Adapting Chain of Contradiction (CoC) for Sarcasm Detection: Extending SarcasmBench with Contradiction Evaluation

Joshua Glaspey
jo248954@ucf.edu
University of Central Florida
Orlando, Florida, USA

ABSTRACT

Sarcasm detection remains a significant challenge in Natural Language Processing (NLP) and Large Language Model (LLM) research due to subtle linguistic cues, sentiment incongruities, and the need for contextual understanding. This project extends *SarcasmBench* by implementing **Chain of Contradiction (CoC)** prompting, a CoT-based framework designed for sarcastic detection. CoC targets sentiment inconsistencies which should make it better suited for detection. CoC prompting was applied in few-shot settings across five benchmark datasets and three large language models: GPT-4o-mini, LLaMA 3-8B, and Qwen 2-7B. Performance was compared against baseline prompting methods, including zero-shot, few-shot, and CoT. Results show that CoC consistently improves recall and F1 score, particularly on datasets with direct sarcasm cues. This work demonstrates that contradiction modeling enhances sarcasm detection and offers a scalable prompting strategy for reasoning-intensive NLP tasks.

Link to GitHub Page: https://github.com/jkglaspey/adapting_chain_of_contradiction_for_sarcasm_detection

1 INTRODUCTION

Sarcasm detection remains a significant challenge in Natural Language Processing (NLP) and Large Language Model (LLM) research due to subtle linguistic cues, sentiment incongruities, and the need for contextual understanding. Sarcasm is a deeply contextual phenomenon, requiring an understanding of both literal and intended meaning, often revealed through rhetorical devices, tonal shifts, or social cues.

Traditional models in NLP, particularly RNN-based architectures, have struggled with this task due to inherent limitations in sequence modeling. These include difficulty with long-range dependencies, prefix bias, and recency bias, which often lead to a misrepresentation of early context in a sentence. CNN-based models also present challenges for sarcasm detection. While they perform well in capturing local phrase-level features, they lack adaptive attention and require increased depth to model longer dependencies, resulting in computational inefficiency and diminished interpretability. In contrast, transformer-based models—especially LLMs—have demonstrated superior performance by leveraging self-attention mechanisms that model entire sequences holistically. These models capture long-range dependencies and complex linguistic patterns, which are critical for understanding sarcasm.

Zhang et al. introduced *SarcasmBench* [26], a benchmark that evaluates 11 LLMs and 8 pre-trained language models (PLMs) across six datasets using zero-shot, few-shot, and chain-of-thought (CoT) prompting methods. However, CoT underperforms in sarcasm detection because it fails to address the non-linear and holistic nature of sarcasm.

This project extends *SarcasmBench* by implementing **Chain of Contradiction (CoC)** prompting, a CoT-based framework designed for sarcasm detection [20]. Unlike CoT, which struggles with sarcasm's indirect nature, CoC explicitly analyzes surface sentiment, infers true intent, and evaluates contradictions between them.

Problem Statement: The goal of this research is to:

- **1.** Improve sarcasm detection performance by integrating CoC prompting into *SarcasmBench's* evaluation framework.
- **2.** Compare CoC against existing prompting strategies (zero-shot, few-shot, CoT) across multiple sarcasm benchmarks.
- **3.** Demonstrate that contradiction-based reasoning enhances sarcasm comprehension.

By addressing these objectives, this project contributes to the effort of enhancing LLM's reasoning abilities in nuanced language tasks.

2 RELATED WORK

2.1 Large Language Models

LLMs have rapidly advanced in recent years, demonstrating exceptional capabilities in natural language understanding, in-context learning, and task generalization [4]. These models are trained on vast text corpora, and are capable of performing a wide range of natural language processing (NLP) tasks such as sentiment analysis [25], question answering [22], and text classification [24]. LLMs are broadly classified into general-purpose and specialized models. General purpose LLMs, such as GPT-4 [13], and LLaMA 3 [16], are designed for a wide array of applications - OpenAI has been at the forefront of these developments with ChatGPT. In contrast, specialized LLMs are fine-tuned for domain-specific tasks, such as document analysis, code generation, etc., by incorporating additional domain knowledge into their training process [12]. Generally speaking, these models have seen refinement over the years, and now have grown to incorporate billions of parameters.

The LLM ecosystem also includes proprietary and open-source models. Proprietary models such as GPT-4 are developed by organizations with private datasets and fine-tuning methods. These typically see state-of-the-art performance, but are inaccessible for direct modification. Open-source models such as LLaMA 3 have gained popularity due to their transparency, adaptability, and cost-effectiveness for research and application-specific customization.

2.2 Sarcasm Detection

Sarcasm detection is a long-standing challenge in NLP due to its reliance on implicit sentiment shifts, contextual understanding, and linguistic ambiguity. Early approaches primarily relied on rule-based and statistical methods, but statistical learning techniques, such as Support Vector Machines (SVM) and Naive Bayes (NB), were later employed to detect patterns in textual data [23]. However, these methods struggled with generalization due to their reliance on hand-crafted features and limited contextual understanding.

