ECE1786 Final Report

Sarcastibot

Jackson Nie (1005282409)
tianchen.nie@mail.utoronto.ca

Yong Kang Kou (1010511849)
yongkang.kou@mail.utoronto.ca

Word count: 1943 / 2000 (Penalty: 0%)

1 Introduction

Sarcasm is a commonly used literary that augments conveyed tone, emotion, and ultimately fundamental meaning. Lack of discernment for sarcasm can cause barriers in language understanding, leading to misinterpretations. Thus, accurate detection is vital in improving sentiment analysis of intelligent systems, allowing them to better respond to a person's intent.

Sarcastibot's objective is to identify sentences containing sarcasm as a class 1 project. Sarcasm is difficult to pick up, often requiring context for accurate analysis. While some sentences are outright sarcastic, many are ambiguous as their meaning depends on the scenario. To capture these nuances, *Sarcastibot* will not only perform a simple binary classification but will also provide a rationale for the classification.

2 Overall Architecture

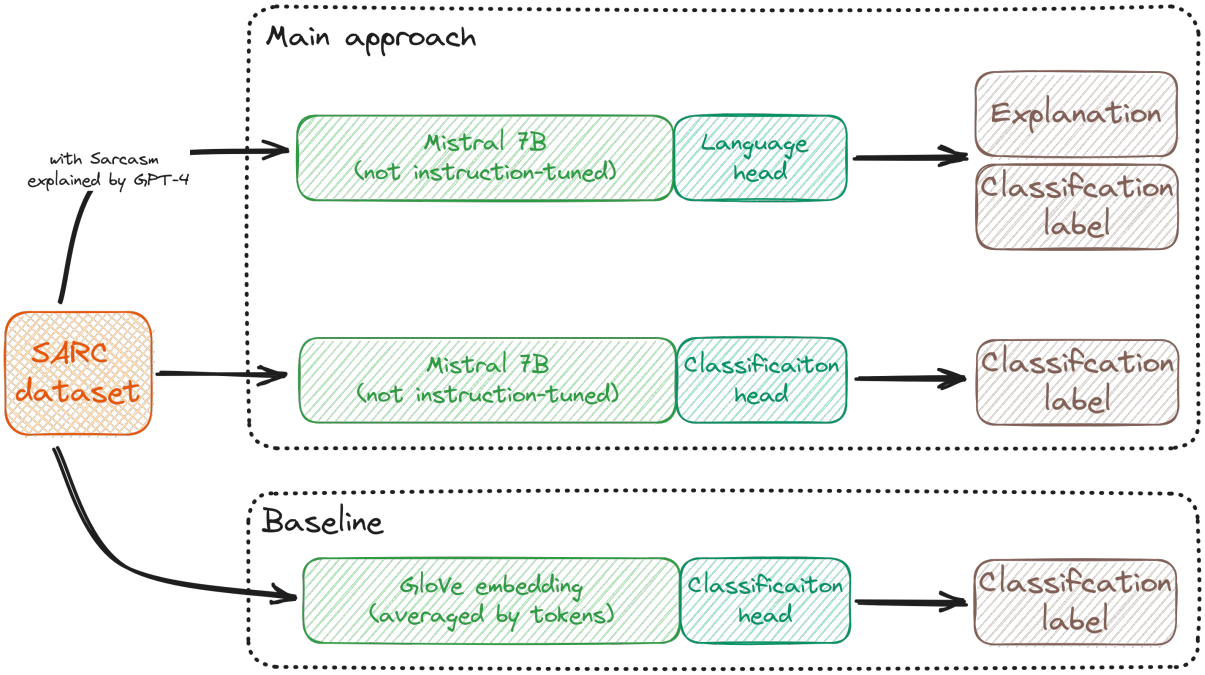

Figure 1: High-level overview of the approaches used in *Sarcastibot*

3 Background & Related Work

Early attempts at sarcasm detection were largely rule-based, relying on features like semantics, and punctuation. Later work focused on better feature generation to be fed into machine learning models.

Chatterjee, Aggarwal, and Maheshwari [1] engineered four features (overtness, exaggeration, acceptability, comparison) based on violations of Grice's Maxims of Quality, in combination with other lexical features including Tf-Idf, number of emoticons and emojis; together with explicit incongruity-based features such as word counts for words with positive and negative sentiments, these features are fed into four dif-

ferent machine learning models (Decision-Trees, Random-Forest-Classifier, Support-Vector-Machine, Gradient-Boosted-Trees). Experimental results showed that the best-performing model is Random-Forest. Testing against their self-collected Twitter data, the best model attained an AUC score of 0.914726.

