ReCEval	0.36	0.31	0.21	0.37	0.28	0.40	0.63	0.25	0.24
ChatGPT	-	0.21	0.17	0.32	0.44	0.41	0.77	-	0.27
GPT-4	**0.57**	**0.47**	**0.51**	**0.51**	**0.61**	0.33	0.82	**0.54**	0.27

Table 3: Results reproduced for the ROSCOE and RECEVAL metrics and ChatGPT and GPT-4 scores for the ROSCOE dataset. Missing ("-") values imply statistically insignificant results.

4 Experimental Setup

Meta Evaluation dataset Several LLMs have been introduced since the ROSCOE benchmark was created. In addition to GPT3 we use chains produced by Flan-T5 (Chung et al., 2022), LLama-2 (Touvron et al., 2023), Vicuna (Zheng et al., 2023) and ChatGPT. We use the same set of 200 GSM8k questions as used in ROSCOE. We design a few shot prompts (Table §8) to prompt LLMs to generate the reasoning step before arriving at the solution. We use the same prompt for all the LLMs in order to ensure consistency. We evaluate each step following the Error Taxonomy in Table 1. Each step is assigned a correct or incorrect label. This setup is more simple for arriving at a good inter-annotator agreement score. The full details of the annotation process are mentioned in Figure 3.

Meta-Evaluation: Baselines We compare our approach with existing baselines ReCEval and ROSCOE. For ReCEval we use the correctness score. For ROSCOE, we compare against semantic similarity (SS), semantic alignment (SA), and logical inference (LI) metrics from ROSCOE. For ROSCOE-SA and -SS, we use the text-similarity models finetuned on reasoning chains [2]. In addition to this, we add competitive baselines that use ChatGPT and GPT-4 to evaluate the correctness of reasoning steps. We prompt these models to evaluate the correctness of every step given the question and previous steps. Prompt provided in Table §8.

Meta-Evaluation: Correlation Measure After scoring each reasoning chain with either metric, we assess whether the score can indicate the presence or absence of each type of error. Following Golovneva et al. (2022a), we use the Somer's-D correlation (Somers, 1962), which measures the ordinal association between two dependent quantities. In our case, we evaluate a metric M against the random variable indicating whether the chain is erroneous ($E \in \{0, 1\}$). Using Kendall's τ coefficient, Somer's-D correlation is defined as:

$$D_{SE} = \tau(E, S)/\tau(E, E).$$

MARoC Setup For our training data, we use the PRM800k dataset Lightman et al. (2023). Every question in this dataset consists of a gold reference solution and annotated model-generated reasoning chains. Human annotators mark each step in the reasoning chain as positive, negative, or neutral. We aim to utilize these step-level annotations to create a reward model that takes as context the question and the previous reasoning steps to predict the correctness of each step. Lightman et al. (2023) actively mine reasoning chains that are correct under the current reward model but produce incorrect answers to effectively use human effort. This ensures a sufficient number of negative steps for reward modeling. Additionally, we also plan to explore synthetic perturbation techniques that would help us increase the size of the dataset to cover all questions. We adopt text perturbation techniques as those mentioned in Golovneva et al. (2022a); Yu et al. (2023).

5 Results

Baseline results : We reproduce the results for the ROSCOE and RECEVAL metrics. We also obtain the scores on ChatGPT and GPT-4 as shown in Table 3. We see that GPT-4 performs better for the following error types: QUAL, COH, COM, FACT, HALL, LOGIC. For RED, ROSCOE-SS has the best performance. ROSCOE-SA has the highest correlation for MATH.

