Using Large Language Models in Irony Detection – a comparative Analysis

1. Introduction

The advent of commercially available large language models, primarily the Generative Pretrained Transformer (GPT) from OpenAI, not only made large language models more accessible to the public, but also opened a plethora of research and commercial avenues previously stalled, halted or considered impossible. GPT belongs in the category of *generative AI*, which describes models that generate text (or other types of data such as images or videos) using patterns analyzed and learned from a set of training data. These models transform a given input into individual chunks (called “tokens”) and attempt to predict what should be the next word in its response based on different parameters (GPT-3.5 has 175 billion of them (Meer, 2024)) and the training data. More specifically, GPT’s feature of chat completion is a demonstration of GPT as a *Large Language Model (LLM)*, meaning a probabilistic computational model which interprets and generates text from training and input data.

Generative AI models have found application in various industries, such as software development, entertainment or customer service among many others. However, these advancements in AI technology have come with significant risks, like an easier creation of deepfakes or the replacement of human jobs with generative AI, such as artists, designers or programmers. One of the major flaws of generative AI however is not its repercussions, but the manner in which output is generated. Due to its nature as generative AI models like GPT only predict the probability of each word to come next in the response and thus allow for flaws like *hallucination*, meaning the creation of text that is grammatically correct but includes misinformation or completely fabricated factoids. Another critical issue with the nature of generative AI as a trained model is that the size, content and context (among other factors) of the training data can create models that are incredibly biased or simply lack necessary information in order to create factually accurate responses. As such, it is important to note that generative AI models like GPT do not “think” in the traditional sense of the word, they merely analyze and predict. If the patterns of the training data or the structures of the model and its processes are faulty, the responses will include errors or inaccuracies. Referring to GPT in specific, the quality of its responses can vary depending on which version of the model is being examined. While earlier models like GPT-2 have been criticized for their lack of coherence and hallucinations (Quach, 2019) (Vincent, 2019), GPT-3 and especially GPT-4 have been praised for their increased accuracy, coherence and ability to preserve quality of generated text over longer interactions (Piper, 2020) (Heaven, 2023) (Bushwick, 2023). Especially GPT-4 showed the effectiveness of generative AI and especially LLMs, by taking a bar exam and achieving “a score that falls in the top 10% of test takers [which] contrasts with GPT-3.5, which scores in the bottom 10%.” (OpenAI, 2024).

One of the major fields influenced by the advancements in AI technology is *Natural Language Processing (NLP)*, which primarily concerns itself with the decoding of information contained within natural language (meaning language which naturally occurs within human society). Through its ability to interpret text and generate accurate responses, LLMs have become a tool used to assess the accuracy of various NLP tasks as well as perform such NLP tasks itself, such as in the field of sentiment analysis. Sentiment analysis describes the use of NLP and machine learning methods for the purpose of identifying and quantifying the meaning, intent and content of information. However, due to the intricacies of human language and the restrictions of rule-based algorithms (meaning algorithms that apply pre-set written rules to a piece of text in order to analyze its contents), high accuracy in certain sentiment analysis tasks has historically been hard to achieve. Some of these difficult tasks include negation detection, semantic overload, multipolarity and sarcasm detection. Sarcasm detection in particular is an almost impossible task to achieve consistently, as even humans sometimes have trouble accurately assessing sarcasm, due to its seemingly contrarian structure of a statement having an opposite meaning than its naïve interpretation. In addition, sarcastic notions can often be lost due to a lack of context or misunderstandings.

The purpose of this paper is to test and compare the performance of multiple LLMs, primarily GPT-3.5 and GPT-4, in sarcasm detection by using the tools provided by OpenAI and datasets containing ironic and non-ironic statements. Section 2 will detail the background of this experiment, giving an idea of the types of statements that will be analyzed and explaining some of the work that has hitherto been done in irony detection with LLMs. Section 3 will list the various tools used in this experiment as well as give definitions of specific terms within the context of this paper and provide an overview of the structure of the experiment, the interactions with the GPT models and special metrics designed for the analysis of acquired data. Section 4 lists the results and scores obtained as part of the experiment, compares the performance of the examined LLMs and discusses their implications for irony detection using GPT or other AI tools. Section 5 will then go over the future of such experimentation, providing examples of further tests that could be done and giving a conclusion for this paper.

2. Background

2.1 Irony and Sarcasm

2.2 Irony detection using LLMs

3. Methods

3.1 Code

The interactions with the GPT model have been programmed using Python in Visual Studio Code with the OpenAI Chat Completions API. Each GPT evaluation occurs in a new conversation, meaning that the model has no context of previous messages when responding to each input. Every time, it is given the same system prompt and a new message for that evaluation. In addition to the OpenAI Chat Completions API, Pandas was used for loading, reading and saving datasets to and from .csv format. Matplotlib and numpy were used to create figures found in this paper and the repository. Openpyxl was used to read excel tables for score calculations.

3.1 Terminology

When referring to a “run” in this context, it is meant that a model was given a specific number of inputs to evaluate from a dataset and the model’s responses or classifications were parsed and assigned a score. When a run has a size (or sample size) of *x*, it is meant that the first *x* lines from the dataset were evaluated during the run. When referencing a “set” or “set of *(x)* runs”, it is meant that multiple runs have been done on the same data and using the same model and prompt. This set of runs then has calculated averages of result values (such as accuracy or F1-Score). The length of a set refers to its number of runs. A “row”, “line”, “post” or “posting” refers to one specific input, meaning an individual tweet or reddit post from a dataset.

3.2 Datasets

Multiple different datasets were used and compared to ensure that no specific wording or type of input (such as short tweets as opposed to longer reddit threads) would noticeably skew the result values. The main dataset used for evaluation is a subset created by Barbieri et al. for their TweetEval project (Barbiery, Comacho-Collados, Neves, & Espinosa-Anke, 2020), which aimed at providing evaluation frameworks for multiple NLP tasks such as Emoji Recognition, Irony Detection and Hate Speech Detection. For irony detection Barbieri et al. created balanced subsets using the subtask A datasets from the SemEval-2018 irony detection task (task 3) (International Workshop on Semantic Evaluation, 2018). The dataset used in TweetEval to train language models for irony detection, named “tweet\_eval\_irony\_train”, from here on designated “irony\_train”, “dataset 1” or “main dataset” will be the main set used for testing. The dataset to train TweetEval irony detection, named “tweet\_eval\_irony\_test” will be designated “irony\_test”. A manual selection of tweets from “irony\_train”, named “manual\_selection” or “manual dataset”, was created to vet tweets to include more clear examples of irony or non-irony and remove potential mislabelings or unclear/debatable labelings. It was selected as a subset of irony\_train and contains 100 tweets (of the original 2862), some of which are also selected from the first 100 rows of the main dataset the manual set contains exactly 50 ironic and 50 non-ironic tweets. In addition, TweetEval’s irony validation dataset was also included as “tweet\_validate” or “tweet\_val as well as a dataset containing 1950 reddit comments annotated with irony and non-irony, which has been included as “fixedsetreadin”, “reddit comment dataset” or “reddit dataset”.