On the deep learning side, Mandal and Mahto [2] proposed a CNN-LSTM architecture. The word embeddings are passed through 1-D convolutions before being fed into a bidirectional LSTM. Targeting a news-headline dataset, they were able to achieve an 86.16% accuracy. Some limitations include an input length of 20 and that the embeddings are not pretrained, despite having a small dataset with 26709 samples. The authors faced frequent overfitting, preventing a larger model from being developed.

4 Data

4.1 Source of Data

The dataset used is the Reddit sarcasm dataset (SARC) made by Khodak *et al.* [3], containing 1.3 million sarcastic comments harvested from the social news website. 2000 examples spread equally across the two classes are sampled from the original dataset for further processing.

4.2 Processing: Multiclass Approach

SARC is labeled with binary labels to indicate sarcasm. There was an attempt to assign more fine-grained labels: "not sarcastic" (0), "positive sentiment negative situation sarcasm" (1), and "positive situation negative sentiment sarcasm" (2), loosely based on prior work [4]. These labels are mutually exclusive and cover a majority of the dataset. GPT-4 was used to perform this labeling at scale. Following the best practices that help ensure the correctness of the labeling, a small subset was used at the beginning to refine the prompt.

Before prompt fine-tuning, GPT-4 was tasked to create examples of sarcastic sentences that matched the labels. GPT-4 struggled to give an example for label 2, and it took three corrections before it gave a correct example. Thus, it seems that GPT-4 does not understand label 2.

To mitigate this issue, chain-of-thought prompting [5] was used by giving examples of sentences of each class, labeling them with the corresponding label, and giving an explanation in the prompt. However, when labeling 1000 sarcastic examples, 706 were labeled 1 while only 255 were labeled 2, and the remaining 39 were labeled with other labels. Furthermore, when sampling the data, there was a considerable amount of mismatch – many examples that belonged to label 2 were labeled as 1 and vice versa. This inaccuracy is potentially caused by GPT-4′s lack of knowledge.

4.3 Processing: Explanation Approach

Since the previous approach did not work out well, the direction pivoted towards the stretch goal mentioned in the proposal. Similar to the multiclass method, chain-of-thought prompting [5] was used. Example explanations that emphasized reasoning, sarcastic term detection, and ambiguity explanation were provided. After experimenting with a few sample datasets, frequent disagreement between GPT-4 and the binary labels provided in SARC, as well as hallucinations were observed. The original labels in the dataset were added to the prompt as mitigation, effectively asking GPT-4 to explain the given label. With the ground truth added the explanations were significantly better, probably due to GPT-4 not needing to go through a classification round. 70 samples were randomly chosen to evaluate the explanation, all of them followed the label provided, and although there were a few hallucinations, most of the explanations exceeded expectations providing ambiguity analysis and sarcastic-term detection.

comment	glad to see they've upped their prices to deliver high quality advertising like this.
explanation	Praises price increase ironically pointing out bad quality.

Table 1: GPT-4 explanation for sarcastic comment

As seen from Table 1, gpt-4 can capture sarcasm, and further determine the meaning and irony of the comment.

comment	What about outside of that town, you know, beyond thunderdome?
explanation	This is not sarcastic, but rather simply a question about something outside of the given town. The tone could arguably hint sarcasm-ambiguity depending on context, as the tone can be seen as skeptical or cynical.

Table 2: GPT-4 explanation for non-sarcastic comment

Table 2 shows a non-sarcastic example with an interesting explanation. It provides not only reasoning but also an analysis of potential sarcasm-ambiguity.

4.4 Split

The final dataset follows a 8:1:1 train-val-test split.

5 Architecture and Software

Mistral 7B [6] is the backbone of the main model of *Sarcastibot*. It is architecturally similar to Llama2 7B [7] with some minor tweaks, but its authors have shown that it outperforms Llama2 7B with a performance comparable to Llama2 13B. Due to limited computing power, the choice was to use the best model that could be finetuned under the given constraints. Finetuning was performed against the non-instruction tuned

version of Mistral 7B. From Figure 1, it can be seen that finetuning is performed with two different model heads – the language head for next token prediction and the linear classification head for binary classification. The original labels from SARC are used as-is for classification while the language head is trained using explanations generated by GPT-4. For text generation, the input text follows the format shown in Listing 1 and the objective is to predict the body of the answer, i.e. after the answer heading.

```
### Question: Explain if sarcasm is present in the comment below. Provide a
classification label of 0 or 1 at the end to indicate absence or presence of
sarcasm respectively.

<Sarcastic comment goes here>

### Answer:

LABEL: <label>
```