The adoption of deep learning methods significantly improved sarcasm detection by leveraging neural architectures such as CNNs [8], LSTMs [7], and Graph Convolutional Networks (GCNs) [9]. These models enhanced feature extraction and enabled end-to-end sarcasm classification without manual feature engineering. More recently, sarcasm detection has shifted toward pre-training models (PLMs) such as BERT [5], and RoBERTa [10], which use contextual embeddings to improve classification accuracy.

However, even advanced models often fail to capture the contradiction between surface sentiment and implied meaning—an essential feature of sarcasm. This motivates the need for prompting frameworks that can explicitly reason about such discrepancies.

2.3 Chain-of-Thought (CoT) and Chain of Contradiction (CoC) Prompting

Chain-of-Thought (CoT) prompting enhances LLM reasoning by decomposing tasks into sequential reasoning steps. Originally introduced to improve arithmetic and logical problem solving, CoT guides the model through a series of intermediate reasoning tokens $z_1, ..., z_n$ to derive a final output y shown by the equation below [26].

$$[z_1, ..., z_n, y] = p_{CoT(\theta)}(z_1, ..., z_n, y|x)$$

Wei et al. (2022) formally introduced CoT and demonstrated its effectiveness in structured problem-solving [18]. Though, its success was highly dependent on high-quality prompts. To improve adaptability, methods like Auto-CoT [27] automate prompt construction, while Tree-of-Thought (ToT) [21] and Graph-of-Thought (GoT) [3] extend into non-linear reasoning structures.

However, sarcasm detection does not follow a strict step-by-step reasoning process. To address this, CoC prompting explicitly models sarcasm as a contradiction between sentiment and true intent [20]. CoC better aligns with human sarcasm comprehension and has been shown to outperform standard CoT in sarcasm benchmarks.

3 METHODOLOGY

This project evaluates the effectiveness of CoC prompting in sarcasm detection by integrating it into the *SarcasmBench* evaluation framework. This approach involves implementing few-shot CoC on multiple LLMs and comparing its performance against zero-shot, few-shot, and Chain-of-Thought (CoT) prompting strategies.

3.1 CoC Prompting

CoC decomposes sarcasm detection into three stages:

- **Surface Sentiment Analysis:** Identifying literal sentiment via keywords, phrases, or emojis.
- **True Intention Deduction:** Analyzing rhetorical devices, tone, and common sense to infer deeper meaning.
- **Contradiction Evaluation:** Comparing sentiment and intent to classify sarcasm.

Unlike traditional CoT prompting, which assumes a stepwise reasoning process, CoC targets sentiment incongruence which should make it better suited for sarcasm detection.

The CoC prompt construction is taken from Yao et al. [20], and is provided below. Sequentially, three separate prompts provided to the LLM as a dialogue. [X] represents a placeholder for the input text to be inserted.

- **Step 1.** "Given the input sentence [X], what is the SURFACE sentiment, as indicated by clues such as keywords, sentimental phrases, emojis? Make your answer concise."
- **Step 2.** "Deduce what the sentence really means, namely the TRUE intention, by carefully checking any rhetorical devices, language style, unusual punctuations, common senses. Make your answer concise."
- **Step 3.** "Based on Step 1 and Step 2, evaluate whether the surface sentiment aligns with the true intention. If they do not match, the sentence is probably 'Sarcastic'. Otherwise, the sentence is 'Not Sarcastic'. Return the label only."

To illustrate how CoC prompting is applied in practice, Table 1 provides an example of a full prompt-response interaction. It demonstrates how each step explicitly guides the model through surface sentiment analysis, intent deduction, and contradiction evaluation.

Step	Example
Surface Sentiment	`Prompt: "What is the surface sentiment of: 'Oh great, another Monday'?"` `Response: "Positive, suggested by 'great'"`
True Intention	`Prompt: "What is the true intention of the sentence?"` `Response: "Negative—speaker dislikes Mondays"`
Contradiction Evaluation	`Prompt: "Do surface and true sentiment match?"` `Response: "No → Sarcastic"`

Table 1: Example of CoC prompting interaction with sarcasm detection. The prompts are simplified in this example to emphasize the interaction between the inputs and LLM.

3.2 Benchmark Datasets

To ensure a fair evaluation, five of the original six sarcasm detection datasets from *SarcasmBench* are utilized. Any augmentations to each dataset with respect to the original experiment [26] is mentioned in their descriptions, but due to large-scale size similarities, the results are generalizable. The Riloff [15] dataset was omitted for this experiment due to it not being publicly available for download (not available in her publications page).