Analysis will include relevant results across all datasets. The dataset used for a specific run or series of runs will be designated in the discussion section.

3.3 Prompts and Models

Multiple different prompts were used in order to achieve different goals, such as the base binary evaluation of irony content as well as different degrees of confidence or emotional categorizations. As such, the runs are divided into different prompts, each with different intended classification goals. Each prompt also has various subprompts used for prompt engineering with the goal to determine how applying changes to a prompt, however small, may influence the results of a run or the experiment as a whole. The main focus of the experiments lies with GPT, as such each prompt and subprompt has been run on GPT 3.5 as well as GPT 4 with adequate run sizes. For the GPT-3.5 model, the OpenAI designated model ‘gpt-3.5-turbo’ was used (due to no other model being available for GPT-3 or GPT-3.5), whereas for GPT-4, the model ‘gpt-4’ was used. When referencing GPT-3.5 in the context of this experiment, the variant gpt-3.5.turbo is meant.

The following table contains the main prompts (without subprompts) used in the experiment, as well as their classification type and the purpose of the prompt.

|  |  |  |
| --- | --- | --- |
| **Run type name** | **Purpose** | **Default Main System Prompt given to GPT** |
| binary | Default binary yes/no evaluation | You are an irony detector. Respond with '1' (for yes) or '0' (for no) depending on whether you think the following statements are ironic. |
| confidence | Determine confidence in binary evaluation | You are an irony detector. Respond with '1' (for yes) or '0' (for no) depending on whether you think the following statements are ironic, and add a percentage value of how confident you are in your assessment. Make sure your response format is '[1 or 0] [Confidence Percentage]' |
| percentage | Determine how ironic a message is with a percentage value | You are an irony detector. Respond to messages with your evaluation of how ironic the message is, given only as a percentage, such as '55%'. |
| sentimentchoice | Assign a message one of multiple sentiments | You are a sentiment detector. Assign the following tweets a sentiment from the following list depending on which you consider most appropriate: angry, sad, ironic, happy. Respond only with one word. |
| sure | Binary evaluation, then ask if model is sure about its judgement | See run type “binary”, followed up with: “Are you sure? Answer with 'Yes' or 'No'.” |
|  |  |  |
|  |  |  |

Using code, the prompt is used when calling the OpenAI GPT Chat Completions feature by inserting the prompt as the system prompt, which instructs the model. The model then receives one of the postings as an input without additional context and responds. If a model returns a response that is not of a format supported by (or close to) the requirements of the prompt, that response (and entry) is disregarded and counted as an error. For example, during binary classification, the model may return:

*“I'm not sure about that statement. I can't detect irony in it.”*

or

*“I'm sorry, I couldn't detect any irony in that statement. 0”*

If an answer is close to a desired format (such as “Yes.” Or “no” for a classification into “Yes” and “No”), preprocessing steps have been taken to still evaluate those answers as valid. As such, failure cases are few and far between, can be treated as errors and simply disregarded from the final score calculations. The size of the runs is significant enough such that these errors do not influence the overall accuracy or outcome. For instance, a run with a length of 1000 in the “binary” run type using the gpt-3.5-turbo model on the main dataset resulted in an average of 3 such errors.

3.4 Consistency

When evaluating the consistency of a set, the responses GPT gives are evaluated by first counting the amount of correct and wrong evaluations for each classification. If a post is ironic, a set of 10 runs will usually (if no errors occur) result in 10 evaluations from GPT. These evaluations are then counted using a threshold. If the overall proportion of correct evaluations out of the length of the set is equal to or greater than the threshold, this post is counted as being *consistently correct*, meaning GPT associates the post with its correct label most of the time (depending on the threshold). For example, if a post has a label of 1 (ironic) in the dataset, and a set of 10 runs returns the evaluations (1, 1, 1, 0, 1, 1, 0, 1, 0, 1), then 7/10 = 0.7 of the evaluations correctly identify the post with its actual label. If the threshold is equal to or smaller than 0.7, this post is counted as being *consistently correctly* interpreted by GPT. *Consistently incorrect* posts are evaluations that are being incorrectly identified with the wrong label consistently throughout the set. This condition is met if the number of correct evaluations is lower than or equal to 1 – the threshold for *consistent correctness,* in the above example this would amount to 1 – 0.7 = 0.3. If the proportion of correct evaluations for a post is lower than or equal to this threshold, the row is counted as *consistently incorrect* or *consistently wrong*. When referring to *absolutely correct* or *absolutely incorrect* evaluations, it is meant that none of the returned GPT evaluations correctly identify a post with its label (e.g., a post has a label of 1 (ironic) but every GPT run in a set returns this post as 0 (non-ironic)). This can be the case as a misinterpretation from GPT, however, it can also be the case that a post is mislabeled in the dataset. These evaluations will be done when discussing the results of GPT runs in Section 4. An evaluation that is *absolutely correct/incorrect* also counts as *consistently correct/incorrect*. A notable factor however is that *absolutely consistent* rows may only be compared between sets of the same run length, as an increased run length also drastically increases the possibilities for GPT to classify a row differently. For example, a set of 10 runs obviously has more *absolutely consistent* rows than a set of 100 runs, as the likelihood that GPT will classify a row in the same category 100 times is lower than a uniform classification in only 10 runs. Contested rows refer to evaluations that aren’t consistent (i.e., meet neither the threshold for consistent or absolute correctness nor the threshold for consistent or absolute incorrectness).

