Universität Passau

Chair for Computational Rhetoric and
Natural Language Processing

Prof. Dr. Annette Hautli-Janisz

Fakultät für Informatik und Mathematik

Bachelorarbeit

Using Large Language Models in Irony Detection – a comparative analysis

Jonas Barth, 0174 8465887, barth37@ads.uni-passau.de

Matrikelnummer: 103010

Studiengang: B. Sc. Informatik

Fachsemester: 10

Table of Contents

1. Introduction

The advent of commercially available AI tools, 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 the next word in their response based on different parameters (GPT-3.5, as an example, has 175 billion parameters (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 using training and input data.

One of the major flaws of generative AI is the manner in which output is generated. 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. 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, in particular 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. Through their ability to interpret text and generate accurate responses, LLMs have become a tool used to perform various NLP tasks, 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, multipolarity and irony detection. Irony detection in particular is an almost impossible task to achieve consistently as even humans sometimes have trouble accurately assessing irony, due to its seemingly contrarian structure of a statement having an opposite meaning than its naïve interpretation. In addition, ironic 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 irony 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 LLMs. 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

Irony comprises situations or statements which describe the opposite of what is expected or meant to happen. Different types of irony exist, including situational irony (such as the elevators at an elevator repair school breaking down) or verbal irony. The latter describes statements which are intended to convey the opposite sentiment of their literal meaning, such as “I love it when my phone just breaks for no reason”. Sarcasm is a term related to verbal irony in the sense that a sarcastic statement also actually means the opposite of what is said. However, sarcasm is specifically meant to mock or ridicule, and thus often does not include the negative connotation. An example of a sarcastic statement could be “What a great choice to get a white carpet while having two dogs who love to play outside!”. While technically different concepts, sarcasm and verbal irony are closely related. Thus, when referring in this paper to “irony” or “ironic”, it could be that the sentiment is actually sarcastic. However, due to the fact that the experiments of this paper will only be discussing verbal irony or sarcasm in the form of social media posts, it is possible to group the two together in the same concept.

2.2 Irony detection using LLMs

Previous work in the field of irony detection has been done using LLMs, specifically GPT, to classify irony. Aytekin et al. (Aytekin, 2024) used one of the same datasets as used in this paper from SemEval-2018 Task 3, however, the set is not balanced (but in some cases slightly edited) unlike the subset of the task 3 set used in this paper. Aytekin et al. tested multiple models of GPT, including GPT-2 based models as well as GPT-3 and GPT-3.5 based models. The paper also included irony classification on a dataset with a more detailed breakdown of irony into different subtypes.

Gole et al. (Montgomery Gole, 2023) conducted irony detection experiments using multiple GPT models, including GPT-3, GPT-3.5 and GPT-4. *SARC*, which is a dataset including over 1 million reddit comments from various subreddits, is filtered to create a balanced subset containing only comments from the r/politics subreddit, called *pol-bal*, which was used in the experiments. Gole et al. then formed a prompt containing multiple comments in a thread to evaluate irony from GPT. Due to the prompt phrasing, the content of the dataset being political subreddit comments and different versions of fine-tuning, their results are not directly comparable to the results obtained in this paper.

Mu et al. (Yida Mu, 2023) used GPT-3.5-turbo, as well as fine-tuned BERT-large and a fine-tuned version of LLaMA trained by LAIONAI for their sarcasm detection experiments and a different dataset which includes irony to non-irony in a roughly 1:4 ratio on circa 5 thousand rows. Since their results are obtained using different datasets with different weights of irony to non-irony as well as some different models, their work is not directly comparable to the results obtained in this paper.

Most papers conducting irony-detection research using LLMs are done on different datasets or using different parameters, methodologies or models compared to this paper. Only one of these papers includes GPT-4, which is expected as it was released fairly recently. Thus, this paper will be one of the first to conduct extensive experimentation on GPT-4 for irony detection. Comparisons to other work will be made when appropriate, but fundamentally, most other research done is complementary to the results obtained by these experiments.