- **IAC-V1 [11] & IAC-V2 [14]:** Online debate corpora containing sarcastic and non-sarcastic comments. IAC-V1 describes

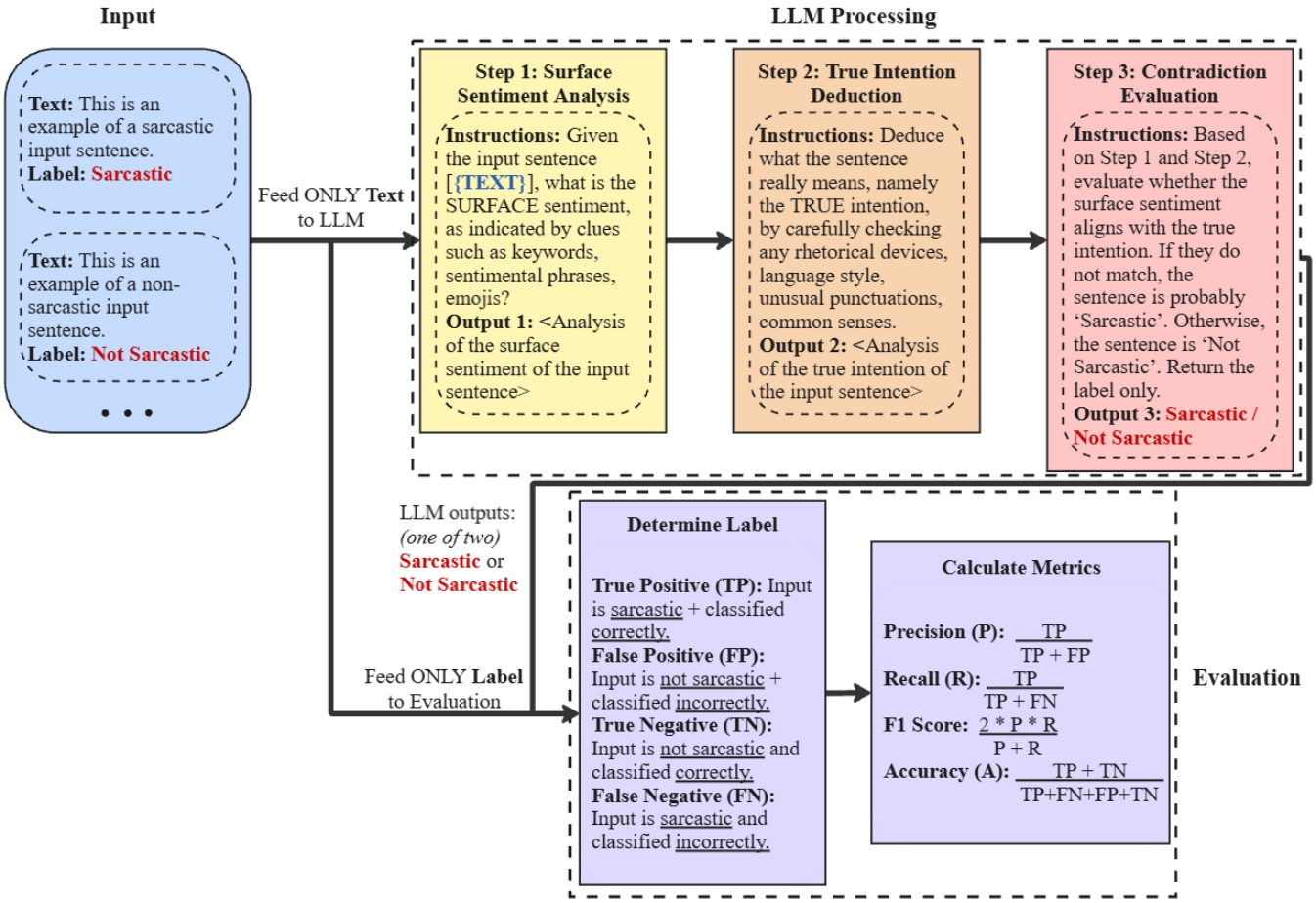

Figure 1: Input Processing Pipeline. Input sentences are provided labeled as either "Sarcastic" or "Not Sarcastic." The text prompt is fed to the LLM being evaluated, and judged to be sarcastic or not through CoC prompting (3 stages). This output is compared to the ground truth label, updating the evaluation metrics (P, R, F1, A).

1995 data in the dataset, but only 1993 were scanned in the program. This dataset is missing two of the original inputs. IAC-V2 matches to the version used in the original.

- **Ghosh [6]:** A large Twitter dataset with sarcasm-labeled tweets. In the original evaluation, 7,804 noisy tweets were filtered out with no other explanation or reference. This experiment includes those additional noisy tweets.
- **iSarcasmEval [2]:** A dataset where authors explicitly indicate sarcastic intent. This dataset matches exactly to the version used in the original experiment.
- **SemEval 2018 Task 3 [17]:** A benchmark dataset for irony detection in English tweets. This dataset matches exactly to the version used in the original experiment.

Dataset	Total Inputs	Sarcastic	Non-Sarcastic
IAC-V1	1,993	998	995
IAC-V2	6,520	3,260	3,260
Ghosh	41,780	19,488	22,292
iSarcasmEval	1,400	200	1,200
SemEval 2018	4,618	2,222	2,396

Table 2: Dataset statistics used in this study.

Table 2 tabulates the number of sarcastic/non-sarcastic data within each dataset.