Listing 1: Training input format for fine-tuning with language head

Recent advances in fine-tuning techniques have made it possible to finetune LLMs with limited resources by not performing full finetuning. Instead, small amounts of additional trainable weights are used to directly or indirectly perform fine-tuning. Some of the techniques include LoRA [8] and P-tuning [9], which has been implemented in the PEFT library with tight integration with the Transformers library that is used to obtain the Mistral 7B model. This is however still insufficient to train the model under our compute budget, quantization of the model was necessary to make it fit within the memory restrictions, as demonstrated by prior works [10]. P-tuning was used to finetune the model with classification head while QLoRA [10] was used for the variant with language head.

Virtual tokens	16
Encoder hidden size	128

Table 3: Hyperparameters for P-tuning (classification)

r	8
α	16
dropout	0.05

Table 4: Hyperparameters for QLoRA (text generation)

6 Baseline Model or Comparison

The baseline model utilizes GloVe embeddings [11] to obtain a representation of the comment by averaging the embeddings over the entire comment. This averaged embedding is then fed into a linear layer to perform binary classification.

7 Quantitative Results

Since *Sarcastibot* performs binary classification, a suitable metric to compare performance across various models is accuracy.

Model	Accuracy
Baseline	55.0
Mistral 7B with classification head	70.0
Mistral 7B with language head	85.0