3. Methods

3.1 Code

The interactions with the GPT models 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. This was done in order to prevent bias based on previous messages. In addition to the OpenAI Chat Completions API, Pandas was used for loading, reading and saving datasets to and from .csv and .xlsx 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. The code and results as well as the paper can be found in the corresponding GitHub repository (Barth, 2024).

3.2 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, the model’s responses or classifications were parsed, and assigned a score. When a run has a length (or size) of *x*, it is meant that the first *x* lines from the dataset were evaluated during the run. When referencing a “set”, “run set” or “set of length *x*”, it is meant that *x* 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 F_1-Score). A “row”, “line”, “post” or “posting” refers to one specific input, such as an individual tweet or reddit post, from a dataset. When referring to *actual irony* or *actual non-irony* in this context, it is in reference to how the rows are labeled in the dataset, rather than how the rows are evaluated by a model in a given run or run set.

3.3 Datasets

Multiple different datasets were used to ensure that no specific wording or type of input (such as short tweets as opposed to longer reddit threads) would skew performance impressions. The main dataset used for evaluation is a dataset 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), which contains tweets labeled with “1” (ironic) or “0” (non-ironic) depending on their irony content. This dataset used in TweetEval to train language models for irony detection, named “tweet_eval_irony_train”, from here on designated “irony_train” or “main dataset”, will be the main set used for analysis in this paper. The first 100 rows of this paper consist of 49 irony-labeled and 51 non-irony-labeled tweets. A second dataset containing tweets with a more detailed breakdown of their irony content into sarcasm, ironic, figurative (meaning both irony and sarcasm) and regular (meaning non-irony) (John, 2020) was altered to manually create a dataset named “manual_select_odd”, from here on referred to as the “manual dataset”. For this purpose, and in accordance with the motivations explained in Section 2, irony, sarcasm and figurative classes were converted to class “1” (for irony) and regular classifications converted to “0” (for non-irony). The manual dataset was preprocessed to remove most hashtags (due to a large number of irony-labeled rows containing for example “#sarcasm” or “#IRONIC”) and consists of 100 rows (50 ironic, 50 non-ironic) that have been selected in order to create a subset with more clear examples of irony and non-irony and to remove potential mislabelings or debatable labelings. In addition, a dataset containing 1950 reddit comments annotated with irony and non-irony (Tatman, 2017) has been included as “fixedsetreadin”, “reddit comment dataset” or “reddit dataset”. Its first 100 rows contain 29 posts labeled as ironic and 71 posts labeled as non-ironic. The dataset used for a specific run or series of runs will be designated in the discussion section.

3.4 Prompts and Models

Multiple different prompts were used in order to achieve different goals. As such, the runs are divided into different prompts, each with different intended classification goals. For the main prompt, various alterations were created used for prompt engineering with the goal of determining how applying changes to this prompt, however small, may influence the results of a run or run set. These altered prompts will be referred to as “sub prompts” in the rest of the paper. The main focus of the experiments lies on GPT. Because of this, each prompt and sub prompt has been run on GPT-3.5 as well as GPT-4 with adequate run sizes and set lengths. 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 sub prompts) 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 '50%'.
sentimentchoice	Assign a message one of multiple sentiments	You are a sentiment detector. Assign posts a sentiment from the following list depending on which you consider most appropriate: angry, sad, ironic, happy, neutral, confused. Respond only with one word.

Table 1: The different main prompts used as system prompts for GPT in the experiment.

Using code, the prompt is inserted as the system prompt (which is the main set of instructions given to the model) when calling the OpenAI GPT Chat Completions feature. 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 implemented in the code to still evaluate such answers as valid. As such, failure cases are few and far between, can be treated as errors and 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.