To ensure consistency across datasets and reduce preprocessing-related noise, a series of data cleanup procedures were applied before evaluation. Each dataset was parsed programmatically to extract the input text and corresponding sarcasm label. Given differences in formatting across sources, a custom script was used to normalize inputs and filter malformed entries.

Notably, emojis were converted into token-friendly text markers. All emojis in the input text were replaced by hashtag-style tags corresponding to their semantic meaning. For example, a smiley face emoji "☺" was converted to "#smile", and similar substitutions were applied using the Unicode emoji name. This normalization allowed the language models to interpret emotional tone more

reliably, as many LLMs struggle with raw emoji symbols in plaintext prompts.

Inputs that failed to conform to expected structure, such as missing labels or corrupt formatting, were excluded during parsing. Furthermore, if an LLM failed to produce a valid sarcasm label after the CoC prompting stages (e.g., generating a conversational response instead of returning "Sarcastic" or "Not Sarcastic"), that input was also skipped from evaluation. These measures ensured the integrity of the evaluation set and helped isolate model behavior from data-related inconsistencies.

3.3 Models

Token estimation was conducted to assess model cost-efficiency. Each input text averages 50 tokens, while each CoC prompt interaction (3 steps) requires an estimated 223 tokens. With 56,311 total inputs and a 3:1 input-output ratio, the total token budget is approximated as:

$$((223 + 50) \times 56,311) \times \frac{4}{3} \approx 20,497,204 \text{ tokens.}$$

GPT-4o-mini is selected for affordability, priced at $0.60/M input tokens and $2.40/M output tokens. In contrast, GPT-4 Turbo costs $10.00/M and $30.00/M respectively.

Due to the resource constraints of this project, only a subset of models were used in the final evaluation—selected for their balance of performance and feasibility in local or affordable deployment. Three large language models were evaluated:

- **GPT-4o-mini:** Chosen for its low-cost inference through the OpenAI API, this model provides a practical baseline for high-performance prompting strategies.
- **LLaMA 3-8B:** An open-source model that supports local inference with acceptable memory requirements, making it ideal for experimentation under hardware limitations.
- **Qwen 2-7B:** Another open-source model capable of local deployment, selected for its recent performance on reasoning tasks and its ability to support multi-step prompting [19].

Both LLaMA and Qwen were run locally using the `oLLaMA` framework [1], which simplifies deployment and hardware integration for large models. This allowed for controlled experimentation without API throttling or cost limitations.

4 EVALUATION & RESULTS

4.1 Experimental Setup

CoC will be evaluated on each of the provided models using each of the available datasets. The previous results of zero-shot I/O prompting, few-shot I/O prompting, and few-shot CoT prompting will be compared directly to the results of few-shot CoC prompting. From *SarcasmBench*, Random, ChatGPT (GPT-3.5), GPT-4 Turbo, LLaMA 3-8B, and Qwen 2-7B will be selected. Performance will directly be compared using the evaluation metrics provided in the next section.

4.2 Evaluation Metrics

The following evaluation metrics are compared: **precision (P), recall (R), accuracy (A),** and **F1 score.** The metrics are calculated using the following equations.

$$P = \frac{TP}{TP+FP}$$
$$R = \frac{TP}{TP+FN}$$
$$F1 = \frac{2*P*R}{P+R}$$
$$A = \frac{TP+TN}{TP+FN+FP+TN}$$

In these equations, a positive detection is sarcastic, and a negative detection is not sarcastic. Therefore, TP (True Positives) is the number of sarcastic samples correctly identified, FP (False Positives) is the number of non-sarcastic samples identified as sarcastic, FN (False Negatives) is the number of sarcastic samples identified as not sarcastic, and TN (True Negatives) is the number of non-sarcastic samples identified as not sarcastic.

4.3 Performance Analysis

Table 3 reports accuracy, precision, recall, and F1 scores across all five datasets using multiple prompting methods and models. The results highlight several key trends:

1. CoC improves recall but slightly reduces precision. GPT-4o-mini using few-shot CoC achieves consistently high recall (87.6% on IAC-V1, 94.0% on IAC-V2), outperforming few-shot CoT and I/O in recall across most datasets - ignoring GPT-4 Turbo's recall scores. However, this comes at the cost of lower precision (e.g., 59.3% on IAC-V1), suggesting the model over-predicts sarcasm when contradiction is detected.

2. LLaMA 3-8B benefits most from CoC prompting. While its zero-shot and few-shot CoT performance lags, LLaMA 3-8B improves significantly under few-shot CoC prompting. On Ghosh, the F1 score rises from 50.6% to 79.0%, indicating that CoC helps weaker models align more closely with the reasoning required for sarcasm detection. This effect is likely because Ghosh contains direct and often exaggerated sarcastic cues, which are easier to parse with CoC's structured contradiction steps. However, the only counterexample is found on the SemEval dataset, where recall decreases from 100.0% (zero-shot IO) to 58.4% (few-shot CoC), suggesting that LLaMA's CoC implementation can miss subtle cues when sarcasm is more abstract or reliant on implicit irony.