Table 5: Validation performance of models

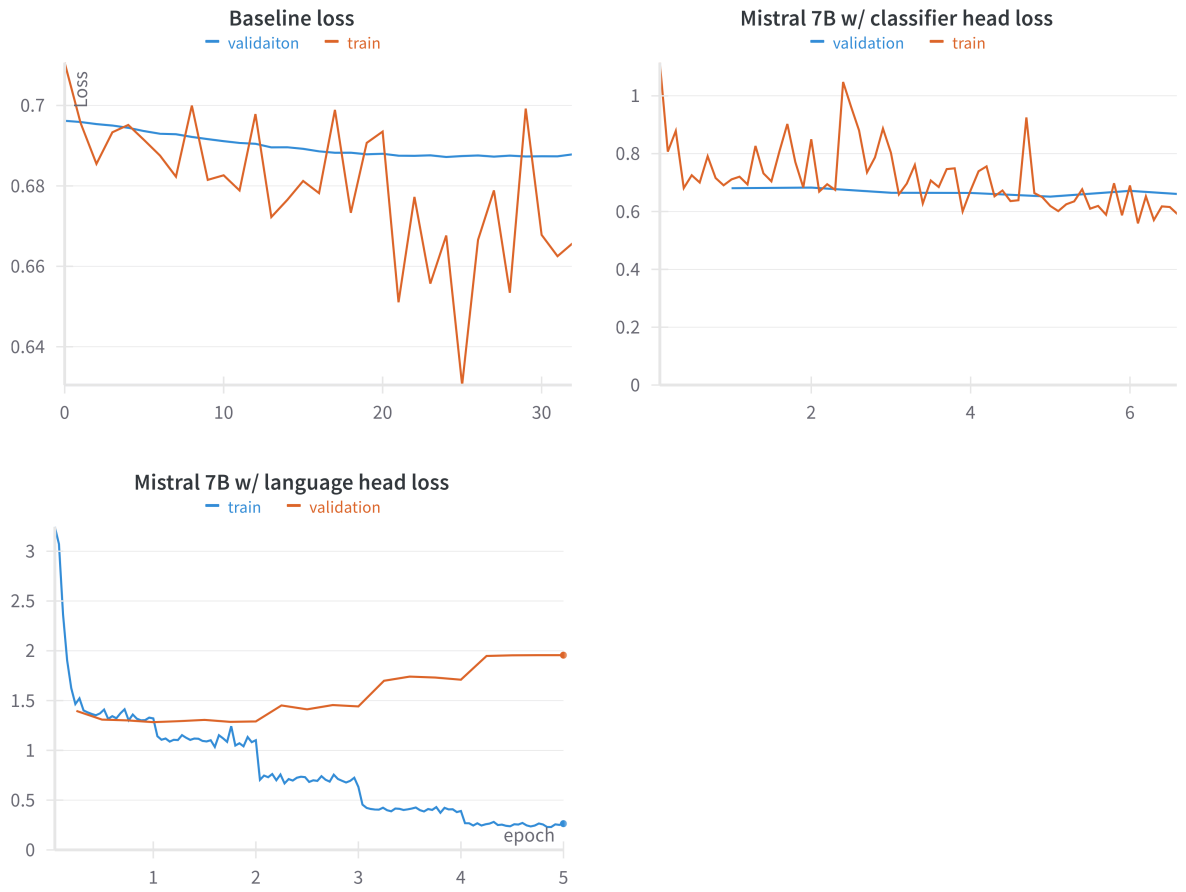

Figure 2: Loss curves

Table 5 is obtained by evaluating the checkpoint with the lowest validation loss and it shows that Mistral 7B with the language head performs the best. For this evaluation, the text generation is performed with greedy decoding. Intriguingly, even when using a parameter-efficient fine-tuning technique that allows for less than 0.3% of the total parameters to be trained, Mistral 7B in text generation mode can overfit the training data easily.

Another surprise is that the baseline performed better than random guessing, beyond margins of error. This is likely due to the presence of keywords such as ‘totally’, ‘ob-

viously' etc that are frequently used when being sarcastic. It is interesting how much better Mistral 7B performs when tuned as a text generator rather than a vanilla classifier.

8 Qualitative Results

8.1 Baseline

word	cosine distance
continues	0.296
ivanovski	0.289
79-79	0.278
cosponsor	0.275
albright	0.266
12212	0.265
49-89	0.264
44-131	0.264
sleptsova	0.261
channer	0.261

Table 6: Top 10 most similar words for classifier weights by cosine distance

Since the baseline is similar to what was done in Assignment 2, the trained weights of the baseline model are compared against the word embeddings, and the most similar words by cosine distance are shown in Table 6. However, this failed to provide any insights since these words do not make much sense in the context of sarcasm. This is largely expected since the model does not perform very well.

8.2 Mistral 7B with explanations

comment	Isn't more accurate when it comes to warlord considering they can drag everyone a mile in any direction after GB?
explanation	This sentence is not sarcastic. The user is simply stating their opinion about a game feature.
prediction	0
ground truth	0

Table 7: Non-sarcastic example

comment	rich and homeless are banned from being homeless, this equality!
explanation	The sentence is sarcastic because it implies that the rich and homeless are not allowed to be homeless, which is an absurd and ironic statement.
prediction	1
ground truth	1

Table 8: Sarcastic example

From Table 7 and Table 8, the model can extract and understand the greater context, especially for the example in Table 7 where no mention of any game is present. This is likely due to the knowledge the model acquired during pre-training, which made it sensitive to any vague keywords present.

comment	I get a new Herschel backpack in a different color every 6 months.
explanation	Although this statement is unambiguous, some might interpret it in a joking manner. The statement doesn't appear to be sarcastic, but its degree of sincerity is ambiguous.
prediction	0
ground truth	1

Table 9: Wrong classification for sarcastic comment

Arguably more interesting results come from the "wrong" conclusion. Looking at Table 9, sarcasm can sometimes be very subtle and context-dependent. The model handles this ambiguity in some cases, but since it has to make a decision, that decision might not always be aligned with the ground truth.

comment	He probably stole them from the cyclist he bashed earlier that day.
explanation	This statement is sarcastic, as the speaker is suggesting that the theft of the food is somehow the result of the earlier assault, which is not explicitly stated in the preceding sentence without any context.
prediction	1
ground truth	0

Table 10: Some hallucination in explanations

Mistral 7B is a LLM and suffers from hallucinations much like others. It will occasionally make references to prior comments that are not provided, though there are only 3 out of 200 observations like these.

9 Discussion and Learnings

9.1 Modelling

Mistral 7B when finetuned with a language head, performed very well. While it is unable to achieve near-perfect accuracy, this is understandable since sarcasm is highly complicated and more context is necessary to decipher ambiguous cases. The stronger classification performance is likely due to the extra explanation involved, which is like chain-of-thought prompting. Without performing tedious prompt engineering, it is possible to benefit from a similar method when finetuning. For specific tasks like this project, finetuning proves itself to be a viable alternative that is likely more reliable in terms of training outcome compared to prompt engineering against an instruction-tuned model.

While LLMs can be used as a classifier like smaller language models, many benefits of LLMs are no longer available if used this way. Since running an LLM is expensive, the extra marginal cost to use as a text generator is a worthy tradeoff if data is available to improve its capability like in this project. Not a lot of data is needed to finetune the model as a text generator since the model head can be reused, unlike classification.