3.5 Consistency Metric

When discussing 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 for said post. These evaluations are then counted using a threshold. If the overall proportion of correct evaluations out of all evaluations for a post is equal to or greater than the threshold, this post is counted as being *consistently correct*, meaning GPT classifies the post with its actual 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, 1, 1, 1, 0, 1, 0, 1), then 8/10 = 0.8 of the evaluations correctly identify the post with its actual label. Using a threshold of for example 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 minus 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 of 0.3, the row is counted as *consistently incorrect* or *consistently wrong*. If a row is described as just *consistent*, it means that it is either *consistently correct* or *consistently incorrect*. When referring to *absolutely correct* or *absolutely incorrect* evaluations, it is meant that all (or 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), making the classification *absolutely incorrect*). 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 cases will be examined 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* (meaning *absolutely correct* or *absolutely incorrect*) 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 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 correctness* nor the threshold for *consistent incorrectness*). As a standard across all runs, 0.7 is used as the threshold for consistency. While this threshold is not very high due to the fact that in a more optimal case far more than 70% of evaluations should be consistent for a workable model, throughout testing this emerged as a fitting threshold for comparisons and evaluation.

3.6 Scoring Comparisons

Scores, such as accuracy, precision or F_1-Score, are rounded to two decimal points. When a score is prefaced with a tilde symbol (~) it indicates a rounded difference or change. For example, the difference between 0.635 (rounded 0.64) and 0.624 (rounded 0.62) is actually 0.011, which would round to 0.01. However, when looking at the difference between the rounded values 0.64 and 0.62, the difference would be 0.02. In such cases, the latter number will be used and prefaced with tilde to indicate a rounded difference of ~0.02. When a run set is mentioned without a specified parameter, it is assumed that the model and run type is the same as the model and run type of the current section. Run set designations are based on their parameters, in the following format:

gpt-[model number]-[run type]-[sub prompt]-[dataset]

For example, a run set may be called *gpt-3.5-binary-default-main* if the GPT-3.5-turbo model is used with the basic binary prompt and no sub prompt on the main dataset. When nothing is mentioned about the length of the set or the size of its runs, it is assumed that the run set has the standard parameters of set length 10 and run size 100. In most cases, deviation is low enough such that a set length of 10 is enough to get consistent average scores within ~0.02 deviation of each score, which is considered as a similar enough average to compare it to other run sets in this paper. If a set is explicitly named as having more runs, it is because the results of multiple run sets of length 10 had large enough deviations to warrant more runs in a set in order to arrive at a more stable average. In addition, when looking at an averaged graph of the *true positive, false negative, false positive* and *true negative* scores (also named "matrix scores") of a run set, the standard deviation between *true positives* and *false negatives* as well as between *false positives* and *true negatives* is likely each the same, as these score pairs are in a 1 to 1 relationship. If deviation of such a pair is mentioned, it is each the same and can thus be mentioned as one. The only reason the scores would differ within pairs is if there are uncounted error responses from the model that were not fixable in preprocessing. In such a case, the deviation values and errors will be mentioned separately. Otherwise, small numbers of errors that don't impact scoring will not be discussed each time, as explained above.

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 or sub prompt may include the same points of analysis.

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.

4.1.1.1 Default Prompt

GPT-3.5

Using the default binary prompt with GPT-3.5 on the main dataset, the run set gpt-3.5-binary-default-main resulted in an average accuracy of 0.62, average precision of 0.57, average recall of 0.90 and average F_1-Score of 0.70. Figure 1 shows the score averages obtained from this set of runs. Note that the distribution of scores is relatively low, indicating they remain largely similar or the same across runs.

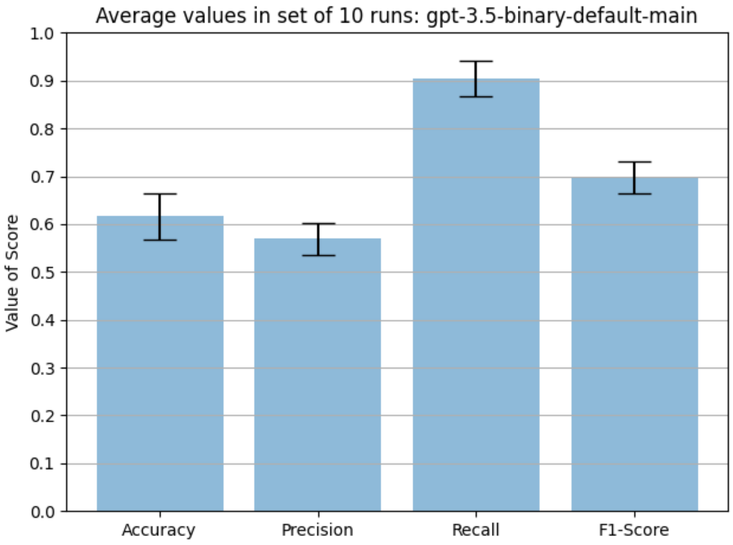

Figure 1: The score averages with distribution measures for a set of 10 runs of the binary prompt using the gpt-3.5-turbo model and the main dataset.

