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

Ben Yao[1], **Yazhou Zhang**[2,3*], **Qiuchi Li**[1*], **Jing Qin**[3]
[1]University of Copenhagen
[2]Tianjin University
[3]The Hong Kong Polytechnic University
yzhou_zhang@tju.edu.cn, qiuchi.li@di.ku.dk

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, in a way that does not necessarily follow a step-by-step fashion. To verify the validity of this argument, we introduce a new 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), which elicits LLMs to detect human sarcasm by considering sequential and non-sequential prompting methods. Through a comprehensive empirical comparison on four benchmarks, we highlight three key findings: (1) CoC and GoC show superior performance with more advanced models like GPT-4 and Claude 3.5, with an improvement of 3.5%. (2) ToC significantly outperforms other methods when smaller LLMs are evaluated, boosting the F1 score by 29.7% over the best baseline. (3) Our proposed framework consistently pushes the state-of-the-art (i.e., ToT) by 4.2%, 2.0%, 29.7%, and 58.2% in F1 scores across four datasets. This demonstrates the effectiveness and stability of the proposed framework.

Code —
https://github.com/qiuchili/llm_sarcasm_detection.git

Introduction

Recent large language models have demonstrated impressive performance across 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-steps 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 popular paradigm is to decompose complex problems into a series of intermediate solution steps, 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, due to its inherent ambivalence and figurative nature, sarcasm detection is often considered a holistic and irrational cognitive process that does not conform to step-by-step logical reasoning. The main reasons are two fold: (1) sarcasm expression does not strictly conform to formal logical structures, such as the law of hypothetical syllogism (i.e.,*if* $\mathcal{A} \Rightarrow \mathcal{B}$ *and* $\mathcal{B} \Rightarrow \mathcal{C}$, *then* $\mathcal{A} \Rightarrow \mathcal{C}$). 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 often considered a fluid combination of various cues. Each cue holds equal importance, and there is no rigid sequence of steps among them. 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 paradigms. 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)*. Each method has its own focus and advantages. In this work, *cue* is similar to *thought*, being a coherent language sequence related to linguistics, context, or emotion that serves as an intermediate indicator for identifying sarcasm, such as rhetorical devices or emotional words. More specifically,

- **CoC.** It harnesses the quintessential property of sarcasm (the contradiction between surface sentiment and true intention). It aims to: (1) identify the surface sentiment by extracting keywords, etc.; (2) deduce the true intention by scrutinizing rhetorical devices, etc.; and (3) determine the inconsistency between them. It is 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 as nodes, with the relations across cues represented as edges. Unlike CoC and ToT, it goes beyond following a fixed hierarchy or linear reasoning path. Like CoC, GoC follow a step-by-step reasoning process.
- **BoC.** BoC is a bagging approach that constructs a pool

*Corresponding authors.

of diverse cues and randomly samples multiple cue subsets. LLMs are employed to generate multiple predictions based on these subsets, and such predictions are aggregated to produce the final result. It is 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 tensor product. It allows language models to leverage higher-order interactions among the cues. ToC can be visualized as a 3D volumetric structure. Like BoC, ToC views sarcasm detection as a non-step-by-step reasoning process.

These four methods evolve from linear to nonlinear structure, and from a single perspective to multiple perspectives. They together form a comprehensive theoretical framework (SarcasmCue). The diverse design of the methods makes our framework adaptive to various sarcasm detection scenarios.

We present empirical evaluations of the proposed prompting approaches across four benchmarks over 4 SOTA LLMs (i.e., GPT-4o, Claude 3.5 Sonnet, Llama 3-8B, Qwen 2-7B), and compare their results against 3 SOTA prompting approaches (i.e., standard IO prompting, CoT and ToT). Three key observations are highlighted: (1) When the base model is more advanced (such as GPT-4 and Claude 3.5 Sonnet), CoC and GoC show superior performance against the state-of-the-art (SoTA) baseline with an improvement of 3.5% ↑. (2) ToC achieves the best performance when smaller LLMs are evaluated. For example, in Llama 3-8B, ToC's average F1 score of 65.24 represents a 29.7% improvement over the best baseline method, ToT. (3) Our proposed framework consistently pushes SoTA by 4.2%, 2.0%, 29.7% and 58.2% in F1 scores across four datasets. This demonstrates the effectiveness of the proposed framework. The main contributions are concluded as follows:

- Our work is the first to investigate the stepwise reasoning nature of sarcasm detection by using both sequential and non-sequential prompting methods.
- We propose a new prompting framework that consists of four sub-methods, viz. CoC, GoC, BoC and ToC.
- Comprehensive experiments over four datasets demonstrate the superiority of the proposed prompting framework.