3. Qwen 2-7B shows the highest variability. Although Qwen underperforms in zero-shot and few-shot CoT prompting, it shows significant improvements with few-shot CoC—especially on Ghosh, where F1 increases from 55.9% to 85.0% - the best performance of any model. This suggests that Qwen benefits from the explicit intermediate reasoning stages provided by CoC. Ghosh's tweet-based structure features overt sarcasm marked by hashtags or emojis, making it well-suited to contradiction-based modeling. However, in data sets such as SemEval or iSarcasmEval, which contain more implicit or author-labeled sarcasm, Qwen's performance remains inconsistent, likely due to the challenges in identifying intention without strong lexical indicators.

4. CoC's effectiveness is dataset-dependent. While Ghosh and both IAC datasets show substantial improvements with CoC—due to their contextual structure and conversational or tweet-based sarcasm—iSarcasmEval and SemEval yield more modest gains. These latter datasets include sarcasm that is often inferred by authors rather than directly expressed, requiring higher-level pragmatics or world knowledge that even structured prompting cannot fully resolve. CoC performs best on datasets where sarcasm manifests

Table 3: Performance on five datasets. Bold indicates models using the CoC prompting method. Underlined values represent the best results across LLMs. From SarcasmBench, ChatGPT 3.5, GPT-4 Turbo, LLaMA 3-8B, and Qwen 2-7B using zero-shot IO, few-shot IO, and few-shot CoT are used. GPT 4.0 mini, LLaMA 3-8B, and Qwen 2-7B using few-shot CoC prompting are used.

	IAC-V1				IAC-V2				iSarcasmEval			
Model	A	P	R	F1	A	P	R	F1	A	P	R	F1
ChatGPT (Zero-shot IO)	63.6	61.2	81.8	70.0	56.4	50.2	91.6	64.9	51.6	14.3	91.7	26.2
GPT-4 Turbo (Zero-shot IO)	72.2	73.3	85.1	78.7	71.4	65.1	92.9	76.6	65.6	25.6	89.5	39.8
ChatGPT (Few-shot IO)	69.4	74.3	72.1	73.2	72.2	67.8	83.1	75.1	76.1	34.7	85.3	49.2
GPT-4 Turbo (Few-shot IO)	73.3	75.4	84.6	79.6	74.5	70.0	86.2	77.2	79.3	37.0	89.5	52.3
ChatGPT (Few-shot CoT)	64.7	64.4	81.5	69.6	61.9	56.4	82.8	67.3	53.6	15.6	87.7	33.6
GPT-4 Turbo (Few-shot CoT)	72.2	72.4	85.9	78.6	69.5	63.4	93.0	75.4	65.9	25.4	86.8	39.3
GPT-4o-mini (Few-shot CoC)	63.7	59.3	87.6	70.7	67.7	61.6	94.0	74.4	53.6	22.3	90.5	35.8
LLaMA 3-8B (Zero-shot IO)	60.4	60.1	93.5	73.8	52.1	51.2	97.8	67.2	13.4	12.8	100.0	22.7
LLaMA 3-8B (Few-shot IO)	43.4	61.5	12.9	21.3	56.4	68.1	24.7	36.3	84.6	21.4	7.9	11.5
LLaMA 3-8B (Few-shot CoT)	52.5	61.7	53.2	57.1	52.3	52.2	57.8	54.9	55.5	13.2	44.7	20.4
LLaMA 3-8B (Few-shot CoC)	61.9	63.9	53.1	58.0	67.1	68.8	60.9	64.6	68.6	27.5	74.2	40.2
Qwen 2-7B (Zero-shot IO)	56.6	36.3	76.9	49.3	51.8	28.9	57.8	38.6	46.1	18.2	36.4	24.3
Qwen 2-7B (Few-shot IO)	61.5	63.5	84.7	73.0	53.5	53.3	75.1	62.7	33.0	16.7	78.8	26.4
Qwen 2-7B (Few-shot CoT)	54.7	42.5	51.4	46.5	52.9	40.1	56.4	40.1	53.7	20.3	71.8	28.5
Qwen 2-7B (Few-shot CoC)	61.5	71.0	39.0	50.3	68.8	78.2	52.1	62.5	77.1	31.7	52.5	39.6