When looking at the number of classifications divided into their predicted and actual labelings, the average matrix scores for each type of classification are shown in Figure 2.

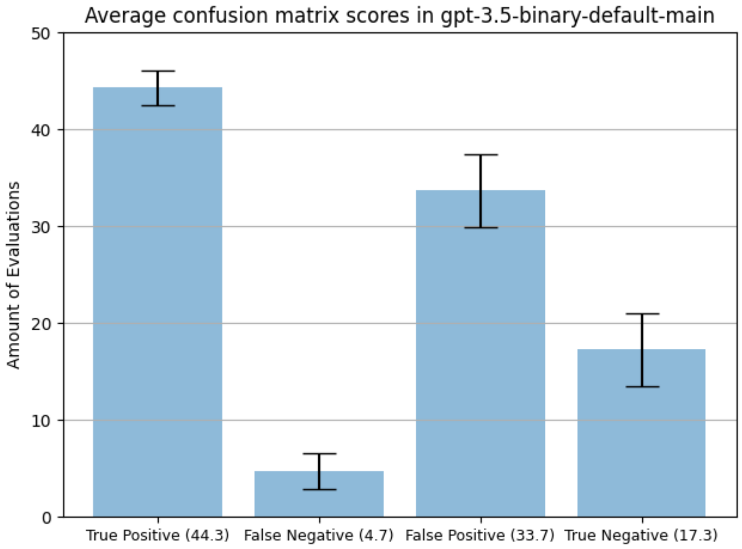

Figure 2: The average values of true positive (tp), false negative (fn), false positive (fp) and true negative (tn) evaluations from the run set gpt-3.5-binary-default-main. Average values are also in parentheses next to their respective label.

The highest number of 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 the pair 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 on average were correctly identified as ironic, whereas only 17.3 out of 51 (just ~34%) of all non-ironic posts were correctly labeled as such. This, in addition to the very high number of false positives and higher 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-3.5 is biased to interpret statements as such.

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 the standard 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. Even with the comparatively low threshold of 0.7, 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 non-irony and only 1 consistently incorrect post was 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 definitively 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 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*" are 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 component).

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

This post is also labeled as non-ironic. Out of 10 evaluations, 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 labeled 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.

While 29 out of 32 absolutely correct evaluations are of actual irony posts, every single absolutely incorrect evaluation comes from a non-ironic row. 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 their interpretation in the run set. However, this leaves the dataset with 19 contested rows, which didn't meet the 0.7 threshold of unified classification score. These contested rows are separated into 3 ironic posts and 16 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.