Related Work

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. 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. However, its performance hinged on the quality of manually crafted prompts. To fill this gap, Auto-CoT was proposed to automatically construct demonstrations with questions and reasoning chains (Zhang et al. 2022). Furthermore, Yao et al. (2024) introduced a non-chain prompting framework, namely ToT, which made LLMs consider multiple different reasoning paths to decide the next course of action. 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.

Scheme	Seq?			Non-Seq?	
	Chain?	Tree?	Graph?	Set?	Tensor?
IO	✗	✗	✗	✗	✗
CoT	☑	✗	✗	✗	✗
ToT	☑	☑	✗	✗	✗
GoT	☑	☑	☑	✗	✗
SarcasmCue	☑	☑	☑	☑	☑

Table 1: Comparison of prompting methods.

However, 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 an urgent need to develop non-sequential prompting approaches.

Sarcasm Detection

Sarcasm detection has evolved from early statistical learning based approaches to traditional neural methods, and further advanced to modern neural methods epitomized by Transformer models. In early stage, statistical learning based approaches mainly employ statistical learning techniques, e.g., SVM, NB, etc., to extract patterns and relationships within the data (Zhang et al. 2023). As deep learning based architectures have shown the superiority, numerous base neural networks, e.g., such as CNN (Jain, Kumar, and Garg 2020), LSTM (Ghosh, Fabbri, and Muresan 2018), GCN (Liang et al. 2022), etc., have been predominantly utilized during the middle stage of sarcasm detection research. Now, 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 (Liu et al. 2023).

Different from them, we propose four prompting methods to make the first attempt to explore the potential of prompting LLMs in sarcasm detection.

The Proposed Framework: SarcasmCue

The proposed SarcasmCue framework is illustrated in Fig. 1. We qualitatively compare SarcasmCue with 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.

Task Definition

Given the data set $\mathcal{D} = \{(\mathcal{X}, \mathcal{Y})\}$, where $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$ denotes the input text sequence and $\mathcal{Y} =$

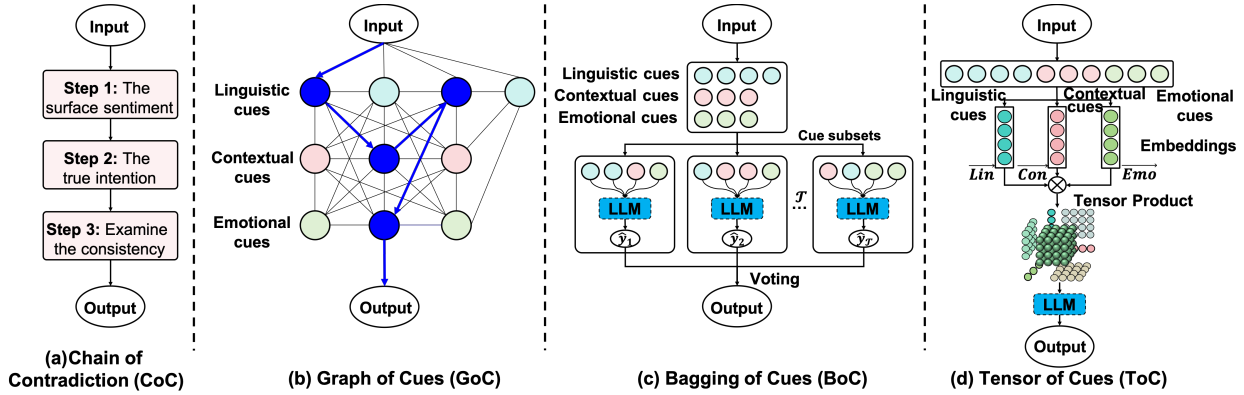

Figure 1: An illustration of our SarcasmCue framework that consists of four prompting sub-methods.

$\{y_1, y_2, \ldots, y_n\}$ denotes the output label sequence. We use $\mathcal{L}_\theta$ to represent a large language model with parameter θ. Our task is to leverage a collection of cues $\mathcal{C} = \{c_1, c_2, ..., c_k\}$ to brige the input $\mathcal{X}$ and the output $\mathcal{Y}$, where each cue c_i is a coherent language sequence that serves as an intermediate indicator toward identifying sarcasm.

Chain of Contradiction

We capture the inherent paradoxical nature of sarcasm, which is the incongruity between the surface sentiment and the true intention, and propose *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. 1 (a)). Each cue $c_k \sim \mathcal{L}_\theta^{CoC}(c_k|\mathcal{X}, c_1, c_2, ..., c_{k-1})$ is sampled sequentially, then the output $\mathcal{Y} \sim \mathcal{L}_\theta^{CoC}(\mathcal{Y}|\mathcal{X}, c_1, ..., c_k)$. A specific instantiation of CoC involves three steps:

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?

c_1 is the output sequence, which can be formulated as $c_1 \sim \mathcal{L}_\theta^{CoC}(c_1|\mathcal{X}, p_1)$.