	SemEval Task 3				Ghosh				**Average Scores**			
Model	A	P	R	F1	A	P	R	F1	A	P	R	F1
ChatGPT (Zero-shot IO)	52.2	48.3	99.7	65.1	63.3	58.2	90.4	71.4	57.4	46.4	91.0	59.5
GPT-4 Turbo (Zero-shot IO)	76.1	62.8	98.1	76.5	79.8	73.5	93.3	82.2	73.0	60.1	91.8	70.8
ChatGPT (Few-shot IO)	68.9	60.9	92.6	71.2	76.8	72.3	86.2	75.4	72.7	62.0	83.9	68.8
GPT-4 Turbo (Few-shot IO)	81.1	68.3	97.7	80.4	83.9	80.7	88.9	84.6	78.4	66.3	89.4	74.8
ChatGPT (Few-shot CoT)	64.9	53.4	84.4	65.4	69.8	64.3	80.9	71.7	63.0	50.8	83.5	61.5
GPT-4 Turbo (Few-shot CoT)	75.1	61.8	97.7	75.7	80.8	74.5	93.6	83.0	72.7	59.5	91.4	70.4
GPT-4o-mini (Few-shot CoC)	65.5	60.5	81.8	69.5	75.8	65.9	99.6	79.3	65.3	53.9	90.7	65.9
LLaMA 3-8B (Zero-shot IO)	40.2	39.9	100.0	57.0	50.4	50.2	99.7	66.8	43.3	42.8	98.2	57.5
LLaMA 3-8B (Few-shot IO)	60.8	51.7	20.0	28.8	51.3	63.3	6.2	11.3	59.3	53.2	14.3	21.8
LLaMA 3-8B (Few-shot CoT)	51.1	40.9	51.8	45.7	49.9	49.9	51.4	50.6	52.3	43.6	51.8	45.7
LLaMA 3-8B (Few-shot CoC)	65.8	66.6	58.4	62.2	78.3	72.6	86.5	79.0	68.3	59.9	66.6	60.8
Qwen 2-7B (Zero-shot IO)	45.1	30.7	51.0	38.3	65.6	54.4	87.7	68.3	53.0	33.7	62.0	43.8
Qwen 2-7B (Few-shot IO)	54.7	50.3	95.8	65.7	57.9	55.6	93.2	69.8	52.1	47.9	85.5	59.5
Qwen 2-7B (Few-shot CoT)	56.7	47.4	60.2	50.9	61.4	61.3	57.6	55.9	55.9	42.3	59.5	44.4
Qwen 2-7B (Few-shot CoC)	64.2	69.5	45.5	55.0	85.0	79.8	90.1	85.0	71.3	66.0	55.8	58.5

through clear surface-intent mismatches; in contrast, when sarcastic intent is subtle, indirect, or annotation-driven, performance gains are limited.

5. GPT-4 Turbo remains strongest overall. Despite CoC's gains, GPT-4 Turbo with few-shot I/O or CoT often holds the best balance between precision and recall. However, GPT-4o-mini offers comparable F1 in many cases at a fraction of the cost.

4.4 Per-Dataset Observations

IAC-V1: GPT-4 Turbo achieves the highest F1 score, though GPT-4o-mini with CoC shows the best recall (of GPT models). CoC helps models better identify sentiment mismatches, but precision remains a challenge due to more subtle sarcasm in early forum threads.

IAC-V2: GPT-4 Turbo again performs strongest overall, but LLaMA 3-8B shows notable improvements under CoC prompting. The structured prompt helps open-source models adapt to the nuanced rhetorical question subset.

Ghosh: CoC shows the largest gain across all models, particularly for Qwen 2-7B. The overt sarcasm markers in the dataset, such as hashtags and exaggerated tone, align well with CoC's contradiction-driven structure.

iSarcasmEval: All models perform modestly. The author-labeled sarcasm of the dataset and the short input length reduce the effectiveness of CoC, possibly due to the limited context for contradiction modeling.

SemEval 2018 Task 3: CoC improves precision but often misses recall, especially for Qwen and LLaMA. This reflects the difficulty in extracting intent when sarcasm is indirect or ironic rather than sentiment driven.

5 DISCUSSION & CHALLENGES

The results of this project demonstrate that Chain of Contradiction (CoC) prompting can serve as a valuable enhancement to traditional prompting strategies in sarcasm detection, particularly when integrated with large language models like GPT-4o-mini, LLaMA 3-8B, and Qwen 2-7B. The CoC framework consistently improved recall across most datasets, especially those containing direct sarcasm cues (e.g., Ghosh and IAC-V2) - which may have benefitted from the data cleanup of converting emojis to text. This suggests that contradiction-based reasoning helps models detect implicit sentiment mismatches that are often overlooked by conventional zero-shot or few-shot prompting. Additionally, the improvement seen in open-source models like LLaMA 3-8B and Qwen 2-7B highlights CoC's adaptability beyond proprietary APIs.

However, several challenges persisted throughout the project. First, model outputs under CoC prompting were not always well-behaved. Some inputs resulted in unstructured or dialogue-style completions instead of the required "Sarcastic" or "Not Sarcastic" labels-particularly with LLaMA 3-8B. These instances were filtered out, but they represent a gap in prompt controllability that could impact reproducibility. Second, performance on more subtle datasets like iSarcasmEval and SemEval remained limited, even under CoC prompting. This reflects the broader difficulty of sarcasm detection when cues are ambiguous, author-labeled, or heavily reliant on external knowledge. Additionally, maintaining consistent formatting across five diverse datasets required substantial preprocessing and filtering, which may have introduced mild distributional shifts in comparison to SarcasmBench's original configuration.

