

Is Sarcasm Detection A Step-by-Step Reasoning Process in Large Language Models?

Ben Yao^a, Yazhou Zhang^{b,c}, Qiuchi Li^a, Jing Qin^b

^aUniversity of Copenhagen, ^bThe Hong Kong Polytechnic University, ^cTianjin University

Abstract

Elaborating a series of intermediate reasoning steps significantly improves the ability of large language models (LLMs) to solve complex problems, as such steps would evoke LLMs to think sequentially. However, human sarcasm understanding is often considered an intuitive and holistic cognitive process, in which various linguistic, contextual, and emotional cues are integrated to form a comprehensive understanding of the speaker’s true intention, which is argued not be limited to a step-by-step reasoning process. To verify this argument, we introduce a new prompting framework called **SarcasmCue**, which contains four prompting strategies, *viz.* chain of contradiction (CoC), graph of cues (GoC), bagging of cues (BoC) and tensor of cues (ToC), which elicits LLMs to detect human sarcasm by considering sequential and non-sequential prompting methods. Through a comprehensive empirical comparison on four benchmarking datasets, we show that the proposed four prompting methods outperforms standard IO prompting, CoT and ToT with a considerable margin, and non-sequential prompting generally outperforms sequential prompting.

1 Introduction

Sarcasm is a subtle linguistic phenomenon that uses rhetorical devices such as hyperbole and figuration to convey true sentiments and intentions that are opposite to the literal meanings of the words used (Wen et al., 2023; Zhang et al., 2023b). Sarcasm detection aims to combine different types of cues, such as linguistic features, contextual information, emotional knowledge, to form a comprehensive understanding of the author’s sarcastic attitude. Owing to its inherent ambivalence and figurative nature, sarcasm detection has persistently proven a formidable challenge spanning the

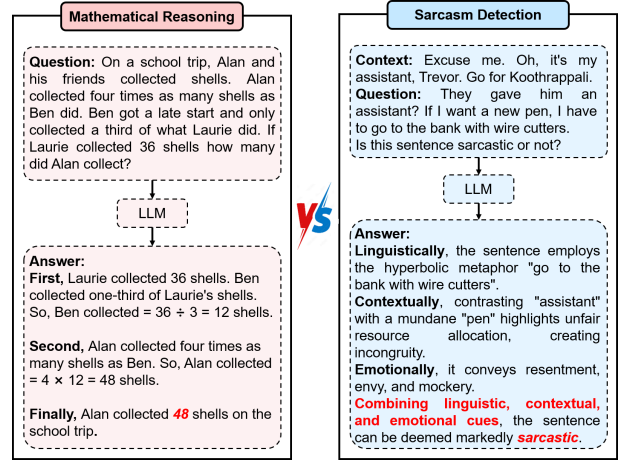


Figure 1: The comparison of the processes of mathematical reasoning and sarcasm detection.

eras from feature engineering to prompt engineering (Yue et al., 2023; Zhang et al., 2023a).

Recent large language models have demonstrated impressive performance in downstream natural language processing (NLP) tasks, in which “System 1” - the fast, unconscious, and intuitive tasks, e.g., sentiment classification, topic analysis, etc., have been argued to be successfully performed (Cui et al., 2024). Instead, increasing efforts have been devoted to the other class of tasks - “System 2”, which requires slow, deliberative and multi-step thinking, such as logical, mathematical, and commonsense reasoning tasks (Wei et al., 2022). To improve the ability of LLMs to solve such complex problems, a widely adopted technique is to decompose complex problems into a series of intermediate solution steps prior to answer generation, and elicit LLMs to think step-by-step, such as chain of thought (CoT) (Wei et al., 2022), tree of thought (ToT) (Yao et al., 2024), graph of thought (GoT) (Besta et al., 2024), etc.

However, sarcasm detection, as a holistic, intuitive, and non-rational cognitive process, is ar-

guably in noncompliance with step-by-step logical reasoning due to two main reasons: (1) sarcasm expression does not strictly conform to formal logical structures, such as the law of hypothetical syllogism (i.e., $\text{if } A \Rightarrow B \text{ and } B \Rightarrow C, \text{ then } A \Rightarrow C$). For example, “*Poor Alice has fallen for that stupid Bob; and that stupid Bob is head over heels for Claire; but don’t assume for a second that Alice would like Claire*”; (2) sarcasm judgment is typically a fluid combination of various cues, where each cue holds equal importance to the judgment of sarcasm, and there is no rigid sequence of steps among them. As shown in Fig. 1, linguistic, contextual and emotional factors are all crucial for rendering the sentence as sarcastic. Hence, the main research question can be summarized as:

RQ: *Is human sarcasm detection a step-by-step reasoning process?*

To answer this question, we propose a theoretical framework, called **SarcasmCue**, based on the sequential and non-sequential prompting paradigm. It consists of four prompting methods, i.e., *chain of contradiction* (CoC), *graph of cues* (GoC), *bagging of cues* (BoC) and *tensor of cues* (ToC). A *cue* is similar to a *thought*, which is concretely a coherent language sequence related to linguistics, context, or emotion that serves as an intermediate indicator toward identifying sarcasm, such as rhetorical devices, emotional words, etc. Each of the four prompting methods has its own focus and advantages. Specifically,

- **CoC.** It builds upon CoT prompting and harnesses the quintessential property of sarcasm (namely the contradiction between surface sentiment and true intention). It aims to: (1) identify the literal meaning and surface sentiment by extracting keywords, sentimental phrases, etc.; (2) deduce the true intention by scrutinizing special punctuation, rhetorical devices, cultural background, etc.; and (3) determine the inconsistency between surface sentiment and true intention. It has a typical linear structure.
- **GoC.** Generalizing over CoC, GoC frames the problem of sarcasm detection as a search over a graph and treats various cues (e.g., linguistic, contextual, emotional cues, etc.) as nodes, with the relations across cues represented as edges. Different from CoC and ToT, It allows language models to flexibly choose and weigh multiple cues when detecting sarcasm, rather than following a fixed hierarchy or linear reasoning path,

unconstrained by the need for unique predecessor nodes. It represents a graphical structure. In summary, both CoC and GoC follow a step-by-step reasoning process.