9.2 Data Labeling

Data labeling is a complex endeavor. Although a dataset may seem balanced on a binary scheme, adding even 1 new class can cause severe imbalance. GPT-4 is not perfect – it can hallucinate, and use labels that contradict the instructions provided in the prompt. It is also expensive to use and highly rate-limited thus not scalable. As we have discovered in this project, it is helpful to evaluate GPT-4′s knowledge in the area before using it to perform the task. This is an essential step to avoid needless debugging of prompts, and perhaps show that in-context learning is necessary.

9.3 Other Possibilities

A similar project could have been done differently, especially since a lot of effort has been going into instruction tuning LLMs. Following this trend, it would be better if the performance of GPT-4 is evaluated thoroughly. The instruction-tuned version of Mistral 7B could have been used and evaluated too, reducing the cost and limitations of using an API. Another possible investigation would be to train a classifier with a language head that outputs the label directly as text. This will help confirm the effectiveness of chain-of-thought under this use case as well.

10 Individual Contributions

10.1 Jackson Nie

- Wrote initial baseline model definition and training loop
- Researched/experimented different labeling schemes for multi-class labeling

- Tuned multi-class labeling prompt and performed labeling
- Hand-evaluated 50 samples of muti-class-labeled data
- Tuned explanation labeling prompt and performed labeling
- Hand-evaluated 70 samples of explanation-labeled data

10.2 Yong Kang Kou

- Refactored initial baseline implementation to use PyTorch Lightning
- Implemented finetuning of Mistral 7B with quantization and parameter-efficient finetuning
- Trained baseline and main models
- Converted format of labeled dataset with explanations to CSV
- Evaluated validation-text output of Mistral 7B with language head

Appendix

Permissions

Jackson Nie
To post video: wait till see video
To post final report: yes
To post source code: no

Yong Kang Kou
To post video: wait till see video
To post final report: yes
To post source code: no

Bibliography

[1] N. Chatterjee, T. Aggarwal, and R. Maheshwari, “Sarcasm Detection Using Deep Learning-Based Techniques”, *Deep Learning-Based Approaches for Sentiment Analysis*. in Algorithms for Intelligent Systems. Springer, Singapore, pp. 237–258, 2020. doi: 10.1007/978-981-15-1216-2_9.

[2] P. Mandal and R. Mahto, “Deep CNN-LSTM with Word Embeddings for News Headline Sarcasm Detection”. pp. 495–498, May 2019. doi: 10.1007/978-3-030-14070-0_69.

[3] M. Khodak, N. Saunshi, and K. Vodrahalli, “A Large Self-Annotated Corpus for Sarcasm”.

[4] S. K. Bharti, K. S. Babu, and S. K. Jena, “Parsing-Based Sarcasm Sentiment Recognition in Twitter Data”, in *Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015*, in ASONAM '15. New York, NY, USA: Association for Computing Machinery, Aug. 2015, pp. 1373–1380. doi: 10.1145/2808797.2808910.

[5] J. Wei *et al.*, “Chain-of-Thought Prompting Elicits Reasoning in Large Language Models”, no. arXiv:2201.11903. arXiv, Jan. 2023. doi: 10.48550/arXiv.2201.11903.

[6] A. Q. Jiang *et al.*, “Mistral 7B”, no. arXiv:2310.06825. arXiv, Oct. 2023. doi: 10.48550/arXiv.2310.06825.

[7] H. Touvron, L. Martin, and K. Stone, “Llama 2: Open Foundation and Fine-Tuned Chat Models”.

[8] E. J. Hu *et al.*, “LoRA: Low-Rank Adaptation of Large Language Models”, no. arXiv:2106.09685. arXiv, Oct. 2021. doi: 10.48550/arXiv.2106.09685.

[9] X. Liu *et al.*, “GPT Understands, Too”. Mar. 2021.

[10] T. Dettmers, A. Pagnoni, A. Holtzman, and L. Zettlemoyer, “QLoRA: Efficient Finetuning of Quantized LLMs”, no. arXiv:2305.14314. arXiv, May 2023. doi: 10.48550/arXiv.2305.14314.

[11] J. Pennington, R. Socher, and C. Manning, “Glove: Global Vectors for Word Representation”, in *Proceedings of the 2014 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, Doha, Qatar: Association for Computational Linguistics, 2014, pp. 1532–1543. doi: 10.3115/v1/D14-1162.