2. We thus 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 punctuations, common senses.

c_2 is the output sequence, which can be formulated as $c_2 \sim \mathcal{L}_\theta^{CoC}(c_2|\mathcal{X}, c_1, p_2)$.

3. Let LLM 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.

Compared to CoT which prompts LLM to reason step-by-step in an open way, our CoC strategy provides specifically designed instructions for each step. Still, it presumes that the cues are linearly correlated, and detects human sarcasm through step-by-step reasoning.

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 graph based paradigm that allows LLMs to flexibly choose and weigh multiple cues, unconstrained by the need for unique predecessor nodes (Fig. 1 (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.

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 emotions implied in the text, including *emotional words*, *special symbols (such as emojis)* and *emotional contrasts*. Hence, GoC can obtain *4+3+3=10* cues.

2. Graph construction. In $\mathcal{G} = (V, E)$, 10 cues are regarded as vertices, constituting the vertex set V, the supplement relations across cues are regarded as edges. 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 associate $\mathcal{G}$ with LLM detecting sarcasm process. To advance this process, the cue evaluator $\mathcal{E}$ assesses the current progress by asking the LLM whether the cumulative cues obtained thus far are sufficient to yield an accurate judgment. The search goes to an end if a positive answer is returned; otherwise, the detection process proceeds by instructing the LLM to determine which additional cues to select and in what order. In this work, an LLM will act as the cue evaluator, similar to ToT.

We employ a voting strategy to determine the most valuable cue for selection, by deliberately 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: {*keywords*}, judge if this provides over 95% confidence for accurate detection. If so, output the result. Otherwise, from the remaining cues {*rhetorical devices*, *punctuation*, ...}, vote the most valuable one to improve accuracy and confidence for the next step.

This step can be formulated as $\mathcal{E}\left(\mathcal{L}_\theta^{GoC}, c_{j+1}\right) \sim Vote\left\{\mathcal{L}_\theta^{GoC}\left(c_{j+1}|\mathcal{X}, c_{1,2,...,j}\right)\right\}_{c_{j+1}\in\{c_{j+1},...,c_k\}}$. Until the final judgment is reached, the most valuable cue are always selected in a greedy fashion. Although GoC enables the exploration of many possible paths across the cue graph, its nature remains grounded in a step-by-step reasoning paradigm.

Bagging of Cues

We relax the assumption that the cues are interrelated in detecting sarcasm. We introduce *bagging of cues*, a ensemble learning based paradigm that allows LLMs to independently consider varied combinations of cues without assuming a fixed order or dependency among them (Fig. 1 (c)).

BoC constructs a pool of the pre-defined 10 cues $\mathcal{C}$. From this pool, $\mathcal{T}$ subsets are obtained through $\mathcal{T}$ random samplings, where each subset $\mathcal{S}_t$ consists of q $(i.e., 1 \leq q \leq 10)$ 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 result. This approach embraces randomness in cue selection, enhancing the LLM's ability to explore numerous potential paths. BoC consists of three key steps:

1. Cue subsets construction. A total of $\mathcal{T}$ cue subsets $\mathcal{S}_{t\in[1,2,...,\mathcal{T}]} = \{c_{t1}, c_{t2}, ..., c_{tq}\}$ are created by randomly sampling without replacement from the complete pool of cues $\mathcal{C}$. Each sampling is independent.

2. LLM prediction. For each subset $\mathcal{S}_t$, a LLM $\mathcal{L}_\theta^{BoC}$ 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 $\hat{y}_t \sim \mathcal{L}_\theta^{BoC}\left(\hat{y}_t|\mathcal{S}_t, \mathcal{X}\right)$.

3. Prediction aggregation. The predictions $\{\hat{y}_1, \hat{y}_2, ..., \hat{y}_\mathcal{T}\}$ are combined using majority voting to yield the final prediction Y.

BoC does not follow the step-by-step reasoning paradigm for sarcasm detection.

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 stereo paradigm that allows LLMs to amalgamate three types of cues ($viz.$ linguistic, contextual and emotional cues) into a high-dimensional representation. (Fig. 1 (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. For example:

> Extract the linguistic cues from the input sentence for sarcasm detection, such as keywords, rhetorical devices, punctuation and language style.