- **BoC.** In contrast, BoC and ToC are proposed based on the assumption that sarcasm detection is not a step-by-step reasoning process. BoC is a bagging approach that constructs a pool of diverse cues and creates multiple cue subsets through randomly sampling q cues at each round. LLMs are employed to generate multiple predictions based on these subsets, and such predictions are aggregated to produce the final result via majority voting. It has a set-based structure.
- **ToC.** ToC treats each type of cues (namely linguistic, contextual, and emotional cues) as an independent, orthogonal view for sarcasm understanding and constructs a multi-view representation through the tensor product of these three types of cues. It allows language models to leverage higher-order interactions among the cues. ToC can be visualized as a 3D volumetric structure, where each coordinate axis corresponds to a distinct type of cue. This tensorial method aims to offer a more comprehensive and expressive means of fusing diverse cues.

We present empirical evaluations of the proposed prompting approaches across four sarcasm detection benchmarks over 2 SOTA LLMs (i.e., GPT-4o, LLaMA 3-8B), and compare their results against 3 SOTA prompting approaches (i.e., standard IO prompting, CoT, ToT). We show that the proposed four prompting methods outperforms standard IO prompting, CoT and ToT with a margin of 2%, and non-sequential prompting generally outperforms sequential prompting. Between the two LLMs, GPT-4o consistently beats LLaMA by a striking margin across all tasks.

The main contributions are concluded as follows:

- Our work is the first to investigate the step-wise nature of sarcasm judgment by using both sequential and non-sequential prompting methods.
- We propose a new prompting framework that consists of four sub-methods, *viz.* chain of contradiction (CoC), graph of cues (GoC), bagging of cues (BoC) and tensor of cues (ToC).

- Comprehensive experiments over four datasets demonstrate the superiority of the proposed prompting framework in zero-shot sarcasm detection.

2 Related Work

This section reviews two lines of research that form the basis of this work: CoT prompting and sarcasm detection.

2.1 Chain-of-Thought Prompting

Inspired by the step-by-step thinking ability of humans, CoT prompting was proposed to “prompt” language models to produce intermediate reasoning steps that lead to the final answer. Wei et al. (2022) made a formal definition of CoT prompting in LLMs and proved its effectiveness by presenting empirical evaluations on arithmetic reasoning benchmarks. This work pioneered the use of CoT prompting in NLP. However, its performance hinged on the quality of manually crafted prompts, which was a costly and unstable process. To fill this gap, Auto-CoT was proposed to automatically construct demonstrations with questions and reasoning chains (Zhang et al., 2022). Different from Auto-CoT, Diao et al. (2023) presented an Active-Prompt approach to determine which questions were the most important and helpful to annotate from a pool of task-specific queries, for reducing the human engineering workload. The impressive results of CoT prompting have sparked a surge of exploration into designing CoT prompting strategies across various tasks (Fei et al., 2023; Li et al., 2023; Zheng et al., 2023). For instance, Wang et al. (2024) used formal grammars as the intermediate reasoning steps for domain-specific language generation.

Furthermore, Yao et al. (2024) introduced a non-chain prompting framework, namely ToT, which made LLMs consider multiple different reasoning paths and self-evaluated choices to decide the next course of action. They proved the effectiveness of the ToT approach on the tasks requiring non-trivial planning or search. Beyond CoT and ToT approaches, Besta et al. (2024) modeled the information generated by an LLM as an arbitrary graph (i.e., GoT), where units of information were considered as vertices and the dependencies between these vertices were edges. Although the above-mentioned approaches have shown exceptional performance on various arithmetic and logical reasoning tasks, all of them adopt the sequential decoding

paradigm of “let LLMs think step by step”. Contrarily, it is argued that sarcasm judgment does not conform to step-by-step logical reasoning, and there is a need to develop non-sequential prompting approaches.

2.2 Sarcasm Detection

Sarcasm detection is habitually treated as a text classification task, where the target is to identify whether the given text is sarcastic or not (Zhang et al., 2024). It has evolved from early rule based and statistical learning based approaches to traditional neural methods, such as CNN, RNN, and further advanced to modern neural methods epitomized by Transformer models. In early stage, the rule based approaches infer the overall sarcasm polarity based on the refined sarcasm rules, such as the occurrence of the interjection word (Zhang et al., 2023a). Statistical learning based approaches mainly employ statistical learning techniques, e.g., SVM, RF, NB, etc., to extract patterns and relationships within the data (Zhou et al., 2023).

As deep learning based architectures have shown the superiority over statistical learning, numerous base neural networks, e.g., such as CNN (Jain et al., 2020), LSTM (Ghosh et al., 2018), GCN (Liang et al., 2022), etc., have been predominantly utilized during the middle stage of sarcasm detection research, aiming to learn and extract complex features in an end-to-end fashion. As the field of deep learning continues to evolve, sarcasm detection research has stepped into the era of pre-trained language models (PLMs). An increasing number of researchers are designing sophisticated PLM architectures to serve as encoders for obtaining effective text representations. For example, Liu et al. (2022) proposed a dual-channel framework by modeling both literal and implied sentiments separately. They also constructed two conflict prompts to elicit PLMs to generate the sarcasm polarity (Liu et al., 2023b). Qiao et al. (2023) presented a mutual-enhanced incongruity learning network to take advantage of the underlying consistency between the two modules to boost the performance. Tian et al. (2023) proposed a dynamic routing Transformer network to activate different routing transformer modules for modeling the dynamic mechanism in sarcasm detection.