Future improvements to this work can build upon the extended SarcasmCue framework proposed by Yao et al. [20], which introduces three additional prompting strategies beyond CoC: **Graph of Cues (GoC)**, **Bagging of Cues (BoC)**, and **Tensor of Cues (ToC)**. GoC organizes multiple linguistic, emotional, and contextual cues into a graph structure, allowing LLMs to flexibly traverse interconnected reasoning paths. BoC adopts an ensemble-style approach, where diverse cue subsets are randomly sampled and evaluated independently, with predictions aggregated via majority voting. ToC goes a step further by fusing linguistic, contextual, and emotional cues into a high-dimensional tensor representation, enabling higher-order cue interactions. These non-linear, multi-cue prompting strategies may provide more robust generalization than CoC—especially on datasets where sarcasm cues are subtle or contextually entangled.

Further improvements can be made by increasing the robustness and generalizability of the evaluation. Reinforcement learning techniques can be explored to dynamically optimize the CoC prompting steps. For example, a reward-guided policy could penalize contradictions between predicted and reference labels. Such strategies may help stabilize model behavior, especially in edge cases where sarcasm is weakly expressed or contextually ambiguous. Additionally, replicating this experiment across a broader range of models—especially instruction-tuned or PLLMs—would help validate whether the observed dataset-specific trends persist beyond the current model set. This could clarify whether certain sarcasm benchmarks are intrinsically harder or simply more sensitive to architectural or prompting differences.

6 CONCLUDING REMARKS

This project investigated the effectiveness of Chain of Contradiction (CoC) prompting in sarcasm detection by integrating it into the SarcasmBench evaluation framework. Through structured decomposition of surface sentiment and inferred intent, CoC provided a more targeted reasoning strategy for identifying sarcasm in contextually rich text. Across five benchmark datasets and three large language models—GPT-4o-mini, LLaMA 3-8B, and Qwen 2-7B—CoC prompting demonstrated consistent improvements in recall and F1 score over traditional prompting baselines, particularly on datasets featuring explicit or structurally simple sarcasm. These results suggest that contradiction-based reasoning is a viable path toward enhancing LLM comprehension of nuanced language phenomena.

While performance varied by dataset and model, the findings highlight several promising directions for future work. Incorporating more advanced prompting strategies—such as Graph of Cues or Tensor of Cues—may further improve sarcasm detection by modeling multi-faceted reasoning over emotional and contextual signals. Additional model replication and experimentation will be necessary to validate the observed trends across architectures and domains. Ultimately, this project successfully met its goals of integrating CoC prompting into an established sarcasm detection framework, evaluating its performance across multiple models and benchmarks, and demonstrating that contradiction-based reasoning can meaningfully enhance large language models' ability to detect sarcasm.

REFERENCES

[1] [n. d.]. Ollama — ollama.com. https://ollama.com/. https://github.com/ollama/ollama.

[2] Ibrahim Abu Farha, Silviu Vlad Oprea, Steven Wilson, and Walid Magdy. 2022. SemEval-2022 Task 6: iSarcasmEval, Intended Sarcasm Detection in English and Arabic. In *Proceedings of the 16th International Workshop on Semantic Evaluation (SemEval-2022)*, Guy Emerson, Natalie Schluter, Gabriel Stanovsky, Ritesh Kumar, Alexis Palmer, Nathan Schneider, Siddharth Singh, and Shyam Ratan (Eds.). Association for Computational Linguistics, Seattle, United States, 802–814. https://doi.org/10.18653/v1/2022.semeval-1.111

[3] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of Thoughts: Solving Elaborate Problems with Large Language Models. *Proceedings of the AAAI Conference on Artificial Intelligence* 38, 16 (March 2024), 17682–17690. https://doi.org/10.1609/aaai.v38i16.29720

[4] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight,

Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. 2021. Evaluating Large Language Models Trained on Code. arXiv:2107.03374 [cs.LG] https://arxiv.org/abs/2107.03374

[5] Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, Jill Burstein, Christy Doran, and Thamar Solorio (Eds.). Association for Computational Linguistics, Minneapolis, Minnesota, 4171–4186. https://doi.org/10.18653/v1/N19-1423

[6] Aniruddha Ghosh and Tony Veale. 2016. Fracking Sarcasm using Neural Network. In *Proceedings of the 7th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis*, Alexandra Balahur, Erik van der Goot, Piek Vossen, and Andres Montoyo (Eds.). Association for Computational Linguistics, San Diego, California, 161–169. https://doi.org/10.18653/v1/W16-0425

[7] Debanjan Ghosh, Alexander R. Fabbri, and Smaranda Muresan. 2018. Sarcasm Analysis using Conversation Context. arXiv:1808.07531 [cs.CL] https://arxiv.org/abs/1808.07531

[8] Deepak Jain, Akshi Kumar, and Geetanjali Garg. 2020. Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN. *Applied Soft Computing* 91 (03 2020), 106198. https://doi.org/10.1016/j.asoc.2020.106198