3.4 Scoring Comparisons

[Scores are rounded to two decimal points, when nothing else is said about a set its assumed to be 10 runs length 100 each, explain that it is assumed that 10 runs is good enough to give an average, but for the sake of comparison, discussion and confirmation that scores don’t wildly change between sets sometimes multiple sets will be examined, but sets will not be not quantified, score differences of up to 0.03 between sets considered acceptable, matrix graphs standard deviation will be the same between tp and fn as well as fp and tn]

4. Results & Discussion

This section will discuss the experiments run using different prompts and datasets, prompt engineering and scoring based on the methods outlined in Section 3. Not every run type may include the same points of analysis. (-> Different prompts have more errors)

4.1 GPT

4.1.1 Run type: Binary

The first and main run type is the binary classification of tweets into ironic and non-ironic. The base default prompt for this type of run is:

*You are an irony detector. Respond with '1' (for yes) or '0' (for no) depending on whether you think the following statements are ironic.*

Prompt engineering has been used to create multiple different prompts similar to the base prompt that will be explained in each subsequent section. The main and longest section will focus on the default prompt.

4.1.1.1 Default Prompt

GPT-3.5

Using the default prompt with GPT-3.5 and a run length of 100 (49 non-ironic, 51 ironic rows) on the main dataset, the average accuracy over a set 10 runs (gpt-3.5-binary-default-10-1) is 0.62, the average precision 0.57, the average recall 0.90 and the average F1-Score is 0.70. Figure 1 shows the score averages obtained from this set of runs. Note that the distribution of scores is quite low, indicating they remain largely similar or the same across runs. This is further exemplified by another set of 10 runs (2024-08-01\_14-28) resulting in almost the same scores (Accuracy: 0.61, Precision: 0.57, Recall: 0.89, F1-Score: 0.69), meaning even across 10 runs these scoring metrics remain remarkably consistent, with differences of only ~0.01 at most, if any.

A graph of a number of values

Description automatically generated

Figure 1: The score averages with distribution measures for a set of 10 runs of the binary prompt using the gpt-3.5-turbo model and the main dataset.

When looking at the number of classifications divided into their predicted and actual labelings, the average values for each type of classification are as follows:

A graph with blue bars

Description automatically generated

Figure 2: The average values of true positive (tp), false negative (fn), false positive (fp) and true negative (tn) evaluations from the run set gpt-3.5-binary-default-10-1. Actual average values are in parentheses after their respective label.

As seen in Figure 2, the highest number of correct classifications are the *true positive* labelings. However, while *false negatives* are low, the number of *false positives* is exceedingly high. The highest amount of deviation in score distribution is seen in both categories of actual non-irony labels (false positives and true negatives), while the actual “ironic” labeled rows are relatively consistent with lower deviation in comparison. Overall, 44.3 out of 49 posts were correctly identified as ironic, whereas only 17.3 out of 51 (just ~34%) of all non-ironic posts were correctly labelled as such. This, in addition to the very high number of false positives and high deviation in actual non-irony rows, may indicate a tendency for GPT-3.5 to simply classify most statements as ironic, as it is possible that due to the prompt being phrased as specifically irony detection, GPT is primed to interpret statements as ironic.

Looking at consistency, most classifications of irony and non-irony are the same over all 10 runs, with only slight deviations in some cases. Using a threshold of 0.7, 81 of 100 rows are consistent, with 52 consistently correct and 29 consistently incorrect evaluations. Out of these, 32 are absolutely correct and 10 absolutely wrong. These numbers are reflected in other sets as well, with 2024-08-01\_24-28 containing 83/100 consistent rows, having 54 consistently correct and 29 consistently incorrect rows. Even with a decently high threshold of 0.7 for consistency, the amount of incorrect consistency is concerning for irony detection. In addition, when examining the types of consistency, consistently correct posts were ironic 45 out of 52 times, with only 7 being non-irony, whereas out of the 29 consistently incorrect posts 28 were labelled non-irony and only 1 consistently incorrect post being ironic. As such a large number of posts, almost 30% of all rows and over half of the non-ironic rows, are being consistently incorrectly identified as ironic, it supports the hypothesis that GPT-3.5 is primed to identify posts as ironic, and thus incorrectly labels most rows, even neutral or definitive non-ironic ones, as ironic. Looking at the posts that were consistently incorrect, they include (for example):

*Need to get back in to college.. #feeling #this*

This is a post labeled as 0 (non-irony). However, 8 out of 10 evaluations in 2024-08-01\_13\_34 interpreted this post as ironic. The punctuation of this post (specifically the double periods) makes almost no difference, as altering this line to include either three periods (“*Need to get back in to college… #feeling #this”)* doesn’t change the consistency, and neither does removing the periods or only placing one. However, when removing both hashtags and otherwise leaving the post the same, GPT-3.5 consistently interprets the statement as non-ironic, meaning that specifically the hashtags “*#feeling #this*” is causing the statement to be interpreted as ironic most of the time. It is possible that GPT-3.5 considers the hashtags to indicate an ironic statement, in the sense that irony is used to express the opposite of what is written (i.e., “feeling this” is interpreted as an ironic statement).

*@user @user you don't know a damned thing about baseball, do you?*

This post is also labelled as non-ironic. Out of 10 evaluations in 2024-08-01\_13\_34, 9 considered this post ironic. It is possible that GPT-3.5 recognizes “*do you?*” as a rhetorical question and rules the statement as ironic. Removing the two “*@user*” doesn’t change GPT’s classifications or consistency.

*well today is gonna be a great day ðŸ‘Œ*

This is the only post that was labelled ironic, but consistently interpreted as non-ironic by GPT-3.5 (here in 9 out of 10 cases). The last series of characters (*ðŸ‘Œ*) represents the OK hand sign emoji in Unicode (👌). Without more information, it is difficult to determine the true intention of the post. While it can be ironic, there is interpretations of this post that don’t include irony. It is however interesting that GPT-3.5 consistently analyzes this post as non-ironic, even if there is debate as to the true intention. Removing the emoji string at the end does not change the result of GPT’s classifications.

There are a number of absolutely correct (32) and absolutely incorrect (10) rows. While 29 out of 32 absolutely correct evaluations are of ironic posts, every single absolutely incorrect evaluation comes from a non-ironic line. This fits with the ratio of consistent correctness as well, with the vast majority of consistently correct rows being ironic, and the vast majority of consistently incorrect rows being non-ironic. The complete absence of absolutely incorrect ironic classifications is notable, though not surprising given only 1 ironic row is consistently incorrectly identified.

As mentioned before, 81 out of 100 rows are consistent in interpretation in the gpt-3.5-binary-default-10-1 run set. However, this leaves the dataset with 19 contested rows, which didn’t meet the 0.7 threshold of unified classification score. These rows are separated into 3 contested ironic posts and 16 contested non-ironic posts, already implying that non-ironic posts are more likely to be contested than ironic ones. Looking at some of the contested rows, they include the following posts:

*I refuse to be weak... #workout #motivation #fitfam*

This post is classified as ironic 6 times and as non-ironic 4 times, while being labeled non-ironic in the dataset. It is in fact not an ironic statement, and it is questionable why GPT-3.5 considers it ironic in the majority cases. It is possible that as in the case above, GPT-3.5 may consider the plethora of hashtags to imply irony as they can be interpreted as being intentionally placed to ridicule the statement. However, as this row is contested, this interpretation of hashtags is a matter of each individual evaluation, and some consider it to rightly be non-ironic in nature.