[2]We use facebook/roscoe-512-roberta-base models for computing chain embeddings

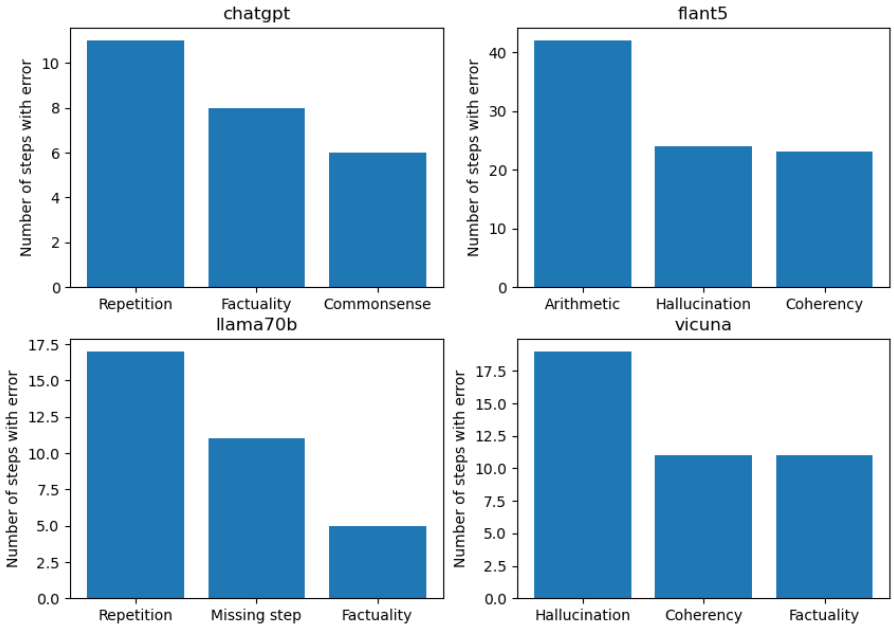

Figure 2: The top 3 error types for each LLM according to our taxonomy

Annotated Dataset Analysis We analyze the performance of the 4 LLMs based on our human-annotations for a sample of 120 questions [3]. Table 2 shows the total number of questions with at least one error. Note that while we sample equal number of questions (30) for each LLM, we remove the questions for which invalid chains are longer than the maximum number of tokens in the output of the LLM. We observe that ChatGPT outperforms the rest, but still makes an error **38%** of the time. Next, we present a fine-grained analysis of different errors made by the four LLMs. Figure 2 shows the top 3 error types for each LLM. We observe that 3/4 LLMs make factuality related errors. Vicuna and FlanT5 are prone to making coherency and hallucination related errors alluding to their inability to understand mathematical reasoning. Further, FlanT5 most commonly makes arithmetic errors, which highlights its specific limitation in evaluating mathematical equations. We also look at the least commonly made errors by each LLM and find that LLMs rarely make redundancy-related and grammatical errors. This can be linked to their superior natural language understanding.

6 Discussions

The absence of a diverse benchmark dataset for Chain of Thought (CoT) chains generated by open-source Language Models (LLMs) is a notable limitation, as the varied distributions and error patterns among different LLMs may lead to biased conclusions when relying solely on chains from a specific model (Peyrard, 2019a). Our observation that ChatGPT outperforms others, despite still having a 38% error rate, underscores the necessity of considering a range of LLMs. The fine-grained analysis of error types reveals intriguing insights, with three out of four LLMs demonstrating factuality-related errors. Notably, Vicuna and FlanT5 exhibit challenges in coherency and hallucination, indicating difficulties in comprehending mathematical reasoning. Furthermore, FlanT5's tendency toward arithmetic errors emphasizes specific limitations in evaluating mathematical equations. This nuanced analysis underscores the importance of a diverse set of LLMs in evaluating and understanding CoT performance.

Models exclusively fine-tuned on GPT-3 chains may encounter challenges in generalization when applied to our benchmark. Our proposed approach involves utilizing this newly annotated benchmark for evaluating the performance of our fine-tuned models. Notably, our observation indicates that GPT-4 currently surpasses the performance of existing pre-trained metrics, presenting an opportunity for substantial improvement. To address this gap, we intend to employ data augmentation techniques and incorporate new large-scale datasets in our training process. This strategic approach is an-

[3] https://github.com/neeraja1504/Maroc/tree/main

ticipated to enhance the generalizability and overall performance of our metrics.

7 Experimental Plan for the remaining semester

As per our project timeline, we aim to accomplish the following:

1. Obtain the human annotated labels for the complete 200 questions of GSM8K dataset. This will be used in the evaluation of our trained metric.

2. Adapt the fine-tuning strategy used by (Golovneva et al., 2022a) to train our language model(s).

3. Do Supervised fine-tuning(SFT) on the two datasets PRM800k and Mammoth.

4. Comparative analysis of the two training strategies with respect to our and existing benchmarks.

We plan to use Llama and FlanT5 models for training our metric.

References

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

I Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, Pengfei Liu, et al. 2023. Factool: Factuality detection in generative ai–a tool augmented framework for multi-task and multi-domain scenarios. *arXiv preprint arXiv:2307.13528*.

Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Eric Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2022. Scaling instruction-finetuned language models. *arXiv preprint arXiv:2210.11416*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics*, 9:391–409.

Olga Golovneva, Moya Chen, Spencer Poff, Martin Corredor, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2022a. Roscoe: A suite of metrics for scoring step-by-step reasoning. *arXiv preprint arXiv:2212.07919*.

Olga Golovneva, Pan Wei, Khadige Abboud, Charith Peris, Lizhen Tan, and Haiyang Yu. 2022b. Task-driven augmented data evaluation. In *Proceedings of the 2nd Workshop on Natural Language Generation, Evaluation, and Metrics (GEM)*, pages 18–25, Abu Dhabi, United Arab Emirates (Hybrid). Association for Computational Linguistics.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. Measuring mathematical problem solving with the math dataset. *arXiv preprint arXiv:2103.03874*.

Muhammad Khalifa, Lajanugen Logeswaran, Moontae Lee, Honglak Lee, and Lu Wang. 2023. Discriminator-guided multi-step reasoning with language models. *arXiv preprint arXiv:2305.14934*.

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners. *Advances in neural information processing systems*, 35:22199–22213.

Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050*.

Maxwell Nye, Anders Johan Andreassen, Guy Gur-Ari, Henryk Michalewski, Jacob Austin, David Bieber, David Dohan, Aitor Lewkowycz, Maarten Bosma, David Luan, et al. 2021. Show your work: Scratchpads for intermediate computation with language models. *arXiv preprint arXiv:2112.00114*.

Maxime Peyrard. 2019a. Studying summarization evaluation metrics in the appropriate scoring range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100.

Maxime Peyrard. 2019b. Studying summarization evaluation metrics in the appropriate scoring range. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 5093–5100, Florence, Italy. Association for Computational Linguistics.

Archiki Prasad, Swarnadeep Saha, Xiang Zhou, and Mohit Bansal. 2023. Receval: Evaluating reasoning chains via correctness and informativeness.

Yiwei Qin, Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2022. T5score: Discriminative fine-tuning of generative evaluation metrics. *arXiv preprint arXiv:2212.05726*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D Manning, and Chelsea Finn. 2023. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*.

Robert H Somers. 1962. A new asymmetric measure of association for ordinal variables. *American sociological review*, pages 799–811.

Liyan Tang, Tanya Goyal, Alexander R Fabbri, Philippe Laban, Jiacheng Xu, Semih Yavuz, Wojciech Kryściński, Justin F Rousseau, and Greg Durrett. 2022. Understanding factual errors in summarization: Errors, summarizers, datasets, error detectors. *arXiv preprint arXiv:2205.12854*.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Yizhong Wang, Swaroop Mishra, Pegah Alipoormolabashi, Yeganeh Kordi, Amirreza Mirzaei, Atharva Naik, Arjun Ashok, Arut Selvan Dhanasekaran, Anjana Arunkumar, David Stap, Eshaan Pathak, Giannis Karamanolakis, Haizhi Lai, Ishan Purohit, Ishani Mondal, Jacob Anderson, Kirby Kuznia, Krima Doshi, Kuntal Kumar Pal, Maitreya Patel, Mehrad Moradshahi, Mihir Parmar, Mirali Purohit, Neeraj Varshney, Phani Rohitha Kaza, Pulkit Verma, Ravsehaj Singh Puri, Rushang Karia, Savan Doshi, Shailaja Keyur Sampat, Siddhartha Mishra, Sujan Reddy A, Sumanta Patro, Tanay Dixit, and Xudong Shen. 2022. Super-NaturalInstructions: Generalization via declarative instructions on 1600+ NLP tasks. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5085–5109, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in Neural Information Processing Systems*, 35:24824–24837.

Longhui Yu, Weisen Jiang, Han Shi, Jincheng Yu, Zhengying Liu, Yu Zhang, James T Kwok, Zhenguo Li, Adrian Weller, and Weiyang Liu. 2023. Metamath: Bootstrap your own mathematical questions for large language models. *arXiv preprint arXiv:2309.12284*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023a. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Xiang Yue, Xingwei Qu, Ge Zhang, Yao Fu, Wenhao Huang, Huan Sun, Yu Su, and Wenhu Chen. 2023b. Mammoth: Building math generalist models through hybrid instruction tuning. *arXiv preprint arXiv:2309.05653*.

Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *arXiv preprint arXiv:2306.05685*.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

8 Appendix

Table 4 shows the contributions from each member so far and the distribution of work for the remaining tasks. Tables 5 shows the prompt used for generating the step by step chain and table 6 shows the prompts given to the models for evaluation of the steps in a CoT.

Task	Person(s) Responsible
Annotation of new training data	Neeraja, Akshat, Aakriti
Setting up the ROSCOE benchmark	Omkar, Rishabh, Tanay
Preprocessing the PRM800k and Mammoth dataset	Aakriti, Akshat
Reproducing results for ROSCOE	Neeraja, Omkar
Reproducing results for RECEVAL	Tanay, Rishabh
Reproducing results on ChatGPT	Neeraja, Akshat, Aakriti
Reproducing results on GPT-4	Omkar, Rishabh, Tanay
Generation of chains for different LLMs	Neeraja, Akshat, Aakriti
Modification to experiments	Omkar, Rishabh, Tanay
Documentation and Report Writing	All
Future tasks	
Finetuning using SFT approach (13th Nov- 27th Nov)	Neeraja, Akshat, Aakriti
Finetuning using ROSCOE approach (13th Nov- 27th Nov)	Omkar, Rishabh, Tanay
Meta-evaluation annotations (13th Nov- 20th Nov)	All
Documentation and Report Writing	All

Table 4: Member-wise distribution of work for the tasks done and future tasks

Question: Kyle bought last year's best-selling book for $19.50. This is with a 25% discount from the original price. What was the original price of the book?
CoT Answer: Solution: The book was originally $19.50 x 25 / 100 = $1.50 less than the original price. So the original price was $1.50 - $1.50 = $1.50. Answer: 1.

#	Steps	MISS	GRAM	FACT	LOGIC	HALL	RED	REP	COMMON	MATH
1	The book was originally $19.50 x 25 / 100 = $1.50 less than the original price.	no	no	no	no	yes	no	no	no	yes
2	So the original price was $1.50 - $1.50 = $1.50.	no	no	no	no	yes	no	no	no	yes
3	Answer: 1.	no	no	no	yes	no	no	no	no	no

Figure 3: Example of the annotation task. Humans are asked to divide the CoT answer into steps and annotate each steps for all error types.

Given a math word problem, solve it while showing your work.
Question: The Dark Lord needs to transport 1200 pounds of swords to the Enchanted Forest for a battle with the elves. He has 10 squads of 8 orcs each. How many pounds of swords does each orc have to carry?
Lets think step by step
Solution: The Dark Lord has 10 squads of 8 orcs each. To find the total number of orcs, we'll multiply the number of squads by the number of orcs in each squad. Hence total orcs = 10 squads * 8 orcs/squad = 80 orcs. To find out how many pounds of swords each orc has to carry, we'll divide the total pounds of swords by the total number of orcs: Pounds per orc = Total pounds of swords / Total orcs = 1200 / 80 = 15 pounds
Answer: 15
Question: Elsa and her sister watch an opera show every year at Central City Opera. Last year, opera seating cost $85 per ticket. This year, it costs $102. What is the percent increase in the cost of each ticket?
Lets think step by step
Solution: Last year, the opera seating cost $85 per ticket (initial cost) and this year, it costs $102 per ticket (new cost). Hence the increase in cost, is nothing but new cost - initial cost. Increase in cost = $102 - $85 = 17. To find the percent increase, divide the increase in cost by the initial cost and then multiply by 100 to express it as a percentage. Percent Increase = $(17/85)$ * 100 = 20%
Answer: 20
Question: John and his best friend Steve bought 12 cupcakes together. Each cupcake cost 1.50. If they split the costs evenly, how much did each person pay?
Lets think step by step
Solution: Each cupcake costs $1.50. To find the total cost, we'll multiply the cost per cupcake by the number of cupcakes. Hence $1.50/cupcake × 12 cupcakes = $18. Since John and Steve bought the cupcakes together, they need to split the cost evenly. To do this, we'll divide the total cost by 2 (since there are two people). Each person's share = $18 / 2 = $9
Answer: 9
Question: Two runners are competing in a 10-mile race. The first runs at an average pace of 8 minutes per mile, while the second runs at an average pace of 7 minutes per mile. After 56 minutes, the second runner stops for a drink of water. For how many minutes could the second runner remain stopped before the first runner catches up with him?
Lets think step by step
Solution: The first runner runs at an average pace of 8 minutes per mile. So, after 56 minutes, the first runner has covered. Distance of the first runner = (56 minutes) / (8 minutes/mile) = 7 miles. The second runner runs at an average pace of 7 minutes per mile. After 56 minutes, the second runner has covered: Distance of the second runner = (56 minutes) / (7 minutes/mile) = 8 miles. Hence at the 56 minute, the gap between thw two runners is 8 - 7 = 1 mile. The first runner is running at a pace of 8 minutes per mile, and he needs to cover the 1-mile gap to catch up with the second runner. Time to close the gap = (1 mile) * (8 minutes/mile) = 8 minutes. Therefore, the second runner can remain stopped for a maximum of 8 minutes before the first runner catches up with him.
Answer: 8