However, the above-mentioned works still focus on how to utilize PLMs to extract effective features, without leveraging the extraordinary context learn-

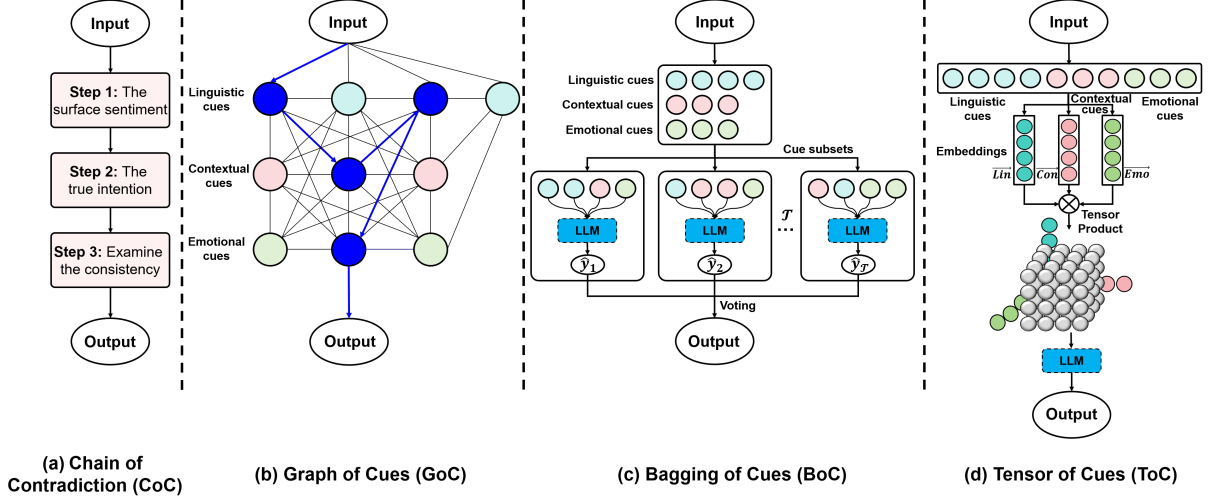


Figure 2: An illustration of our SarcasmCue framework that consists of four prompting sub-methods.

Table 1: Comparison of prompting methods.

Scheme	Seq?			Non-Seq?	
	Chain?	Tree?	Grap?	Set?	Tensor?
IO	✗	✗	✗	✗	✗
CoT	✓	✗	✗	✗	✗
ToT	✓	✓	✗	✗	✗
GoT	✓	✓	✓	✗	✗
SarcasmCue	✓	✓	✓	✓	✓

ing capabilities of LLMs. In contrast, this paper makes the first attempt to explore the potential of prompting LLMs in sarcasm detection.

3 The Proposed Framework: SarcasmCue

The overall schematic illustration of the proposed SarcasmCue framework is illustrated in Fig. 2. We qualitatively compare SarcasmCue to other prompting approaches in Tab. 1. SarcasmCue is the only one to fully support chain-based, tree-based, graph-based, set-based and multidimensional array-based reasoning. It is also the only one that simultaneously supports both sequential and non-sequential prompting methods.

3.1 Task Definition

Consider a sarcasm detection task. Given the data set $\mathcal{D} = \{(\mathcal{X}, \mathcal{Y})\}$, where $\mathcal{X} = \{x_1, x_2, \dots, x_n\}$ denotes the input text sequence and $\mathcal{Y} = \{y_1, y_2, \dots, y_n\}$ denotes the output label sequence. We use \mathcal{L}_θ to represent a large language model with parameter θ . Our task is to leverage a collection of cues $\mathcal{C} = \{c_1, c_2, \dots, c_k\}$ to bridge the input \mathcal{X}

and the output \mathcal{Y} , where each cue c_i is a coherent language sequence related to linguistics, context, or emotion that serves as an intermediate indicator toward identifying sarcasm.

3.2 Chain of Contradiction

We capture the inherent paradoxical nature of sarcasm, which is the incongruity between the surface sentiment and the true intention, and introduce *chain of contradiction*, a CoT-style paradigm that allows LLMs to decompose the problem of sarcasm detection into intermediate steps and solve each before making decision (Fig. 2 (a)). Each cue $c_k \sim \mathcal{L}_\theta^{CoC}(c_k | \mathcal{X}, c_1, c_2, \dots, c_{k-1})$ is sampled sequentially, then the output $\mathcal{Y} \sim \mathcal{L}_\theta^{CoC}(\mathcal{Y} | \mathcal{X}, c_1, \dots, c_k)$. A specific instantiation of CoC involves three steps:

Step 1. We first ask LLM to detect the surface sentiment via the following prompt p_1 :

Given the input sentence $[\mathcal{X}]$, what is the SURFACE sentiment, as indicated by clues such as keywords, sentimental phrases, emojis?

The output sequence $y_1 \sim \mathcal{L}_\theta^{CoC}(\mathcal{Y} | p_1)$ is generated from the language model \mathcal{L}_θ^{CoC} conditioned on input prompt p_1 .

Step 2. We then ask LLM to carefully discover the true intention via the following prompt p_2 :

Deduce what the sentence really means, namely the TRUE intention, by carefully checking any rhetorical devices, language style, unusual punctuation, common senses.

The output sequence, denoted as y_2 , is generated from the language model conditioned on prompt p_2 as well as the previous interaction p_1, y_1 , formu-

lated as $y_2 \sim \mathcal{L}_\theta^{CoC}(\mathcal{Y}|p_1, y_1, p_2)$.

Step 3. We finally ask LLM to examine the consistency between surface sentiment and true intention and make the final prediction:

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.

y_3 is therefore generated based on a joint understanding of the preceding context y_1, y_2 and p_1, p_2, p_3 : $y_3 \sim \mathcal{L}_\theta^{CoC}(\mathcal{Y}|p_1, y_1, p_2, y_2, p_3)$. The sarcasm label is identified from y_3 as the output of CoC.