*@user I'll be a bit sweaty by the time I get to you!*

This is a contested row with an equal distribution of 5 ironic and 5 non-ironic classifications. While being non-ironic and labeled as such, this internal conflict may again indicate GPT-3.5’s predisposition to label posts as ironic due to specifically asking it to determine irony, as there is no clear indication of irony within the post.

To further solidify presented scores and consistency of results, another run set of 100 (2024-08-07\_11-13) was conducted using the same parameters as for gpt-3.5-binary-default-10-1. Figure 3 shows the results from said set. These values remain largely the same, with accuracy remaining at 0.62. Precision and recall remain the same or similar with 0.57 and 0.91 respectively. Overall, F1-Score is also the same with 0.70.

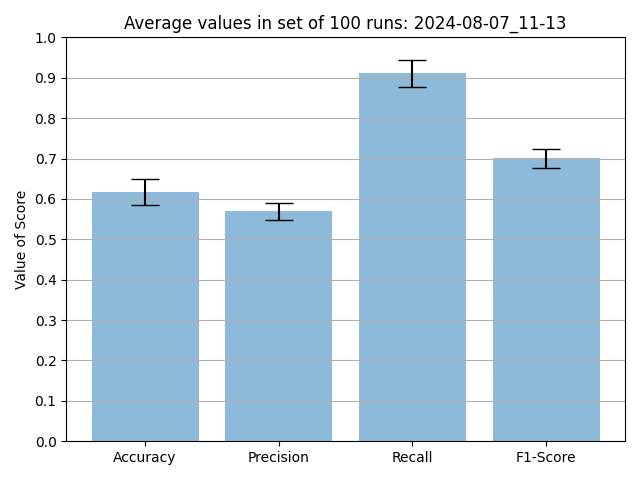


Figure 3: The average scores with distribution measures from a set of 100 using the otherwise same parameters as in Fig. 1 for its runs.

Consistency in this run set with a threshold of 0.7 shows 83 out of 100 rows as consistent. The number of consistent rows also stayed largely the same, with 45 consistently correct ironic rows (same in previous run set) and 9 consistently correct non-ironic rows (7 previously). 2 ironic rows were consistently incorrect (1 previously) and 27 non-ironic rows consistently incorrect (28 previously). Contested were 2 ironic rows (3 previously), and 15 non-ironic rows (16 previously). When examining the same examples from before, all remain in the exact same category of continuity, with only slight changes in the ratio of correct/incorrect classifications. While the number of absolutely correct classifications understandably decreased due to the larger number of runs, there still remained 5 absolutely incorrect classifications. Every single one of these 5 was a non-ironic line which was incorrectly interpreted as ironic. Some examples include:

*Shameless' accounting firms make vast sums advising rich on how to rip off taxpayers - accounting chief|http://t.co/9BSDuJAKXb*

*Breaking up with your girl so you don't have to buy her any presents ||#lowbudget #smartmove #a #good #idea #butscheming doe*

*@user Instead of playing the pompous "do you know who I am card?" , how about you actually make an educated rebuttal?*

It is not obvious why these non-ironic posts are labeled as ironic in every single instance. It is likely that unique features of the posts cause these classifications, as they are each different in writing style, content, intent and rhetorical devices.

When performing a binary run on the reddit dataset across 20 runs (run set 2024-08-14\_12-0, first 100 rows, contain 29 ironic and 71 non-ironic posts), accuracy is 0.35, precision 0.28, recall 0.78 and F1-Score is 0.41. This is a significant decrease in performance compared to runs on the main dataset. It is likely that this is due to the fact that the dataset contains reddit comments, meaning that they will be longer (with an average of ~242 characters per comment across the whole dataset), contain multiple sentences and potentially multiple sentiments within them. In addition, the set is not balanced and contains more non-irony than irony, thus likely making especially GPT-3.5 prone to misclassifications. The standard deviation of classification distribution does not change, with false positives and true negatives still having the highest variation in distribution. Likely due to the smaller number of ironic rows, false positives have increased, and true positives decreased. Consistency shows 89 out of 100 rows as consistent, separated into 24 consistently correct irony and 3 consistently correct non-irony classifications as well as 4 consistently incorrect irony and 58 consistently incorrect non-irony classifications. As expected and observed during runs on the main dataset, consistently incorrect non-irony is the highest metric, followed by consistently correct irony, again pointing to a tendency for GPT-3.5 to classify rows as ironic.

The same binary classification run type run on the manual dataset (2024-08-13\_12-53, 10 runs, length 100 each) results in an accuracy score of 0.58, precision of 0.55, recall of 0.89 and F1-Score of 0.68. The overall results have not largely changed, there is still an overrepresentation of positive evaluations, with the largest standard deviations still occurring with the rows labeled as non-ironic. However, while small, there is a slight decrease in F1-Score, further underlined by another run set of length 10 (2024-08-13\_13-5) with an F1-Score as low as 0.67. All recorded GPT-3.5 binary classifications with this prompt have an F1-Score within the range of 0.69 to 0.71. It is thus a change, albeit small, to have the score be slightly lower. Looking at consistency, a higher number of rows is consistent (from 81 out of 100 to 91 out of 100) and a lower number of rows contested (from 19 to 9), but these changes are largely due to an increase in consistently incorrect non-irony (from 28 to 35). Overall, consistently correct rows have increased by a total of 1 (from 45 irony / 7 non-irony to 44/9), while consistently incorrect rows have increased by a total of 9 (from 1/28 to 3/35). The only absolutely consistent lines are now correct irony (from 29 to 33) and incorrect non-irony (from 10 to 11). Some conclusions that can be drawn from this result are that firstly, performance in scoring decreases when the content of irony becomes clearer. This further supports the supposition of GPT-3.5’s tendency to simply classify most lines as ironic. Results indicate that while GPT-3.5 has become more confident in its evaluations evidenced by the significant decrease (10% of all rows, more than 50% of contested rows) in contested rows, the overall quality of the analysis has not improved, resulting in more consistently incorrect lines.