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 multimodal 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 via the fully-connected layers, i.e., $\vec{Lin} = \left(e_1^l, e_2^l, ..., e_L^l\right)^T$, $\vec{Con} = \left(e_1^c, e_2^c, ..., e_L^c\right)^T$, $\vec{Emo} = \left(e_1^e, e_2^e, ..., e_L^e\right)^T$. Then, a tensor product is computed to combine the cues into a high-dimensional representation $\mathcal{Z} = \left(e_1, e_2, ..., e_L\right)^T$, where

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

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.

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

Experiments

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, Lefever, and Hoste 2018) and MUStARD (Castro et al. 2019).

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

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

Paradigm	Method	IAC-V1		IAC-V2		SemEval 2018		MUStARD		Avg. of F1
		Acc.	Ma-F1	Acc.	Ma-F1	Acc.	Ma-F1	Acc.	Ma-F1	
GPT-4o	IO	70.63	70.05	73.03	71.99	64.03	63.17	67.24	65.79	67.75
	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	67.88
	CoC (Ours)	**72.19**	**71.52**	**73.36**	**72.31**	70.79	70.60	69.42	68.48	**70.73**♣
	GoC (Ours)	65.00	62.91	64.97	61.30	**74.03**♣	**74.02**♣	**70.69**♣	**69.91**♣	67.04
	BoC (Ours)	68.75	67.36	71.35	69.39	62.12	61.85	69.42	68.45	66.76
Claude 3.5 Sonnet	IO	66.56	66.54	**76.78**	**76.62**	75.13	75.11	**74.78**	**74.78**	73.26
	CoT	**71.25**	**71.14**	74.66	74.10	71.56	71.47	73.62	73.53	72.56
	ToT	63.44	62.48	71.88	71.74	68.62	68.61	58.84	54.46	64.32
	CoC (Ours)	69.69	69.40	73.22	73.17	**82.27**♣	**82.23**♣	74.20	74.16	**74.74**♣
	GoC (Ours)	70.94	70.93	74.67	74.18	76.91	76.91	70.00	69.85	72.97
	BoC (Ours)	66.88	66.40	73.61	72.82	70.28	70.07	72.61	71.93	70.31
Llama 3-8B	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	42.60	49.23	44.36	54.93	45.66	44.89
	GoC (Ours)	57.10	54.96	54.22	53.30	57.33	57.24	52.77	52.67	54.54
	BoC (Ours)	**62.50**	59.28	62.57	58.11	65.94	65.50	59.71	56.70	59.90
	ToC (Ours)	62.19	**61.78**♣	**72.95**♣	**72.94**♣	**68.88**♣	**68.21**♣	**61.26**♣	**58.03**♣	**65.24**♣
Qwen 2-7B	IO	56.56	49.32	51.82	38.57	45.15	38.83	54.78	46.17	43.22
	CoT	54.69	46.53	52.88	40.12	43.24	35.79	54.93	45.81	42.06
	ToT	53.44	43.71	50.29	39.62	44.26	38.12	52.90	44.60	41.51
	CoC (Ours)	55.00	45.77	51.92	38.90	43.75	36.37	53.77	44.26	41.33
	GoC (Ours)	55.00	47.35	53.45	42.25	45.03	38.17	54.49	47.49	43.82
	BoC (Ours)	52.50	43.78	52.40	40.24	49.87	45.63	54.06	46.11	43.94
	ToC (Ours)	**71.56**♣	**71.56**♣	**72.33**	**71.76**♣	**68.88**♣	**68.77**♣	**65.94**♣	**61.46**♣	**68.39**♣

Table 2: Performance on four datasets. For LLMs, all strategies are based on a zero-shot setting. **Bold + underline** and underline indicate the best and second-best results for each dataset. ♣ represents significance improvement over the best baseline via unpaired t-test ($p < 0.05$).

- **Prompt tuning.** (1) **IO**, (2) **CoT** (Wei et al. 2022) and (3) **ToT** (Yao et al. 2024) are three SOTA prompting approaches by leveraging advanced prompt approaches to enhance LLM's performance.
- **LLMs.** We involve four general LLMs in the experiment, including (4) **GPT-4o**, (5) **Claude 3.5 Sonnet**, (6) **Llama 3-8B** and (7) **Qwen 2-7B** (Bai et al. 2023). The first two are non-open-source LLMs while the last two are open-source LLMs. All four LLMs are representative of the strongest capabilities of their kinds.

Implementation. We have implemented the prompting methods for **GPT-4o**, **Claude 3.5 Sonnet**, **Llama 3-8B** and **Qwen2-7B**. The GPT-4o and Claude 3.5 Sonnet methods are implemented with the respective official Python API library: openAI[2] and anthropic[3], while the LLaMA and Qwen methods are implemented based on the Hugging Face Transformers library[4]. For ToC, during training, the original LLM (LlaMA 3-8B and Qwen 2-7B) weights are frozen, while the projection layers are trainable (lr = 0.0001, epochs = 20).