Notably, CoC is built based on the presumption that all the cues are linearly correlated, and detects human sarcasm through step-by-step reasoning. Different from the original CoT, however, the steps are explicitly designed for the sarcasm detection context. Further details are presented in Algorithm 1 in App. A.

3.3 Graph of Cues

The linear structure of CoC restricts it to a single path of reasoning. To fill this gap, we introduce *graph of cues*, a GoT-style paradigm that allows LLMs to flexibly choose and weigh multiple cues, unconstrained by the need for unique predecessor nodes (Fig. 2 (b)). GoC frames the problem of sarcasm detection as a search over a graph, and is formulated as a tuple $(\mathcal{M}, \mathcal{G}, \mathcal{E})$, where \mathcal{M} is the cue maker used to define what are the common cues, \mathcal{G} is a graph of “sarcasm detection process”, \mathcal{E} is cue evaluator used to determine which cues to keep selecting and in which order. Unlike ToT and GoT, GoC does not involve the modules of “thought generator” and “thought aggregation”.

1. Cue maker. Human sarcasm judgment often relies on the combination and analysis of one or more cues to achieve an accurate understanding. Such cues can be broadly categorized into three types: linguistic cues, contextual cues and emotional cues. Linguistic cues refer to the linguistic features inherent in the text, including *keywords*, *rhetorical devices*, *punctuation* and *language style*. Contextual cues refer to the environment and background of the text, including *topic*, *cultural background*, *common knowledge*. Emotional cues denote the emotional stance conveyed by the text, including *emotional words*, *special symbols (such as emojis)* and *emotional contrasts*. A total number of $4+3+3=10$ cues are adopted.

2. Graph construction. In $\mathcal{G} = (V, E)$, the cues are regarded as vertices constituting the vertex set V , while the relations across cues form the edge set E . If there is an edge between cues c_k and c_j , it is considered that c_k and c_j are closely related. Given the cue c_k , the cue evaluator \mathcal{E} considers cue c_j to provide the most complementary information to c_k , which would combine with c_k to facilitate a deep understanding of sarcasm.

3. Cue evaluator. We involve \mathcal{G} in the LLM detecting sarcasm process. To advance this process, the cue evaluator \mathcal{E} assesses the current progress towards judging sarcasm by means of determining whether the cumulative cues obtained so far are sufficient to yield an accurate judgment. If so, the search goes to an end. Otherwise, it serves as a heuristic for the search algorithm, determining which additional cues to select and in what order, to further the detection process. Similar to ToT, an LLM is used as the cue evaluator \mathcal{E} .

We employ a voting strategy to determine the most valuable cue for selection, by explicitly comparing multiple potential cue candidates in a voting prompt, such as:

Given an input text \mathcal{X} , the target is to accurately detect sarcasm. Now, we have collected the keyword information as the first step: $\{\text{keywords}\}$, judge if this provides over 95% confidence for accurate detection. If so, output the result. Otherwise, from the remaining cues $\{\text{rhetorical devices, punctuation, ...}\}$, vote the most valuable one to improve accuracy and confidence for the next step.

This step can be formulated as $\mathcal{E}(\mathcal{L}_\theta^{GoC}, c_{j+1}) \sim \text{Vote}\{\mathcal{L}_\theta^{GoC}(c_{j+1}|\mathcal{X}, c_{1,2,...,j})\}_{c_{j+1} \in \{c_{j+1}, ..., c_k\}}$. In a nutshell, it greedily selects the most valuable cue until the final judgment is reached.

Although the GoC enables the exploration of many possible paths across the cue graph, its nature remains grounded in a step-by-step reasoning paradigm (see Algorithm 2 in App. A).

3.4 Bagging of Cues

We further relax the assumption that the cues for sarcasm detection are inter-related. We introduce *bagging of cues*, an ensemble learning based paradigm that allows LLMs to independently consider varied combinations of cues without assuming a fixed order or dependency among them (Fig. 2 (c)).

BoC constructs a pool of the pre-defined $k = 10$ cues \mathcal{C} . From this pool, \mathcal{T} subsets are random

sampled, each consisting of q (i.e., $1 \leq q \leq k$) cues. BoC thus leverages LLMs to generate \mathcal{T} independent sarcasm predictions \hat{y}_t based on the cues of each subset. Finally, such predictions are aggregated using a majority voting mechanism to produce the final sarcasm detection result. This approach embraces randomness in cue selection, enhancing the LLM’s ability to explore numerous potential paths, thus improving the robustness and accuracy of sarcasm detection. BoC consists of the following key steps:

Step 1. Cue subsets construction. A total of \mathcal{T} cue subsets $\mathcal{S}_{t \in [1, 2, \dots, \mathcal{T}]} = \{(c_{t_1}, c_{t_2}, \dots, c_{t_q}), t \in [1, 2, \dots, \mathcal{T}]\}$ are created by randomly sampling without replacement from the complete pool of cues \mathcal{C} . Each sampling is independent.

Step 2. LLM prediction. For each subset \mathcal{S}_t , an LLM \mathcal{L}_θ^{BoC} is used to independently make sarcasm prediction through the comprehensive analysis of the cues in the subset and the input text. This can be conceptually encapsulated as $\hat{y}_t \sim \mathcal{L}_\theta^{BoC}(\mathcal{Y}|\mathcal{S}_t, \mathcal{X})$.

Step 3. Prediction aggregation. These individual predictions are then combined using an aggregation function, i.e., majority voting, to yield the final prediction: $Y \sim Vote(\{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_\mathcal{T}\})$.

BoC treats all cues as independent and does not follow the step-by-step reasoning paradigm for sarcasm detection (see Algorithm 3 in App. A).

3.5 Tensor of Cues

CoC and GoC methods mainly handle low-order interactions between cues, while BoC assumes cues are independent. To capture high-order interactions among cues, we introduce *tensor of cues*, a novel paradigm that allows LLMs to amalgamate three types of cues (*viz.* linguistic, contextual and emotional cues) into a high-dimensional representation (Fig. 2 (d)).

