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 is challenging in Natural Language Processing (NLP) due to subtle linguistic cues, sentiment incongruities, and contextual dependencies. This project extends *SarcasmBench* [19] by implementing **Chain of Contradiction (CoC)** prompting, a Chain of Thought (CoT) based framework designed for sarcasm detection. CoC explicitly models sentiment contradictions to improve performance. This study evaluates CoC in few-shot settings on multiple LLMs and compares its performance against zero-shot I/O, few-shot I/O, and few-shot CoT prompting strategies. Preliminary results show that CoC improves **recall** while slightly reducing **precision**, which sggests potential for sarcasm detection tasks.

1 INTRODUCTION

Sarcasm detection remains a major challenge in NLP due to subtle linguistic cues, sentiment congruence, and the need for contextual understanding. Zhang et al. introduced *SarcasmBench* [19], which evaluates 11 LLMs and 8 pre-trained language models (PLMs) across multiple datasets using zero-shot I/O, few-shot I/O, and few-shot CoT prompting strategies. However, CoT struggled with sarcasm's holistic nature. This project enhances the results from *SarcasmBench* by integrating **CoC prompting**, a reasoning framework that explicitly evaluates contradictions between surface sentiment and true intent [17].

Objectives:

- Improve sarcasm detection using CoC prompting.
- Compare CoC against zero-shot, few-shot, and CoT strategies across benchmarks.
- Demonstrate contradiction-based reasoning enhances sarcasm comprehension.

2 RELATED WORK

2.1 Large Language Models

LLMs, including GPT-4 [11] and LLaMA 3 [14], excel in NLP tasks through extensive training on large corpora. They have rapidly advanced in recent years, demonstrating exceptional capabilities in natural language understanding, in-context learning, and task generalization [2]. General purpose LLMs 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 [10]. Open-source models such as LLaMA 3 enable customization and more transparency for developers. However, proprietary models such as GPT-4 offer state-of-the-art performance at the cost of transparency.

2.2 Sarcasm Detection

Early sarcasm detection relied on rule-based and statistical models like SVM and Naive Bayes [18]. These approaches struggled with generalizability because of their dependence on hand-crafted features. Deep learning approaches, including CNNs [6], LSTMs [5], and Graph Convolutional Networks [7], improved feature extraction. Recent work employs pre-trained models such as BERT [3] and RoBERTa [8], which use contextual embeddings to improve classification accuracy.

2.3 CoT and CoC Prompting

CoT prompting enhances LLM reasoninbg by breaking tasks into sequential steps. The LLM is guided through sequential instructions $z_1, ..., z_n$, to discover some result y in a process shown by the equation below [19].

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

Wei et al. (2022) formally introduced CoT[16], but its success was highly dependent on high-quality prompts. CoC prompting explicitly models sarcasm as a contradiction between sentiment and true intent [17]. CoC better aligns with human sarcasm comprehension and has been shown to outperform standard CoT in sarcasm benchmarks.

3 METHODOLOGY

This project evaluates CoC prompting within *SarcasmBench*, comparing it against traditional prompting methods across multiple datasets and LLMs.

3.1 CoC Prompting Framework

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.

The CoC prompt construction is taken from Yao et al. (2024) [17], and is provided below. There are three separate prompts provided to the LLM as a conversation. [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."

3.2 Benchmark Datasets

Five of the original six sarcasm detection datasets from *SarcasmBench* are utilized. Any augmentations to each dataset with respect to the original experiment is mentioned in their descriptions, but due to size similarities, the results are generalizable. The Riloff [13] dataset was omitted for this experiment due to it not being publicly available for download (not available in her publications page).

- **IAC-V1 [9] & IAC-V2 [12]:** Online debate corpora containing sarcastic and non-sarcastic comments. IAC-V1 describes 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. **Total inputs:** 1,993 & 6,520.
- **Ghosh [4]:** 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. **Total inputs:** 41,780.
- **iSarcasmEval [1]:** A dataset where authors explicitly indicate sarcastic intent. This dataset matches exactly to the version used in the original experiment. **Total inputs:** 1,400.
- **SemEval 2018 Task 3 [15]:** A benchmark dataset for irony detection in English tweets. This dataset matches exactly to the version used in the original experiment. **Total inputs:** 4,618.