[2]https://github.com/openai/openai-python

[3]https://github.com/anthropics/anthropic-sdk-python

[4]https://huggingface.co/docs/transformers

Main Results

We report both **Accuracy** and **Macro-F1** scores for **SarcasmCue** and baselines in Table 2.

(1) SarcasmCue consistently outperforms SoTA prompting baselines. The proposed prompting strategies in the **SarcasmCue** framework achieve an overall superior performance compared to the baselines and consistently push the SoTA by 4.2%, 2.0%, 29.7% and 58.2% on F1 scores across four datasets. In particular, by explicitly designing the reasoning steps for sarcasm detection, CoC beats CoT by a tremendous margin on **GPT-4o** and **Claude 3.5 Sonnet**, whilst performing in par with CoT on **Llama 3-8B** and **Qwen 2-7B**. By pre-defining the set of cues in three main categories, GoC and BoC effectively guide LLMs to reason along correct paths, leading to more accurate judgments of sarcasm compared to the freestyle thinking in ToT. For example, the best proposed method, CoC (74.74), brings a 2.0% improvement over the best baseline method, IO (73.26). ToC achieves an effective tensor fusion of multi-aspect cues for sarcasm detection, significantly outperforming other baselines. For instance, it exhibits a 29.7% improvement over the best baseline method, ToT (50.31).

Method	IAC-V1	IAC-V2	SemEval	MUStARD	Avg. of F1
w/o Lin	68.41	**75.62**	77.42	69.66	72.78
w/o Emo	69.65	74.04	**78.70**	**70.57**	**73.24**
w/o Con	70.53	74.91	76.39	70.11	72.99
GoC	**70.93**	74.18	76.91	69.85	72.97
w/o Lin	45.89	42.49	47.47	65.33	50.30
w/o Emo	58.00	56.99	56.81	68.84	60.16
w/o Con	61.71	63.70	69.53	74.80	67.44
BoC	**66.40**	**72.82**	**70.07**	**71.93**	**70.31**
w/o Lin	45.79	51.90	56.01	46.84	50.14
w/o Emo	48.60	49.40	52.38	45.12	48.88
w/o Con	52.51	**53.69**	52.14	48.28	51.66
GoC	**54.96**	53.30	**57.24**	**52.67**	**54.54**
w/o Lin	52.71	57.51	57.53	53.06	55.20
w/o Emo	57.33	59.40	62.01	53.06	57.95
w/o Con	56.88	**60.36**	59.04	52.30	57.15
BoC	**59.28**	58.11	**65.50**	**56.70**	**59.90**
w/o Lin	53.31	67.05	59.20	48.05	56.90
w/o Emo	57.42	67.08	64.01	52.89	60.35
w/o Con	55.26	71.78	63.93	52.48	60.86
ToC	**61.78**	**72.94**	**68.21**	**58.03**	**65.24**

Table 3: Ablation study of BoC, GoC and ToC. All strategies are run on a zero-shot setting. The top part shows results for **Claude 3.5 Sonnet**, and the bottom part for **Llama 3-8**. The best results for each dataset are formatted in **Bold + underline**.

(2) Sarcasm detection does not necessarily follow a step-by-step reasoning process. The 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 **Llama 3-8B** are more indicative to **GPT-4o** and **Claude 3.5 Sonnet**, since the latter models have strong capabilities on their own (IO) and do not significantly benefit from any prompting strategies. For **Llama 3-8B** and **Qwen 2-7B**, non-sequential methods, particularly ToC, show superior performance. In **Llama 3-8B**, ToC achieves an average F1 score of 65.24%, which is 8.9% higher than the best sequential method (GoC at 54.54%). The difference is even more pronounced on **Qwen 2-7B**. Furthermore, The McNemar's test between CoC and BoC on the **Llama 3-8B**'s outputs exhibits χ^2= 117.00, $p < 0.05$, suggesting that the BoC works significantly better than CoC. These result supports our hypothesize that sarcasm has a non-sequential nature.

Ablation Study

Table 3 presents the result of ablation study. *w/o Lin*, *w/o Emo*, *w/o Con* refer to the method where linguistic, emotional and contextual cues are ablated, respectively. To avoid proactive extraction of ablated cues by an LLM, we explicitly "prompt away" the cues in the inputs. An example prompt could be "You can only use the emotional cues and contextual cues, and do not use any linguistic information here" for the *w/o Lin* case.