Another fact about all GPT-3.5 runs done on the main dataset with the default prompt is that while the overall number of consistent classifications does change between run sets, the amount of consistently incorrect classifications does not. In 5 run sets with differing amounts of runs but otherwise the same parameters, exactly 29 rows were consistently incorrectly classified, with only the consistently correct number slightly varying from 50 to 55. Overall, it is fair to say that GPT-3.5 classifications remain remarkably consistent throughout run sets of sufficient sizes. However, there exists only a limited ability to reliably detect irony, due to hinderances such as the large number of false positives or the sizeable number of contested rows.

GPT-4

Looking at the same run type using GPT-4 (OpenAI model gpt-4) to evaluate irony into a binary classification, a run set of 10 with a length of 100, 2024-08-07\_11-32, resulted in the following average scores, seen with distributions in Figure 4:

A graph of a number of values

Description automatically generated

Figure 4: Average scores from a run set of 10 with length 100 using the gpt-4 model.

Most of these scores are an immediate improvement over the same type of run using GPT-3.5, most notably the accuracy increased from 0.62 in gpt-3.5-binary-default-10-1 by 0.15 to 0.77. Another interesting difference to the gpt-3.5-turbo runs is the almost switch in values of precision and recall. Whereas average precision in the exemplary GPT-3.5 run set was 0.57, in this GPT-4 run set it increased massively to 0.86 (difference of 0.29), whereas recall decreased from 0.90 to 0.64 (difference of 0.26). Recall is the only metric to have decreased in score in the overall average of the run set. However, due to the larger increase in precision, calculation of the F1-Score still resulted in a (albeit relatively minor) increase from 0.70 to 0.74. These values stay consistent throughout gpt-4 runs (already indicated by the small amount of deviation seen in Figure 4), as seen in another run using the same parameters, 2024-08-2\_14-2, where the only slight changes were accuracy (0.78), recall (0.66) and F1-Score (0.75).

Already the results show a clear performance improvement compared to GPT-3.5. However, it is necessary to investigate the cause of the changes in especially precision and recall, which lead to the assumption that while the overall irony detection is better, the approaches by which this is achieved may be fundamentally different. This becomes even more apparent when looking at the averaged values of true positives, false negatives, false positives and true negatives. Figure 5 shows an immediate difference between score distributions from GPT-3.5 in Figure 2. While in the GPT-3.5 runs, both true positives and false positives were high, the latter have now dropped to an average of 5.1 over 10 runs, a stark change from the average of 33.7. However, while false positives have dropped, false negatives have increased from 4.7 to 17.5. This, however, is most likely simply due to the overall much larger number of “negative” classifications. When looking at true negatives, their number has seen a large increase, indicating a stronger and better detection of non-ironic content than present in GPT-3.5.

A graph of a number of blue bars

Description automatically generated

Figure 5: The averaged values from the expected and actual label over the run set of gpt-4-binary-default-10-1

It is also notable that the standard deviation is far lower than in the GPT-3.5 runs. This indicates a stronger and more reliable detection (even with increased number of parameters for GPT-4), as opposed to GPT-3.5 which had lower distribution for *true positives* and *false negatives* than the other two scores. For GPT-4 runs, the standard deviation is for all intents and purposes the same for each score. Looking at the scores, it is clear that the overall average number of correct classifications has dramatically increased from 62 out of 100 to 77,4 out of 100 on average, which constitutes a 15,4% increase in performance. More specifically, both false classification scores (*fn, fp*) are lower than each correct classification score (*tp, tn*), unlike during the GPT-3.5 runs, where the number of *false positives* almost doubled the number of *true negatives*. However, it is also necessary to note that the number of *true negatives* is far higher than the number of *true positives*. In fact, an average of 17.5, meaning about 36% of all posts labeled “ironic”, were not correctly identified as such. To contrast this, just 5.1, meaning only 10% of all posts labeled “non-ironic”, were misidentified by the model on average. This indicates the possibility that GPT-4 is decent at irony detection, but much stronger at correctly identifying when no irony is present in a piece of content.

Examining consistency, the vast majority of posts, 98 out of 100, are classified consistently using the threshold of 0.7. 78 of the 98 are consistently correct while 20 are consistently incorrect. The 78 consistently correct rows break down into 32 consistently correct irony and 46 consistently correct non-irony evaluations. Once again, the amount of correct non-irony detection outweighs the amount of correct irony detection. Incorrect posts are separated into 16 consistently incorrect irony and 4 consistently incorrect non-irony classifications. Unlike in the runs with GPT-3.5, the amount of consistently incorrect irony now far outweighs the amount consistently incorrect non-irony, here by a factor of 4. Examining multiple GPT-4 run sets, there is still an obvious performance increase, as while using GPT-3.5, about one third of all evaluations were consistently incorrect, whereas this number has now been reduced to 20 to 21 in all runs. Alongside this, the number of contested rows has also been reduced to only 1 to 3, meaning that the number of consistently correct rows has dramatically increased from 50 to 55 to 77 or 78 (results taken from 5 run sets of each GPT-3.5 and GPT-4). This indicates that not only does GPT-4 deliver more correct evaluations overall, but it also remains more consistent within them. In addition, the amount of inconsistency, meaning contested/unsure rows has been dramatically reduced. Going back to gpt-4-binary-default-10-1, even within consistent categories the amount of absolutely correct and incorrect rows is significantly larger than during GPT-3.5 runs. 26 out of 32 (about 81%) ironic rows were absolutely correct, contrasted with only 29 out of 45 (about 64%) being absolutely correct during the exemplary GPT-3.5 run set gpt-3.5-binary-default-10-1. The number of absolutely correctly interpreted non-ironic posts is even larger at 41 out of 46 (about 89%) posts, while this number was just 3 out of 7 (about 43%) using GPT-3.5. In fact, even when increasing the threshold for consistency to 0.9, still 93 out of 100 rows remain consistent without notable changes in distribution of correct and incorrect interpretations. These results again indicate GPT-4’s far stronger conviction in its evaluations and overall improved performance.

Referring to the examples of incorrect evaluations provided during the binary GPT-3.5 run set, all but 1 have now been correctly identified. The only still incorrectly interpreted post is:

*well today is gonna be a great day ðŸ‘Œ*

As explained earlier, this post is difficult to interpret as ironic without more context and information, making GPT-4’s interpretation of it as non-ironic (in 10 out of 10 cases) a valid evaluation. It is, however, notable that both GPT-3.5 and GPT-4 evaluate this post as consistently non-ironic (in GPT-4’s case even absolutely consistently).