3.3 Models

There are resource and pricing concerns with the model selections. Common to NLP tasks, there is a 3:1 ratio of input to output tokens, and given the datasets, there are 56,311 total input texts. Through GPT-tokenization, a generous average of 50 tokens can be estimated for each input, and each CoT full conversation utilizes: $33 * 3 + 36 * 2 + 52 = 223$ tokens. Therefore, the total number of tokens (input and output) can be estimated as: $((223 + 50) * 56,311) * \frac{4}{3} = 20,497,204$ tokens. This experiment opts to use GPT-4o-mini due to its affordable cost of \$0.60/1M input tokens, and \$2.40/1M output tokens through the API. To replicate the GPT-4 Turbo results, it costs \$10.00/1M input tokens, and \$30.00/1M ouput tokens.

Similarly, due to the constraints of this project (deadline and local resources), a subset of the used models are selected to be run locally. All models to be used are provided below.

- **LLMs: GPT-4o-mini** (cost-effective) and **LLaMA 3-8B** (adaptable).
- **PLMs: BERT** and **DeBERTa**, selected for computational efficiency.

3.4 Experimental Setup

CoC will be evaluated on selected models across all datasets. Results are compared against zero-shot I/O, few-shot I/O, and few-shot CoT strategies. From *SarcasmBench*, Random, ChatGPT (GPT-3.5), GPT-4 Turbo, LLaMA 3-8B, BERT, and DeBERTa will be compared to.

3.5 Evaluation Metrics

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

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

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 PRELIMINARY RESULTS

As of this project milestone, the following has been accomplished.

- All datasets have been accumulated and converted to data structures labeling each text as "Sarcastic" or "Not Sarcastic." Emojis are converted to plain text.
- CoC Prompting method has been integrated and tested.
- GPT-4o-mini evaluation completed after *2 weeks* of throttled API queries.

4.1 Initial Findings

The preliminary findings are provided in **Table 1**. These results can be summarized below.

1. Compared to few-shot CoT prompting, CoC performed better in recall but slightly lower in precision in all datasets. This suggests that CoC is better at identifying sarcastic instances at the cost of more false positives. Not to mention, CoC saw state-of-the-art recall scores for IAC-V1, IAC-V2, and Ghosh datasets.

2. Obviously, GPT-4o-mini underperformed GPT-4 Turbo in scores. CoC was not powerful enough of a technique to present better results, but it seems to be a strong cost-effective alternative.

3. GPT-4o-mini achieved its highest F1 score on the Ghosh dataset, where sarcasm detection often benefits from textual markers in the tweets ("#sarcasm"). Its poor performance on iSarcasmEval ($F1 = 35.8$) suggests that sarcasm understanding is more difficult without blatant labeling. Ultimately, performance was dataset dependent, which is a trend for the previous techniques as well.

These results suggests that CoC prompting offers a promising approach for sarcasm detection, but requires further testing to confirm these findings. Future work involves:

- Running experiments on LLaMA 3-8B, BERT, and DeBERTa.
- Comparing final results across all models.

REFERENCES

[1] 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

[2] 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

[3] 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

[4] 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

[5] 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

[6] 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

[7] 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

[8] 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

[9] 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

[10] 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

[11] 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

[12] 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

[13] 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/

[14] 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

[15] 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

[16] 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

[17] 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

[18] 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

[19] 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

5 APPENDIX

Table 1: Performance Overview

Table 1 presents a comparative analysis of sarcasm detection performance using different prompting strategies. Metrics include accuracy, precision, recall, and F1 score. ChatGPT (3.5), and GPT-4 Turbo results are given from Zhang et al. (2024) [19], while GPT-4o-mini provides results from this experiment.

Table 1: Performance on five datasets. Bold indicates the best results across LLMs. ChatGPT 3.5 & GPT-4 Turbo using zero-shot IO, few-shot IO, and few-shot CoT are compared to GPT 4.0 mini using few-shot CoC prompting.

	IAC-V1				IAC-V2				iSarcasmEval			
Model	Acc	P	R	F1	Acc	P	R	F1	Acc	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

	SemEval Task 3				Ghosh				**Average Scores**			
Model	Acc	P	R	F1	Acc	P	R	F1	Acc	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