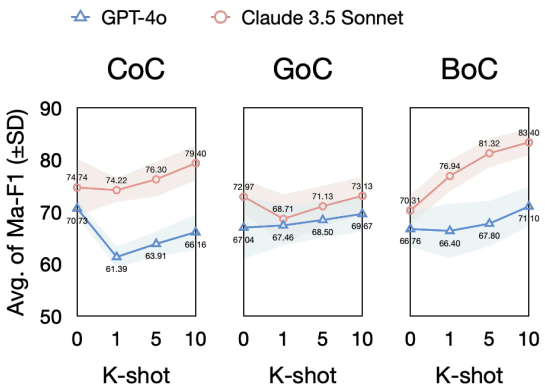

Figure 2: The average Macro-F1 across K-shots for the **GPT-4o** and **Claude 3.5 Sonnet** models.

The experiment results highlight the following conclusions: (a) the removal of any single type of cue leads to a noticeable drop in performance across all datasets, demonstrating the importance of each type of cue in sarcasm detection; (b) linguistic cues appear to have the most significant impact, as removing them leads to a noticeable decrease in performance across most settings; (c) the absence of contextual cues also affects the performance, but to a lesser extent compared to linguistic cues.

Zero-Shot V/S Few-Shot Prompting

Since the above experiments are mainly based on a zero-shot setting, we are curious of whether the conclusions also apply in a few-shot scenario. Therefore, we perform few-shot experiments to evaluate whether the proposed SarcasmCue framework can perform better when a limited number of contextual examples are available. We plot the main results in Fig. 2, we randomly sample $k = \{0, 1, 5, 10\}$ examples from the training set.

As shown in the plot, the number of demonstrations has a significant impact on the results. For example, CoC appears sensitive to the initial introduction of demonstration examples with a slight descent in performance when only 1 example is provided. However, as the number of shots increases to 5 and 10, the performance progressively improves. This trend underscores the effectiveness of CoC in adapting and refining its approach with more examples. In contrast, BoC demonstrates a consistent improvement in performance as the number of shots increases.

Overall, these results demonstrate the robustness and adaptability of the SarcasmCue framework in zero-shot and few-shot scenarios. The framework can effectively utilize limited contextual examples to further improve sarcasm detection, making it suitable for applications where large annotated datasets are not readily available.

Influences of LLM Scales

In an attempt to study the influence of different LLM scales, we evaluate the performance of sarcasm detection of **Qwen** and **Llama** of varying sizes, see Fig. 3.

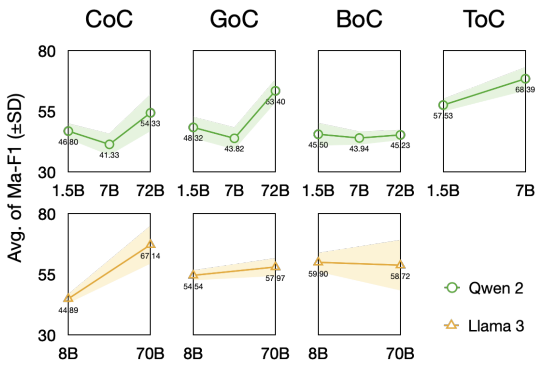

Figure 3: The influence of model scale. The figures in the top and bottom correspond to Qwen and Llama models, respectively.

The key take-aways are two-fold. First, the efficacy of our prompting methods is amplified with increasing model scale. This aligns closely with the key findings of the CoT method (Wei et al. 2022). This occurs because when an LLM is sufficiently large, its capabilities for multi-hop reasoning and understanding language are significantly enhanced. Second, ToC exhibits high sensitivity to model scale, performing significantly better in larger models, making it particularly suitable for larger-scale applications. CoC and GoC demonstrate moderate sensitivity, indicating a balance between performance improvement and scalability. BoC offers robust performance even in smaller models, suggesting its utility in resource-constrained scenarios. Overall, our proposed framework has a high adaptability across various model scales by offering suitable methods.

Error Analysis

Fig. 4 shows the error rates of failure cases in terms of false negative (FN) and false positive (FP) for all four prompting methods in SarcasmCue. CoC, GoC and BoC exhibit higher false positive rates, indicating an over-detection of sarcasm that could lead to the frequent misclassification of normal statements as sarcastic. In contrast, ToC exhibits the lowest overall error rate and the FP and FN rates are indeed much closer to each other, indicating a balanced performance in detecting both sarcastic and non-sarcastic texts. We further analyzed the common patterns among the over-detected examples, and found out that our methods are overly sensitive to certain cues commonly associated with sarcasm, such as negative information, exaggerated language, rhetorical devices, or harmful words. These insights highlight potential directions for future improvements in sarcasm detection methodologies. The higher false positive rates suggest a need for refining these methods to reduce over-sensitivity and improve discrimination between sarcastic and non-sarcastic texts.