Table 5: Prompt given to each LLM for generating the step-by-step chains that are used in §3.1.

You are given a math problem and a step-by-step partial solution to the problem. The given solution could be correct or incorrect. Identify whether the next step to the given solution will lead to the correct final answer or not. If the next step is correct, you should generate "correct". If it is incorrect, you should generate "incorrect". I will give you a few examples to get you started.
Question: Siobhan has 2 fewer jewels than Aaron. Aaron has 5 more jewels than half of Raymond's jewels. If Raymond has 40 jewels, how many jewels does Siobhan have?
Solution: Aaron has 5 more jewels than half of Raymond's jewels, meaning he has 40 + 5 = 45 jewels. Next Step: Siobhan has 2 fewer jewels than Aaron, meaning she has 45 - 2 = 43 jewels.
Output: correct
Question: A teacher teaches 5 periods a day and works 24 days a month. He is paid $5 per period. If he has been working for 6 months now, how much has he earned in total?
Solution: The amount paid to the teacher per day is 5 periods * $5/period = $25 per day. The amount paid for 24 days is $25/day * 24 days = $600.
Next Step: The total amount for 6 months is $600 * 6 = $1800.
Output: incorrect
Question: Brandon's iPhone is four times as old as Ben's iPhone. Ben's iPhone is two times older than Suzy's iPhone. If Suzy's iPhone is 1 year old, how old is Brandon's iPhone?
Solution:
Next Step: Ben's iPhone is 2 * 1 year = 2 years older than Suzy's iPhone. Output: correct
Question: Brandon's iPhone is four times as old as Ben's iPhone. Ben's iPhone is two times older than Suzy's iPhone. If Suzy's iPhone is 1 year old, how old is Brandon's iPhone?
Solution: Ben's iPhone is 2 * 1 year = 2 years older than Suzy's iPhone.
Next Step: Thus, Brandon's iPhone is 2 + 4 years = 6 years old.
Output: incorrect
Question: Wynter went to her local town bike shop to buy her sister a bicycle as her birthday gift. While at the shop, Wynter counted 50 bicycles and 20 tricycles. How many wheels in total did the vehicles she saw have?
Solution: There are 50 bicycles at the shop. Each bicycle has 2 wheels. So, there are 50 * 2 = 100 wheels. There are 20 tricycles at the shop. Each tricycle has 3 wheels. So, there are 20 * 3 = 60 wheels.
Next Step: The total number of wheels is 100 + 60 = 160.
Output: correct
Question: Elsa and her sister watch an opera show every year at Central City Opera. Last year, opera seating cost $85 per ticket. This year, it costs $102. What is the percent increase in the cost of each ticket? Solution: Last year, the opera seating cost $85 per ticket (initial cost) and this year, it costs $102 per ticket (new cost). Hence the increase in cost, is nothing but new cost - initial cost. Increase in cost = $102 - $85 = 17. Next Step: Percent Increase = ($17 / $85) * 100 = 20%
Output: correct
Question: Two runners are competing in a 10-mile race. The first runs at an average pace of 8 minutes per mile, while the second runs at an average pace of 7 minutes per mile. After 56 minutes, the second runner stops for a drink of water. For how many minutes could the second runner remain stopped before the first runner catches up with him?
Solution: The first runner runs at an average pace of 8 minutes per mile. So, after 56 minutes, the first runner has covered. Distance of the first runner = (56 minutes) / (8 minutes/mile) = 7 miles.
Next Step: The second runner runs at an average pace of 7 minutes per mile. After 56 minutes, the second runner has covered: Distance of the second runner = (56 * 7) = 392 miles.
Output: incorrect

Table 6: Prompt used for using ChatGPT and GPT-4 as evaluation metrics.