Looking at some of the consistently incorrectly labeled posts by GPT-4, they include:

Halfway thorough my workday ... Woooo

Changing the spelling from “thorough” to “through” does not impact the evaluation. This post was labeled as ironic but interpreted as non-ironic in 9 out of 10 cases. While it is possible that this post is interpretable as someone genuinely expressing happiness at being halfway through their workday, it is more likely to be ironic in intention. Of note is the classification of this message in GPT-3.5 runs. In run sets gpt-3.5-binary-default-10-1 and 2024-08-01\_14-28 this row is contested, while in 2024-08-01\_16-10 this row is just within the threshold of consistent correctness (with 70 runs evaluating this line correctly out of 100). This row shows the existence of evaluations that are correct or at least contested with GPT-3.5, but completely incorrect using GPT-4. Therefore, it is not possible to regard GPT-4 classifications as a flat improvement in all detection mechanisms (even if, of course, the overall average scores are better), as there are certain rows that feature wordings or phrases that would be classified correctly by GPT-3.5 in more cases than by GPT-4.

*ruling party in power#central#state#misusing their power#PM speaking only in foreign parliment#pm to visit out side india during session*

This post is labeled as non-ironic in the dataset, but every one of the 10 GPT-4 runs in the sets gpt-4-binary-default-10-1, 2024-08-09 and even every single run out of 100 in 2024-08-02 considered this post ironic. It is unclear what considerations lead GPT-4 to this conclusion, but it is very possible that as with GPT-3.5, too many hashtags could imply irony in this statement to GPT-4. When looking at this post in GPT-3.5 evaluations, it was absolutely incorrect in gpt-3.5-binary-default-10-1 and only 1 out of 100 runs in the set 2024-08-01\_16-10 classified it as non-ironic. This further highlights that while generally performance is improved in GPT-4, some lines are still misinterpreted.

*@user lol how and what is a cthulhu ?? Funny autocorrect so helpful*

This post, while being correctly labeled as ironic, is still misinterpreted as non-ironic by GPT-4 in 2024\_08-07\_11-31 and other GPT-4 runs. It is possible that GPT-4 is unable to connect the two sentences to arrive at the implication that the user is only ironically praising the autocorrect feature for likely correcting a word into “cthulhu”, a term unbeknownst to the post author. Because this is not explicitly stated, these statements are regarded as unrelated, and the post labeled as non-ironic. GPT-3.5 also classified this posting consistently wrong in all recorded runs.

The two contested rows of gpt-4-binary-default-10-1 were separated into 1 ironic and 1 non-ironic evaluation each.

*Pulis turned down #NUFC cos he wants to spend a load of money on 30 year old journeymen. Parish wouldn't let him & neither would MA. #cpfc*

This post about football manager Tony Pulis is correctly labeled as non-ironic in the dataset but considered ironic by 5 out of 10 evaluations in gpt-4-binary-default-10-1 and 2024-08-09\_10-21, and by 48 out of 100 evaluations in 2024-08-02\_14-5. The exact reasoning is unclear, though it is possible that GPT-4 interprets the phrasing of “*he wants to spend a load of money on 30 year old journeymen”* as ironic, implying that Pulis doesn’t really want to spend this money. This however is inaccurate, as the statement is directed at questioning Pulis’ spending choices for football players in his club. In every recorded GPT-3.5 run set, this line was consistently incorrectly labeled as ironic. While GPT-4 has improved this somewhat, it is still not close to being consistently correct.

My secret name is lizard squad. I like to ruin people's fun time. Follow and rt to a billion and you'll have fun. #psn #giveitup

This is a post labeled as ironic, however there is no clear and obvious ironic sentiment without more context. It’s possible to interpret *“My secret name is lizard squad. I like to ruin people's fun time.”* as ironic, given that it’s likely not true depending on the intent of the author. The fact that it’s not entirely clear is reflected in GPT-4’s evaluations, with 4 out of 10 evaluations considering this tweet as ironic. This is also reflected in 2024-08-02\_14-5, with 38 out of 100 evaluations as ironic, meaning that overall, GPT-4 is more likely to consider this post as not ironic. GPT-3.5 on the other hand considers this post ironic in 96 out of 100 cases in 2024-08-01\_16-10. While GPT-3.5 is correct regarding this line more often than GPT-4, it is likely again due to its proclivity to classify most things as ironic.

When performing a binary run on the reddit dataset across 10 runs (run set 2024-08-13\_12-20), first 100 rows, contain 29 ironic and 71 non-ironic posts), accuracy is 0.73, precision 0.56, recall 0.42 and F1-Score is 0.48. Immediately an improvement is seen from GPT-3.5 in terms of accuracy and precision. F1-Score is also higher. Recall has decreased, but paired with increased precision this indicates a pattern of labeling fewer rows as positive, but only when confident in correctness of the classification, resulting in fewer false positives but more false negatives. This is reflected in the matrix scores, where true negatives have by far the highest amount, with true positive, false negative and false positive scores all being similar. A notable factor is that while scores overall have decreased with both GPT-3.5 and GPT-4 using this dataset, The difference in average F1-Scores has remained almost the same. Whereas the average F1-Score was 0.69 in run set 2024-08-01\_16-10 (GPT-3.5) and 0.75 in run set 2024-08-02\_13-46 (GPT-4) (both run sets of length 100) resulting in a difference of ~0.06 (due to rounding, difference may not be exact), the difference between the GPT-3 and GPT-4 runs on the reddit dataset was ~0.07 (GPT-3.5: 0.41; GPT-4: 0.48). This is further underlined by another pair of run sets on the reddit dataset (2024-08-13\_12-6 for GPT-3.5, 2024-08-14\_11-19 for GPT-4 respectively), in which F1-Scores are also 0.41 and 0.48 respectively, resulting again in a difference of ~0.07, showing consistency. This is still a minute difference in score compared to runs on the main dataset, leading to the assumption that while using the reddit dataset did have an impact on direct scores, the relation between GPT-3.5 and GPT-4 largely stayed the same or within expectedly small deviations. This may indicate that the type of content being evaluated does not cause significant improvements or declines in performance for one model of GPT that are not present in the other, i.e. the models may not have an advantage or disadvantage based on the type of inputs (reddit comments, tweets, etc.).