[9] Bin Liang, Chenwei Lou, Xiang Li, Min Yang, Lin Gui, Yulan He, Wenjie Pei, and Ruifeng Xu. 2022. Multi-Modal Sarcasm Detection via Cross-Modal Graph Convolutional Network. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Smaranda Muresan, Preslav Nakov, and Aline Villavicencio (Eds.). Association for Computational Linguistics, Dublin, Ireland, 1767–1777. https://doi.org/10.18653/v1/2022.acl-long.124

[10] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. 2019. RoBERTa: A Robustly Optimized BERT Pretraining Approach. arXiv:1907.11692 [cs.CL] https://arxiv.org/abs/1907.11692

[11] Stephanie Lukin and Marilyn Walker. 2017. Really? Well. Apparently Bootstrapping Improves the Performance of Sarcasm and Nastiness Classifiers for Online Dialogue. arXiv:1708.08572 [cs.CL] https://arxiv.org/abs/1708.08572

[12] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. 2024. A Comprehensive Overview of Large Language Models. arXiv:2307.06435 [cs.CL] https://arxiv.org/abs/2307.06435

[13] OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. GPT-4 Technical Report. arXiv:2303.08774 [cs.CL] https://arxiv.org/abs/2303.08774

[14] Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2017. Creating and Characterizing a Diverse Corpus of Sarcasm in Dialogue. arXiv:1709.05404 [cs.CL] https://arxiv.org/abs/1709.05404

[15] Ellen Riloff, Ashequl Qadir, Prafulla Surve, Lalindra De Silva, Nathan Gilbert, and Ruihong Huang. 2013. Sarcasm as Contrast between a Positive Sentiment and Negative Situation. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, David Yarowsky, Timothy Baldwin, Anna Korhonen, Karen Livescu, and Steven Bethard (Eds.). Association for Computational Linguistics, Seattle, Washington, USA, 704–714. https://aclanthology.org/D13-1066/

[16] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023. LLaMA: Open and Efficient Foundation Language Models. arXiv:2302.13971 [cs.CL] https://arxiv.org/abs/2302.13971

[17] Cynthia Van Hee, Els Lefever, and Véronique Hoste. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In *Proceedings of the 12th International Workshop on Semantic Evaluation*, Marianna Apidianaki, Saif M. Mohammad, Jonathan May, Ekaterina Shutova, Steven Bethard, and Marine Carpuat (Eds.). Association for Computational Linguistics, New Orleans, Louisiana, 39–50. https://doi.org/10.18653/v1/S18-1005

[18] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. arXiv:2201.11903 [cs.CL] https://arxiv.org/abs/2201.11903

[19] An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jianxin Yang, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Xuejing Liu, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, Zhifang Guo, and Zhihao Fan. 2024. Qwen2 Technical Report. arXiv:2407.10671 [cs.CL] https://arxiv.org/abs/2407.10671

[20] Ben Yao, Yazhou Zhang, Qiuchi Li, and Jing Qin. 2024. Is Sarcasm Detection A Step-by-Step Reasoning Process in Large Language Models? arXiv:2407.12725 [cs.CL] https://arxiv.org/abs/2407.12725

[21] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023. Tree of Thoughts: Deliberate Problem Solving with Large Language Models. arXiv:2305.10601 [cs.CL] https://arxiv.org/abs/2305.10601

[22] Zhou Yu, Xuecheng Ouyang, Zhenwei Shao, Meng Wang, and Jun Yu. 2023. Prophet: Prompting Large Language Models with Complementary Answer Heuristics for Knowledge-based Visual Question Answering. arXiv:2303.01903 [cs.CV] https://arxiv.org/abs/2303.01903

[23] Yazhou Zhang, Dan Ma, Prayag Tiwari, Chen Zhang, Mehedi Masud, Mohammad Shorfuzzaman, and Dawei Song. 2023. Stance-level Sarcasm Detection with BERT and Stance-centered Graph Attention Networks. *ACM Trans. Internet Technol.* 23, 2, Article 27 (May 2023), 21 pages. https://doi.org/10.1145/3533430

[24] Yazhou Zhang, Mengyao Wang, Chenyu Ren, Qiuchi Li, Prayag Tiwari, Benyou Wang, and Jing Qin. 2024. Pushing The Limit of LLM Capacity for Text Classification. arXiv:2402.07470 [cs.CL] https://arxiv.org/abs/2402.07470

[25] Yazhou Zhang, Mengyao Wang, Youxi Wu, Prayag Tiwari, Qiuchi Li, Benyou Wang, and Jing Qin. 2024. DialogueLLM: Context and Emotion Knowledge-Tuned Large Language Models for Emotion Recognition in Conversations. arXiv:2310.11374 [cs.CL] https://arxiv.org/abs/2310.11374

[26] Yazhou Zhang, Chunwang Zou, Zheng Lian, Prayag Tiwari, and Jing Qin. 2024. SarcasmBench: Towards Evaluating Large Language Models on Sarcasm Understanding. arXiv:2408.11319 [cs.CL] https://arxiv.org/abs/2408.11319

[27] Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic Chain of Thought Prompting in Large Language Models. arXiv:2210.03493 [cs.CL] https://arxiv.org/abs/2210.03493