ToC treats each type of cues as an independent, orthogonal view for sarcasm understanding, and constructs a multi-view representation through the tensor product of such three types of cues. We first ask the LLM to extract linguistic, contextual, and emotional cues respectively via a simple prompt. Taking linguistic cue extraction as an example:

Instruction: Please extract the linguistic cues from the input sentence for sarcasm detection, such as keywords, rhetorical devices, punctuation and language style. Input: $[\mathcal{X}]$

We take the outputs of the LLM’s final hidden layer as the embeddings of the linguistic, contextual and emotional cues, and apply a tensor fusion mechanism to fuse the cues as additional inputs to the sarcasm detection prompt. Inspired by the success of tensor fusion network (TFN) for multi-modal sentiment analysis (Zadeh et al., 2017), we apply token-wise tensor fusion to aggregate the cues. In particular, the embeddings are projected on a low-dimensional space, i.e., $\vec{Lin} = (e_1^l, e_2^l, \dots, e_L^l)^T$, $\vec{Con} = (e_1^c, e_2^c, \dots, e_L^c)^T$, $\vec{Emo} = (e_1^e, e_2^e, \dots, e_L^e)^T$. Suppose the LLM has a hidden dimensionality of d , fully-connected layers $f_{lin}, f_{con}, f_{emo}$ are constructed to map the embeddings to dimensionality of $\{d_l, d_c, d_e\}$, respectively for linguistic, contextual and emotional cues. Then, a tensor product is computed to combine the cues into a high-dimensional representation $\mathcal{Z} = (e_1, e_2, \dots, e_L)^T$, where

$$e_i = \begin{bmatrix} e_i^l \\ 1 \end{bmatrix} \otimes \begin{bmatrix} e_i^c \\ 1 \end{bmatrix} \otimes \begin{bmatrix} e_i^e \\ 1 \end{bmatrix}, \forall i \in [1, 2, \dots, L]. \quad (1)$$

The additional value of 1 facilitates an explicit rendering of single-cue features and bi-cue interactions, leading to a comprehensive fusion of different cues encapsulated in each fused token $e_i \in \mathcal{R}^{(d_l+1) \times (d_c+1) \times (d_e+1)}$. The values of d_l, d_c and d_e are delicately chosen such that the dimensionality of fused token is precisely d^1 . That enables an integration of the aggregated cues to the main prompt via:

Consider the information provided in the current cue above. Classify whether the input text is sarcastic or not. If you think the Input text is sarcastic, answer: yes. If you think the Input text is not sarcastic, answer: no. Input: $[\mathcal{X}]$

The embedded prompt above is **prepended** with the aggregated cue sequence \mathcal{Z} before fed to the LLM. As it is expected to output a single token of “yes” or “no” by design, we take the logit of the first generated token and decode the label accordingly as the output of ToC.

ToC facilitates deep interactions among these cues, providing a powerful and flexible framework for processing complex linguistic phenomena (see Algorithm 4 in App. A). Notably, as ToC manipulates cues on the vector level via neural structures, it requires access to the LLM structure and calls for supervised training on a collection of labeled samples. During training, the weights of the LLM are

¹Otherwise the fused tokens are truncated to d -dim vectors

Table 2: Dataset statistics.

Dataset	Avg. Length	#Train	#Dev	#Test
IAC-V1	68	1,595	80	320
IAC-V2	43	5,216	262	1,042
SemEval 2018	14	3,634	200	784
MUSTARD	14	552	-	138

frozen, and the linear weights in f_{lin} , f_{con} , f_{emo} are updated as an adaptation of LLM to the task context.

4 Experiments

4.1 Experiment Setups

Datasets. Four benchmarking datasets are selected as the experimental beds, *viz.* IAC-V1 (Lukin and Walker, 2013), IAC-V2 (Oraby et al., 2016), SemEval 2018 Task 3 (Van Hee et al., 2018) and MUSTARD (Castro et al., 2019).

IAC-V1 and **IAC-V2** are from the Internet Argument Corpus (IAC) (Lukin and Walker, 2013), specifically designed for the task of identifying and analyzing sarcastic remarks within online debates and discussions. It encompasses a balanced mixture of sarcastic and non-sarcastic comments.

SemEval 2018 Task 3 is collected using irony-related hashtags (i.e. #irony, #sarcasm, #not) and are subsequently manually annotated to minimise the amount of noise in the corpuses. It emphasize the challenges inherent in identifying sarcasm within the constraints of MUSTARD’s concise format, and highlight the importance of context and linguistic subtleties in recognizing sarcasm.

MUSTARD is compiled from popular TV shows including Friends, The Golden Girls, The Big Bang Theory, etc. It consists of 690 samples total of 3,000 utterances. Each sample is a conversation consisting of several utterances. In this work, we only use the textual information. The statistics for each dataset are shown in Table 2.

Baselines. A wide range of SOTA baselines are included for comparison. They are:

- **PLMs.** (1) **RoBERTa** (Liu et al., 2019), (2) **BNS-Net** (Zhou et al., 2023), (3) **DC-Net** (Liu et al., 2022), (4) **QUIET** (Liu et al., 2023a) and (5) **SarcPrompt** (Liu et al., 2023b) are five SOTA PLMs based approaches for sarcasm detection via pre-trained language modeling and refined representations.
- **Prompt tuning.** (6) **IO**, (7) **CoT** (Wei et al., 2022) and (8) **ToT** (Yao et al., 2024) are four

SOTA prompting approaches by leveraging advanced prompt approaches to enhance LLM’s performance.

- **LLMs.** (9) **GPT-4o**² and (10) **LLAMA 3-8B-Instruct**³ are the strongest general LLMs.