Running the binary classification on the manual dataset for GPT-4 (2024-08-13\_12-38), the results consisted of an average of 0.84 in accuracy, 0.91 in precision, 0.75 in recall and an F1-Score of 0.82. This is a measurable increase over the main dataset, as no binary classification GPT-4 run with the default prompt resulted in a score higher than 0.75 or lower than 0.73, meaning that this difference in scores of in some cases as much as ~0.9 is noticeable. Another set (2024-08-13\_13-25) also resulted in an F1-Score of 0.82. GPT-4 continues to have very low standard deviation, with an observably lower deviation in the rows with actual label non-irony (false positives and true negatives). This indicates that GPT-4 is again very confident in its negative evaluations, which becomes apparent when reviewing consistency. Overall consistency is virtually the same with only one more line (from 2 to 3) being contested. Consistently correct classifications have improved from 32 irony / 46 non-irony to 37/46. Consistently incorrect classifications have also improved, going from 16/4 to 12/2. These scores already indicate an improvement especially in positive irony detection, as GPT-4 more correctly identifies rows as ironic. Absolute consistency also improved, going from 26/41 to 35/43 in absolutely correct labelings and from 15/4 to 9/2 in absolutely incorrect labelings. Reviewing the overall performance comparatively, GPT-3.5’s performance actually slightly decreased when using the manual dataset, which is a surprising result and indicates that while it can detect irony, it does so by being far too “trigger-happy” when labelling posts as ironic. GPT-4 shows the expected and not insignificant increase in performance when using a manual dataset which was created to specifically include less error labelings and more clear examples or irony and non-irony. However, GPT-4’s analysis is still not perfect, as some more clear examples of irony are still not recognized.

*Oversleeping is the bestttt.*

This post fairly clearly indicates irony, by mockingly enjoying oversleeping. This post was labeled as ironic, but absolutely consistently evaluated as non-ironic by GPT-4. This indicates is that GPT-4 still has issues with contextualizing certain sentiments of irony and somewhat naively approaches and interprets some sentiments that would be clearer to humans.

4.1.1.2 Sub prompt 1: No detector prompt

This prompt removes the first sentence of the default prompt “You are an irony detector”. The intention of this prompt is to remove the specific order for GPT to detect irony and simply leave it with the classification into irony and non-irony. The full prompt for this run is thus:

*Respond with '1' (for yes) or '0' (for no) depending on whether you think the following statements are ironic.*

GPT-3.5

This prompt run on the main dataset (2024-08-14\_13-25, run length 100, set length 20) resulted in an average accuracy score of 0.64, precision of 0.61, recall of 0.71 and F1-Score of 0.66. While not large, there is a difference in average scores to a binary run using the default prompt, particularly in terms of F1-Score (from 0.70 to 0.66). Precision and recall are more balanced, while precision was noticeable lower in gpt-3.5-binary-default-10-1 with 0.57, it has slightly improved and gotten closer to recall which was 0.90. However, it is also clear that precision has improved far less than recall has declined, leading to an overall worse score. Accuracy as well has improved, though only slightly from 0.61 to 0.64.

A graph with blue bars

Description automatically generated

Figure 6: Results from the set of length 20 2024-08-14\_13-25

Figure 6 shows the results from the confusion matrices of each run. Standard deviation is remarkably high across all scores, leading to very inconsistent scoring (thus an increase of standard length of the run set to 20). Another very important difference is the larger number of true negatives. True negatives were far lower than false positives in the default runs, while now outnumbering them. True positives have decreased, and so have false positives. False negatives also increased overall. This further supports the theory that the default prompt is interpreted by GPT-3.5 in a way that increases the likelihood of marking rows as ironic despite there not being context or reason to support such a classification. Removing this condition to a more neutral phrasing thus increases GPT-3.5’s capability to mark more rows as correctly non-ironic. However, with this improvement comes a decrease in true positives, as less rows are marked as ironic, decreasing correct classifications. Thus, ironically, GPT-3.5 seemingly randomly or at least without good reason classifying rows as ironic leads to a better outcome in terms of F1-Score than a more accurate and precise evaluation of these entries.

Consistency shows 70 out of 100 rows as consistent, a decrease of 11 compared to gpt-3.5-binary-default-10-1. The distribution also contains notable changes. While the default run showed consistently correct irony outweighing consistently correct non-irony by a large margin (45/7), this has changed to an almost equal 29/21, showing that while GPT-3.5 is still better at detecting irony, its capabilities to detect non-irony correctly do noticeably increase when using this prompt. In addition, consistently incorrect rows have also decreased overall, from 1/28 irony/non-irony to 7/13, with a slight increase in consistently incorrect irony but a large decrease in consistently incorrect non-irony. Contested rows have increased from 19 to 30 and gone from 3/16 to 13/17 irony/non-irony. Non-ironic contested rows have thus remained almost the same in number, while ironic contested rows have drastically increased by a factor of over 4. This shows the less convinced irony classifications of GPT-3.5, where instead of classifying most rows as ironic, every consistency score is now more balanced (even if not exactly equal).

GPT-4

Looking at a run set of length 10, 2024-08-14\_13-7, the results showed an average accuracy of 0.75, precision of 0.88, recall of 0.57 and F1-Score of 0.69. Already noticeable is a similar drop in overall F1-Score, whereas for GPT-3.5 it dropped from 0.70 to 0.66 (by ~0.04), for GPT-4 it dropped from 0.74 in 2024-08-02\_11-31 to 0.69 (by ~0.05), reducing by a similar amount for both models when using this prompt. Another noticeable difference to the main prompt on this dataset is a drop in recall from 0.64 to 0.57, whereas accuracy and precision have changed by a maximum average factor of ~0.02, which is still within expected deviations and does not indicate a clear difference.