Extension to New Task

To evaluation the generalization capability of SarcasmCue, we apply it to another complex affection understanding task, **humor detection**. We compare our proposed SarcasmCue

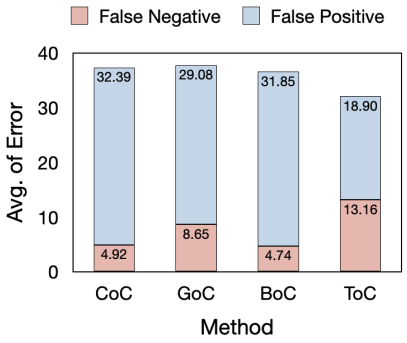

Figure 4: The average error rate of the four prompting methods.

Method	CMMA		UR-FUNNY-V2		Avg. of F1
	Acc.	Ma-F1	Acc.	Ma-F1	
MFN	-	-	64.44	64.12	-
SVM+BERT	55.23	54.08	**69.62**	**69.27**	61.68
CoC	78.14	**58.60**	64.08	60.13	65.24
GoC	79.60	57.42	64.89	61.65	65.89
BoC	**75.81**	58.58	68.71	66.83	**67.48**

Table 4: Performance on two humor detection datasets.

(where the backbone is GPT-4o) with two supervised PLMs (MFN (Hasan et al. 2021) and SVM+BERT (Zhang et al. 2024)) on two benchmarking datasets, CMMA (Zhang et al. 2024) and UR-FUNNY-V2 (Hasan et al. 2019).

As shown in Table 4, our methods (BoC and CoC) surpass the baseline on CMMA, whilst performing in par to the strongest baselines on the UR-FUNNY-V2 dataset. These results highlight the strong generalizability and versatility of our framework, confirming its potential utility across a wide range of affection understanding tasks.

Conclusions

This work aims to study the stepwise reasoning nature of sarcasm detection, and introduces a prompting framework (called SarcasmCue) containing four sub-methods, viz. CoC, GoC, BoC and ToC. It elicits LLMs to detect human sarcasm by considering sequential and non-sequential prompting methods. Our comprehensive evaluations across multiple benchmarks and SoTA LLMs demonstrate that SarcasmCue outperforms traditional methods and pushes the state-of-the-art by 4.2%, 2.0%, 29.7% and 58.2% F1 scores across four datasets. Additionally, the performance of SarcasmCue on humor detection further validate its robustness and versatility.

Limitations. First, the ToC method demands extra computational resources due to its complex multi-view tensor structure. Second, aside from BoC, the effectiveness of the other three approaches in the SarcasmCue framework is largely dependent on the scale of LLM. Finally, this framework primarily focuses on text data while sarcasm detection often requires multi-modal analysis.

Ethical Statement

We are committed to adhere to strict ethical standards, using open and fair datasets while recognizing the societal impact of sarcasm detection and promoting its responsible application.

Acknowledgments

This work is supported by a grant for Collaborative Research with World-leading Research Groups of The Hong Kong Polytechnic University (project no. G-SACF), Natural Science Foundation of Hunan Province of China (242300421412), Foundation of Key Laboratory of Dependable Service Computing in Cyber-Physical-Society (Ministry of Education), Chongqing University (PJ.No: CPS-DSC202103).

References

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; Fan, Y.; Ge, W.; Han, Y.; Huang, F.; Hui, B.; Ji, L.; Li, M.; Lin, J.; Lin, R.; Liu, D.; Liu, G.; Lu, C.; Lu, K.; Ma, J.; Men, R.; Ren, X.; Ren, X.; Tan, C.; Tan, S.; Tu, J.; Wang, P.; Wang, S.; Wang, W.; Wu, S.; Xu, B.; Xu, J.; Yang, A.; Yang, H.; Yang, J.; Yang, S.; Yao, Y.; Yu, B.; Yuan, H.; Yuan, Z.; Zhang, J.; Zhang, X.; Zhang, Y.; Zhang, Z.; Zhou, C.; Zhou, J.; Zhou, X.; and Zhu, T. 2023. Qwen Technical Report. *arXiv preprint arXiv:2309.16609*.

Besta, M.; Blach, N.; Kubicek, A.; Gerstenberger, R.; Podstawski, M.; Gianinazzi, L.; Gajda, J.; Lehmann, T.; Niewiadomski, H.; Nyczyk, P.; et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 17682–17690.

Castro, S.; Hazarika, D.; Pérez-Rosas, V.; Zimmermann, R.; Mihalcea, R.; and Poria, S. 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.