Implementation. We have implemented the prompting methods for **GPT-4o** and **LLaMA3-8B-Instruct**, and reported the performance of PLMs in their original papers. The GPT-4o methods are implemented with the official openAI Python API library⁴, while the LLaMA methods are implemented based on the Hugging Face Transformers library⁵. All prompting strategies are implemented for **GPT-4o** and **LLaMA3-8B-Instruct** except for ToC, which can solely be deployed on open-sourced LLMs. Following previous works in this field, LangChain⁶ is employed for the implementation of ToT and GoC. For the training of ToC, cross-entropy loss between the output logit and the true label token is computed to update the weights of the fully-connected layers.

4.2 Main Results

We report both **Accuracy** and **Macro-F1** results for **SarcasmCue** and baselines in a zero-shot setting in Table 3, except for ToC which requires supervised training for context adaption.

LLMs do not possess a unique advantage on sarcasm detection. Since sarcasm indicates the manifestation of sentiments and intentions opposite to the literal meaning of the texts, it usually violates logical reasoning pipelines that LLMs are known to excel at (Wei et al., 2022). This is empirically validated in the experiment where LLMs are observed to have consistently lower performance over PLMs in terms of average F1 scores across the four datasets. This highlights the need to investigate prompting strategies for adapting LLMs for sarcasm detection, towards which this work has made the first attempt and achieved preliminary success.

Human sarcasm detection does not necessarily follow a step-by-step reasoning process. The

²<https://openai.com/index/hello-gpt-4o/>

³<https://llama.meta.com/llama3/>

⁴<https://github.com/openai/openai-python>

⁵<https://huggingface.co/docs/transformers>

⁶<https://github.com/langchain-ai/langchain>

Table 3: Performance on four datasets. For LLMs, all strategies but ToC are based on a zero-shot setting.

Paradigm	Method	IAC-V1		IAC-V2		SemEval 2018		MUSTARD		Avg. of F1
		Acc.	Ma-F1	Acc.	Ma-F1	Acc.	Ma-F1	Acc.	Ma-F1	
PLMs	RoBERTa	72.10	72.90	82.70	82.70	73.90	72.80	66.27	65.16	73.39
	BNS-Net	66.13	65.95	75.93	75.92	73.47	73.44	-	-	71.77
	DC-Net	66.50	66.40	82.10	82.10	76.70	76.30	-	-	74.93
	QUIET	-	-	-	-	-	-	72.36	72.13	-
	SarcPrompt	75.20	75.20	84.90	84.90	76.90	76.60	66.58	66.63	75.78
GPT-4o	IO	70.63	70.05	73.03	71.99	64.03	63.17	67.24	65.79	68.14
	CoT	61.56	58.49	58.83	56.42	58.92	51.99	58.11	55.76	55.67
	ToT	71.56	71.17	70.63	69.07	63.90	63.02	69.00	68.27	67.46
	CoC (Ours)	72.19	71.52	73.36	72.31	70.79	70.60	69.42	68.48	70.73
	GoC (Ours)	85.38	68.08	64.97	61.30	74.03	74.02	70.69	69.91	68.33
	BoC (Ours)	68.75	67.36	71.35	69.39	62.12	61.85	69.42	68.45	66.79
LLaMA 3-8B-Instruct	IO	55.94	46.40	54.70	43.74	49.36	44.46	54.64	44.99	44.90
	CoT	56.25	47.28	54.22	42.96	49.36	44.55	54.20	44.86	44.91
	ToT	52.50	48.98	55.95	53.05	50.64	48.63	54.35	50.56	50.31
	CoC (Ours)	56.25	46.95	54.03	42.6	49.23	44.36	54.93	45.66	44.89
	GoC (Ours)	57.10	54.96	42.20	41.61	57.33	57.24	52.77	52.67	51.62
	BoC (Ours)	62.50	59.28	62.57	58.11	59.82	58.40	59.71	56.70	58.12
	ToC (Ours)	70.31	70.29	79.08	79.07	77.93	76.86	73.33	72.85	74.77

comparison between sequential (CoT, CoC, GoC, ToT) and non-sequential (BoC, ToC) prompting strategies fails to provide clear empirical evidences on whether sarcasm detection follows a step-by-step reasoning process. Nevertheless, the results on **LLaMA3-8B-Instruct** are more indicative to **GPT-4o**, since the latter has a strong capacity on its own (IO) and does not significantly benefit from any prompting strategies on its top. On **LLaMA3-8B-Instruct** where in-context learning is necessary for sarcasm detection due to its poor IO performance, non-sequential approaches can apparently offer more benefits over sequential ones, with a remarkable margin consistently present on all four datasets. This seems to support our hypothesize that sarcasm has a non-sequential nature.

SarcasmCue successfully adapts LLMs to sarcasm detection. The proposed prompting strategies in the **SarcasmCue** framework achieve an overall superior performance to the baseline prompting methods and bring about accuracy increase over the original LLMs in a zero-shot setting. In particular, by explicitly designing the reasoning steps for sarcasm detection, CoC beats CoT by a tremendous margin on GPT-4o, whilst performing in par with CoT on **LLaMA3-8B-Instruct**, an interesting result that further suggests the non-sequential nature of sarcasm detection. By pre-defining the set of cues on 3 main aspects, GoC

and BoC manage to guide LLMs to reason along the correct paths, leading to more accuracy judgment of sarcasm than the freestyle thinking in ToT. The proposed trainable neural architecture in ToC achieves an effective tensor fusion of multi-aspect cues for sarcasm detection, pushing the capacity to a comparable level to PLMs without tuning the LLM parameters.

5 Acknowledgments

This work is supported by National Science Foundation of China under grant No. 62006212, Fellowship from the China Postdoctoral Science Foundation (2023M733907), Natural Science Foundation of Hunan Province of China (242300421412).