Figure 7: 2024-08-14\_13-7

As seen in Figure 7 and if compared directly to Figure 5, there is no noticeable difference in standard deviation or the general distribution of scores, unlike for GPT-3.5. Consistency shows a similar number of rows consistent at 96 out of 100 (98 for GPT-4 on the main prompt). The distribution also stays similar, with 74 and 22 rows being consistently correct or wrong respectively, whereas these numbers were 78 and 20 for the main prompt in gpt-4-binary-default-10-1. The 74 consistently correct rows separate into 27 irony and 47 non-irony evaluations, whereas the 22 consistently incorrect rows separate into 19 irony and 3 non-irony evaluations. There are no large differences between this set and gpt-4-binary-default-10-1, except for a comparatively minute decrease in both correct irony detection, with consistently correct irony going from 32 to 27 and consistently incorrect irony from 16 to 19, whereas contested rows have gone from 1 irony and 1 non-irony to 3 irony and 1 non-irony. Overall, there is thus a small but notable decrease in consistent irony detection, which may indicate a similar tendency for GPT-4 to now classify less things as ironic after removing the specific instruction for it to do so. Regarding the contested rows, both from gpt-4-binary-default-10-1 have been consistent in the “no detector” run. The contested row marked as ironic is consistently incorrect at 9 out of 10 evaluations, whereas the contested row marked as non-ironic is consistently correct with 10 out of 10 evaluations. Every one of the 4 contested rows in the “no detector” run is a row that was consistent in gpt-4-binary-default-10-1, with the following correct/wrong classifications: 10/0, 10/0, 10/0 and 0/10. This shows that changing the prompt to not include the detector indication influences not just the margins around the threshold, but can sometimes cause rows that were otherwise even absolutely consistent to become contested. Overall while performance in terms of F1-Score did drop an almost equal amount for both GPT-3.5 and GPT-4 when removing the detector instruction from the prompt, GPT-4 shows a far smaller, harder to detect impact on scoring metrics with a small reduction in true positives being the main cause of the decline in score. GPT-3.5 on the other hand had a complete rebalancing of the scoring metrics, with large changes to some scores and a more unbiased view of posts, causing a fairer evaluation which however still caused a decrease in overall score. Thus, while interesting insights have been gained, the “no detector” runs overall are not an improvement over the base prompt for GPT-3.5 as well as GPT-4.

4.1.1.3 Sub prompt 2: Yes/No answer prompt

This prompt changes the phrasing of the instruction to label rows with “Yes” or “No” instead of “1” or “0” to test whether the type of binary classification has any influence over the expected outcome. The full prompt for the run sets examined in this section is:

*You are an irony detector. Respond with 'Yes' or 'No' depending on whether you think the following statements are ironic.*

GPT-3.5

This prompt run on the main dataset (2024-08-16\_11-39, set length 20, run length 100) resulted in an average accuracy score of 0.66, precision of 0.65, recall of 0.65 and F1-Score of 0.65. These scores at first glance are very even, and when looking at their distribution shown in Figure 8, it becomes clear that not only are these scores almost exactly equal, the standard deviation of these scores is also lower compared to the main prompt, seen in Figure 1. Overall, this prompt sees an increase in accuracy (0.62 to 0.66) and precision (0.57 to 0.65) as well as a massive drop in recall (0.90 to 0.65). F1-Score also dropped (0.70 to 0.65).

A graph of a number of values

Description automatically generated

Figure 8: Scores

While there is no clear reason for recall to drop so much consistently throughout all runs by simply changing the answer method from binary to “yes” and “no”, the major drop in recall indicates a more selective and more sensitive evaluation. This is evidenced by the averaged *tp*, *fn*, *fp* and *tn* values, seen in Figure 9.

A graph of a number of blue bars

Description automatically generated

Figure 9: LMAO

The standard deviation is about equal for each metric, whereas in gpt-3.5-binary-default-10-1, deviation was significantly higher on the false positives and true negatives. This shift indicates that there is more variation, and about an equal amount of it, for both positive and negative labelings. There is a large increase in true negatives as well as false negatives, and a decline in both true positives and false positives. However, GPT-3.5 now correctly identifies true negatives most of the time, whereas using the default prompt (seen in Figure 2), false positives far outnumbered the true negatives. The phrasing of the prompt has thus resulted in an almost equal capability of identifying irony and non-irony in the main dataset, and an average correct evaluation of about 2/3 of all rows. This is further reflected in the consistency scores. 79 out of 100 rows are consistent, with 59 consistently correct (28 irony, 31 non-irony) and 20 consistently incorrect (10 irony, 10 non-irony) evaluations. The contested rows are separated into 11 ironic and 10 non-ironic rows. These scores show a remarkable balance in every metric, indicating that GPT-3.5’s irony and non-irony detection are about equal for this phrasing of the prompt and dataset. Compared to the main prompt run on this dataset, this sub prompt introduces a change in behavior coming in the form of a significantly better recognition of non-irony (which is even better than its irony detection), fewer incorrect irony classifications and a model overall less likely to classify the majority of rows as ironic.

Examining this prompt on the reddit dataset (run 2024-08-17\_15-18, set length 20, run length 100) expectedly resulted in worse scores overall, with average accuracy of 0.47, precision at 0.27, recall at 0.48 and F1-Score at 0.35, a decline in most scores compared to the main prompt on the reddit dataset (2024-08-14\_12-0). Once again, a strong drop in recall indicates a more selective process, which is also reflected in the consistency metrics and standard deviation of matrix score distribution, the latter of which slightly increased for every score, indicating a less certain and more varied evaluation. With 89 out of 100 rows consistent, the 27 consistently correct classifications separated into 24 irony and only 3 non-irony rows. This number has changed to 11 irony and 20 non-irony rows. A similar development in the other direction can be seen with consistently incorrect rows, as where 2024-08-14\_12-0 saw only 4 ironic lines consistently incorrect, 58 non-ironic lines were consistently incorrectly classified as ironic. This number has improved in this sub prompt as well, going to 12 ironic and 30 non-ironic consistently incorrect lines. However, a large number of non-ironic lines are now contested, at 21 out of 27 contested lines being labeled with non-irony, which is also a sizeable increase from only 11 lines being contested in 2024-08-14\_12-0. Overall, the reddit dataset run set sees most of the same changes (increased accuracy, drop in recall, decreased F1-Score) going from the main prompt to this sub prompt as the main dataset, and even some of the same behaviors, such as a significantly improved ability to detect non-irony and far fewer false ironic classifications, coming at the cost of some correct ironic classifications. These behaviors are also observable on the main dataset between the main prompt and this sub prompt, showing that these changes in classification behavior stay consistent and aren’t based on the dataset used.

GPT-4

When run using GPT-4 and the main dataset (2024-08-16\_13-20, set length 10, run length 100), the yes/no prompt resulted in an average accuracy score of 0.75, precision of 0.86, recall of 0.58 and an average F1-Score of 0.69. While average accuracy and precision are the similar to gpt-4-binary-default-10-1 (0.77 and 0.86), recall and F1-Score have dropped, from 0.64 to 0.58 and from 0.74 to 0.69 respectively. As expected with GPT-4, the set shows very low standard deviation for these scores.

A graph of a number of values

Description automatically generated

Figure 10:

Felony murder

4.2 Other LLMs

5. Future & Conclusion

Future experiments: Run with removing hashtags, run on more different sets with DIFFERENT TYPES OF CONTENTS (reddit data, tweets, something else etc.) to check if proportionality stays same, multiple manually selected sets of different 100 or more entries of irony and non-irony to check whether comparatively, run experiment inverted (respond with 1 for non-irony and 0 for irony)