Cui, C.; Ma, Y.; Cao, X.; Ye, W.; Zhou, Y.; Liang, K.; Chen, J.; Lu, J.; Yang, Z.; Liao, K.-D.; 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*, 958–979.

Ghosh, D.; Fabbri, A. R.; and Muresan, S. 2018. Sarcasm analysis using conversation context. *Computational Linguistics*, 44(4): 755–792.

Hasan, M. K.; Lee, S.; Rahman, W.; Zadeh, A.; Mihalcea, R.; Morency, L.-P.; and Hoque, E. 2021. Humor Knowledge Enriched Transformer for Understanding Multimodal Humor. *Proceedings of the AAAI Conference on Artificial Intelligence*, 35(14): 12972–12980.

Hasan, M. K.; Rahman, W.; Zadeh, A.; Zhong, J.; Tanveer, M. I.; Morency, L.-P.; et al. 2019. UR-FUNNY: A multimodal language dataset for understanding humor. *arXiv preprint arXiv:1904.06618*.

Jain, D.; Kumar, A.; and Garg, G. 2020. Sarcasm detection in mash-up language using soft-attention based bi-directional LSTM and feature-rich CNN. *Applied Soft Computing*, 91: 106198.

Liang, B.; Lou, C.; Li, X.; Yang, M.; Gui, L.; He, Y.; Pei, W.; and Xu, R. 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, 1767–1777. Association for Computational Linguistics.

Liu, Y.; Zhang, R.; Fan, Y.; Guo, J.; and Cheng, X. 2023. Prompt Tuning with Contradictory Intentions for Sarcasm Recognition. In *Proceedings of the 17th Conference of the European Chapter of the Association for Computational Linguistics*, 328–339.

Lukin, S.; and Walker, M. 2013. Really? Well. Apparently Bootstrapping Improves the Performance of Sarcasm and Nastiness Classifiers for Online Dialogue. In Danescu-Niculescu-Mizil, C.; Farzindar, A.; Gamon, M.; Inkpen, D.; and Nagarajan, M., eds., *Proceedings of the Workshop on Language Analysis in Social Media*, 30–40. Atlanta, Georgia: Association for Computational Linguistics.

Oraby, S.; Harrison, V.; Reed, L.; Hernandez, E.; Riloff, E.; and Walker, M. 2016. Creating and Characterizing a Diverse Corpus of Sarcasm in Dialogue. In Fernandez, R.; Minker, W.; Carenini, G.; Higashinaka, R.; Artstein, R.; and Gainer, A., eds., *Proceedings of the 17th Annual Meeting of the Special Interest Group on Discourse and Dialogue*, 31–41. Los Angeles: Association for Computational Linguistics.

Van Hee, C.; Lefever, E.; and Hoste, V. 2018. SemEval-2018 Task 3: Irony Detection in English Tweets. In Apidianaki, M.; Mohammad, S. M.; May, J.; Shutova, E.; Bethard, S.; and Carpuat, M., eds., *Proceedings of the 12th International Workshop on Semantic Evaluation*, 39–50. New Orleans, Louisiana: Association for Computational Linguistics.

Wei, J.; Wang, X.; Schuurmans, D.; Bosma, M.; Xia, F.; Chi, E.; Le, Q. V.; Zhou, D.; et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35: 24824–24837.

Yao, S.; Yu, D.; Zhao, J.; Shafran, I.; Griffiths, T.; Cao, Y.; and Narasimhan, K. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36.

Zadeh, A.; Chen, M.; Poria, S.; Cambria, E.; and Morency, L.-P. 2017. Tensor Fusion Network for Multimodal Sentiment Analysis. In Palmer, M.; Hwa, R.; and Riedel, S., eds., *Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing*, 1103–1114. Copenhagen, Denmark: Association for Computational Linguistics.

Zhang, Y.; Ma, D.; Tiwari, P.; Zhang, C.; Masud, M.; Shorfuzzaman, M.; and Song, D. 2023. Stance-level sarcasm detection with bert and stance-centered graph attention networks. *ACM Transactions on Internet Technology*, 23(2): 1–21.

Zhang, Y.; Yu, Y.; Guo, Q.; Wang, B.; Zhao, D.; Uprety, S.; Song, D.; Li, Q.; and Qin, J. 2024. CMMA: Benchmarking

Multi-Affection Detection in Chinese Multi-Modal Conversations. *Advances in Neural Information Processing Systems*, 36.

Zhang, Z.; Zhang, A.; Li, M.; and Smola, A. 2022. Automatic chain of thought prompting in large language models. *arXiv preprint arXiv:2210.03493*.