6 Conclusion

In this work, we aim to study the step-wise reasoning nature of sarcasm detection, and introduce a prompting framework (called SarcasmCue) containing four sub-methods, *viz.* chain of contradiction (CoC), graph of cues (GoC), bagging of cues (BoC) and tensor of cues (ToC). It elicits LLMs for human sarcasm detection by considering sequential and non-sequential prompting methods. Our comprehensive evaluations across multiple benchmarks and state-of-the-art LLMs demonstrate that SarcasmCue outperforms traditional methods, with non-sequential prompting methods (GoC and ToC) showing particularly strong performance. In the

future, we plan to develop the multi-modal version of SarcasmCue for multi-modal sarcasm detection.

7 Limitations

The proposed SarcasmCue model has several limitations: (1) It incorporates only three types of cues – linguistic, contextual, and emotional – while other potentially useful cues, such as multimodal information, have not been integrated, potentially limiting the model’s comprehensive understanding of sarcasm; (2) the performance of SarcasmCue is influenced by the capabilities of the underlying large language models (LLMs), meaning it performs better with more powerful LLMs.

References

- Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 17682–17690.
- Santiago Castro, Devamanyu Hazarika, Verónica Pérez-Rosas, Roger Zimmermann, Rada Mihalcea, and Soujanya Poria. 2019. Towards multimodal sarcasm detection (an _obviously_ perfect paper). In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Florence, Italy. Association for Computational Linguistics.
- Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, et al. 2024. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 958–979.
- Shizhe Diao, Pengcheng Wang, Yong Lin, and Tong Zhang. 2023. Active prompting with chain-of-thought for large language models. *arXiv preprint arXiv:2302.12246*.
- Hao Fei, Bobo Li, Qian Liu, Lidong Bing, Fei Li, and Tat-Seng Chua. 2023. Reasoning implicit sentiment with chain-of-thought prompting. *arXiv preprint arXiv:2305.11255*.
- Debanjan Ghosh, Alexander R Fabbri, and Smaranda Muresan. 2018. Sarcasm analysis using conversation context. *Computational Linguistics*, 44(4):755–792.
- 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*, 91:106198.
- Jia Li, Ge Li, Yongmin Li, and Zhi Jin. 2023. Structured chain-of-thought prompting for code generation. *arXiv preprint arXiv:2305.06599*.
- 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)*, volume 1, pages 1767–1777. Association for Computational Linguistics.
- Yaochen Liu, Yazhou Zhang, and Dawei Song. 2023a. A quantum probability driven framework for joint multi-modal sarcasm, sentiment and emotion analysis. *IEEE Transactions on Affective Computing*.
- 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 preprint arXiv:1907.11692*.
- Yiyi Liu, Yequan Wang, Aixin Sun, Xuying Meng, Jing Li, and Jiafeng Guo. 2022. [A dual-channel framework for sarcasm recognition by detecting sentiment conflict](#). In *Findings of the Association for Computational Linguistics: NAACL 2022*, pages 1670–1680, Seattle, United States. Association for Computational Linguistics.
- Yiyi Liu, Ruqing Zhang, Yixing Fan, Jiafeng Guo, and Xueqi Cheng. 2023b. Prompt tuning with contradictory intentions for sarcasm recognition. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, pages 328–339.
- Stephanie Lukin and Marilyn Walker. 2013. [Really? well. apparently bootstrapping improves](#)

- the performance of sarcasm and nastiness classifiers for online dialogue. In *Proceedings of the Workshop on Language Analysis in Social Media*, pages 30–40, Atlanta, Georgia. Association for Computational Linguistics.
- Shereen Oraby, Vrindavan Harrison, Lena Reed, Ernesto Hernandez, Ellen Riloff, and Marilyn Walker. 2016. [Creating and characterizing a diverse corpus of sarcasm in dialogue](#). In *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, pages 31–41, Los Angeles. Association for Computational Linguistics.
- Yang Qiao, Liqiang Jing, Xuemeng Song, Xiaolin Chen, Lei Zhu, and Liqiang Nie. 2023. Mutual-enhanced incongruity learning network for multimodal sarcasm detection. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 9507–9515.
- Yuan Tian, Nan Xu, Ruike Zhang, and Wenji Mao. 2023. [Dynamic routing transformer network for multimodal sarcasm detection](#). In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 2468–2480, Toronto, Canada. Association for Computational Linguistics.
- 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*, pages 39–50, New Orleans, Louisiana. Association for Computational Linguistics.
- Bailin Wang, Zi Wang, Xuezhi Wang, Yuan Cao, Rif A Saurous, and Yoon Kim. 2024. Grammar prompting for domain-specific language generation with large language models. *Advances in Neural Information Processing Systems*, 36.
- 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.
- Changsong Wen, Guoli Jia, and Jufeng Yang. 2023. Dip: Dual incongruity perceiving network for sarcasm detection. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 2540–2550.
- Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.
- Tan Yue, Rui Mao, Heng Wang, Zonghai Hu, and Erik Cambria. 2023. Knowlenet: Knowledge fusion network for multimodal sarcasm detection. *Information Fusion*, 100:101921.
- Amir Zadeh, Minghai Chen, Soujanya Poria, Erik Cambria, and Louis-Philippe Morency. 2017. [Tensor fusion network for multimodal sentiment analysis](#). In *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, pages 1103–1114, Copenhagen, Denmark. Association for Computational Linguistics.
- Yazhou Zhang, Dan Ma, Prayag Tiwari, Chen Zhang, Mehedi Masud, Mohammad Shorfuz-zaman, and Dawei Song. 2023a. Stance-level sarcasm detection with bert and stance-centered graph attention networks. *ACM Transactions on Internet Technology*, 23(2):1–21.
- Yazhou Zhang, Yang Yu, Qing Guo, Benyou Wang, Dongming Zhao, Sagar Uprety, Dawei Song, Qiuchi Li, and Jing Qin. 2024. Cmma: Benchmarking multi-affection detection in chinese multimodal conversations. *Advances in Neural Information Processing Systems*, 36.
- Yazhou Zhang, Yang Yu, Dongming Zhao, Zuhe Li, Bo Wang, Yuexian Hou, Prayag Tiwari, and Jing Qin. 2023b. Learning multi-task commonness and uniqueness for multi-modal sarcasm detection and sentiment analysis in conversation. *IEEE Transactions on Artificial Intelligence*.
- Zhuosheng Zhang, Aston Zhang, Mu Li, and Alex Smola. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.
- Ge Zheng, Bin Yang, Jiajin Tang, Hong-Yu Zhou, and Sibe Yang. 2023. Ddcot: Duty-distinct chain-of-thought prompting for multimodal reasoning in language models. *Advances in Neural Information Processing Systems*, 36:5168–5191.
- Liming Zhou, Xiaowei Xu, and Xiaodong Wang. 2023. Bns-net: A dual-channel sarcasm

detection method considering behavior-level
and sentence-level conflicts. *arXiv preprint*
arXiv:2309.03658.

A Algorithms of Four Prompting Methods

1. **CoC.** We present further details of CoC in Algorithm 1.

Algorithm 1 Chain of contradiction

Require:

1: **Input:** Sentence \mathcal{X} , an LLM \mathcal{L}_θ

Ensure:

2: **Output:** Sarcasm Label \mathcal{Y}

3: **Step 1:** Detect surface sentiment

4: Output cue c_1 : $c_1 \sim \mathcal{L}_\theta^{CoC}(c_1|\mathcal{X}, p_1)$

5: **Step 2:** Discover true intention

6: Output cue c_2 : $c_2 \sim \mathcal{L}_\theta^{CoC}(c_2|\mathcal{X}, c_1, p_2)$

7: **Step 3:** Evaluate consistency and make prediction

8: Output cue c_3 : $c_3 \sim \mathcal{L}_\theta^{CoC}(c_3|\mathcal{X}, c_1, c_2, p_3)$

9: $\mathcal{Y} = \begin{cases} \text{Sarcastic} & \text{if } c_1 \neq c_2 \\ \text{Not Sarcastic} & \text{otherwise} \end{cases}$

10: **return** \mathcal{Y}

2. **GoC.** We present further details of GoC in Algorithm 2.

Algorithm 2 Graph of Cues (GoC) for Sarcasm Detection

Require:

1: **Input:** Sentence \mathcal{X} , an LLM \mathcal{L}_θ

Ensure:

2: **Output:** Sarcasm Label \mathcal{Y}

3: **1. Graph Construction**

4: Construct graph $\mathcal{G} = (V, E)$ where 10 cues are vertices V and relationships between cues are edges E

5: **2. Sarcasm Detection Process**

6: Initialize selected cues $C_{\text{selected}} = \emptyset, j = 0$

7: Initialize current confidence $\mathbb{C} = 0$

8: **while** $\mathbb{C} < 0.95 \cap j \leq 10$ **do**

9: Select the most valuable cue:

10: $c_{j+1} \sim \text{Vote} \{ \mathcal{L}_\theta^{GoC}(c_{j+1}|\mathcal{X}, c_1, c_2, \dots, c_j) \}_{c_{j+1} \in \{c_{j+1}, \dots, c_{10}\}}$

11: Add c_{j+1} to C_{selected}

12: Update current confidence $\mathbb{C}, j++$

13: Make final judgment based on C_{selected} : $\mathcal{Y} = \mathcal{L}_\theta^{GoC}(\mathcal{Y}|\mathcal{X}, C_{\text{selected}})$

14: **return** \mathcal{Y}

3. **BoC.** We present further details of BoC in Algorithm 3.

4. **ToC.** We present further details of ToC in Algorithm 4.

Algorithm 3 Bagging of cues

Require:

- 1: **Input:** Sentence \mathcal{X} , Cue Pool \mathcal{C} , Number of Subsets \mathcal{T} , Number of Cues per Subset q , an LLM \mathcal{L}_θ

Ensure:

- 2: **Output:** Sarcasm Label Y

- 3: **Step 1: Cue Subsets Construction**

- 4: **for** $t = 1$ **to** \mathcal{T} **do**

- 5: Randomly sample a subset $\mathcal{S}_t = \{c_{t1}, c_{t2}, \dots, c_{tq}\}$ from \mathcal{C}

- 6: **Step 2: LLM Prediction**

- 7: **for** $t = 1$ **to** \mathcal{T} **do**

- 8: Generate sarcasm prediction $\hat{y}_t \sim \mathcal{L}_\theta^{BoC}(\hat{y}_t | \mathcal{S}_t, \mathcal{X})$

- 9: **Step 3: Prediction Aggregation**

- 10: Aggregate predictions using majority voting:

- 11: $Y \sim \text{Vote}(\{\hat{y}_1, \hat{y}_2, \dots, \hat{y}_\mathcal{T}\})$

- 12: **return** Y
-

Algorithm 4 Tensor of cues

Require:

- 1: **Input:** Sentence \mathcal{X} , an LLM \mathcal{L}_θ

Ensure:

- 2: **Output:** Sarcasm Label \mathcal{Y}

- 3: **Step 1: Extract Cues**

- 4: Obtain linguistic cue embeddings $\vec{Lin} = (e_1^l, e_2^l, \dots, e_m^l)^T$, contextual cue embeddings $\vec{Con} = (e_1^c, e_2^c, \dots, e_p^c)^T$, emotional cue embeddings $\vec{Emo} = (e_1^e, e_2^e, \dots, e_s^e)^T$

- 5: **Step 2: Construct Tensor Representation**

- 6: Compute tensor product to combine cues: $\mathcal{Z} = \vec{Lin} \otimes \vec{Con} \otimes \vec{Emo}$

- 7: **Step 3: Sarcasm Detection**

- 8: Take tensor \mathcal{Z} as input to a LLM for sarcasm detection:

- 9: $\mathcal{Y} \sim \mathcal{L}_\theta^{ToC}(\mathcal{Y} | \mathcal{Z}, \mathcal{X})$

- 10: **return** \mathcal{Y}
-