When performing a binary run set on the reddit dataset across 10 runs (run set gpt-3.5-binary-default-main-reddit), average accuracy is 0.35, precision 0.28, recall 0.80 and F_1-Score is 0.42. 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, which are longer (with an average of ~242 characters per comment across the whole dataset compared to an average of ~78 characters per tweet across the main 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 in conjunction with GPT-3.5's tendency to overevaluate irony, false positives have increased, and true positives decreased. Consistency shows 89 out of 100 rows as consistent, separated into 22 consistently correct irony and 5 consistently correct non-irony classifications as well as 3 consistently incorrect irony and 59 consistently incorrect non-irony classifications. As expected due to high false positives, 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 prompt run on the manual dataset (gpt-3.5-binary-default-main-manual) results in an accuracy score of 0.59, precision of 0.55, recall of 0.92 and F_1-Score of 0.69, showing strong similarities to the run set on the main dataset, with the only score changing by more than 0.02 being accuracy at ~0.03. Overall matrix scores have largely not changed. There is still an overrepresentation of positive evaluations, with the largest standard deviations still occurring with the actual non-irony rows. Comparing consistency to gpt-3.5-binary-default-main, a higher number of rows is consistent (from 81 to 92 out of 100. This change shows largely as an increase in consistently incorrect non-irony (from 28 to 36). Overall, consistently correct rows have increased by a total of 2 (from 45/7 to 46/8 irony/non-irony), while consistently incorrect rows have increased by a total of 9 (from 1/28 to 2/36). Absolutely correct irony has increased (from 29 to 36), just as incorrect non-irony (from 10 to 17) while both other absolute scores remained the same. Some conclusions that can be drawn from this result are that for one, performance in scoring stays relatively the same when the content of irony becomes clearer on a manually selected dataset. This further supports the supposition of GPT-3.5's tendency to simply classify most lines as ironic, without actually evaluating the irony content in detail. Results indicate that while GPT-3.5 has become more confident in its evaluations evidenced by the significant decrease in contested rows (which make up only 8 of all rows compared to the 19 on the main prompt run set), the overall quality of the analysis has not improved, resulting in a large increase in consistently incorrect lines.

GPT-4

Looking at the same run type using GPT-4 to evaluate irony into a binary classification, gpt-4-binary-default-main resulted in the following average scores, seen with distributions in Figure 3.

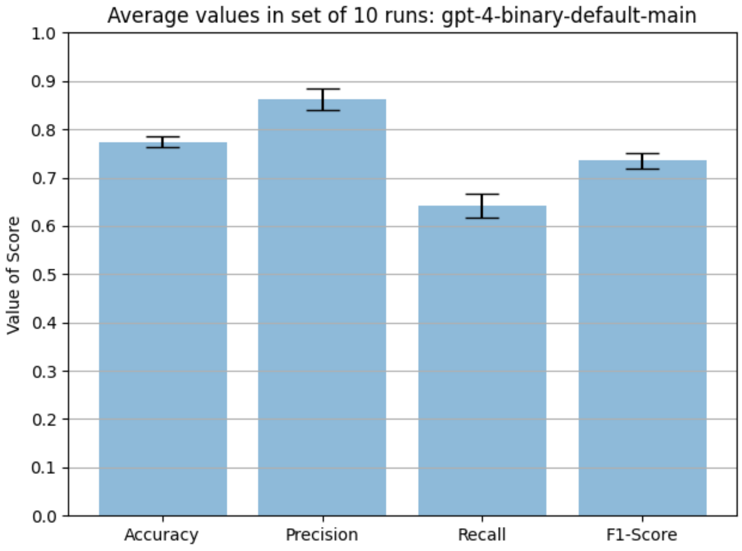

Figure 3: Average scores from run set gpt-4-binary-default-main.

Most of these scores are an immediate improvement over GPT-3.5, most notably the accuracy increased from 0.62 in gpt-3.5-binary-default-main by ~0.15 to 0.77. Another interesting difference 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 F_1-Score still resulted in a (albeit relatively minor) increase from 0.70 to 0.74. The changes in precision and recall indicate a more selective model (lower recall) which however is more effective in its fewer positive evaluations (higher precision), as opposed to the behavior seen with GPT-3.5. These values stay consistent throughout gpt-4 runs, indicated by the exceedingly small amount of deviation seen in Figure 3, which is also notably lower than distributions seen for the scores using GPT-3.5 in Figure 1.

Already, the results show a clear performance improvement compared to GPT-3.5. However, it is necessary to investigate the cause(s) of these 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 matrix scores. Figure 4 shows an immediate difference to score distributions from GPT-3.5 seen 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 true positives have dropped by 12.8, false negatives have increased from 4.7 to 17.5. The latter change in particular, however, is most likely simply due to the overall much larger number of “negative” classifications in comparison to GPT-3.5. When looking at true negatives, their number has seen a large increase from 17.3 to 45.9, indicating a stronger and better detection of non-ironic content than present in GPT-3.5.

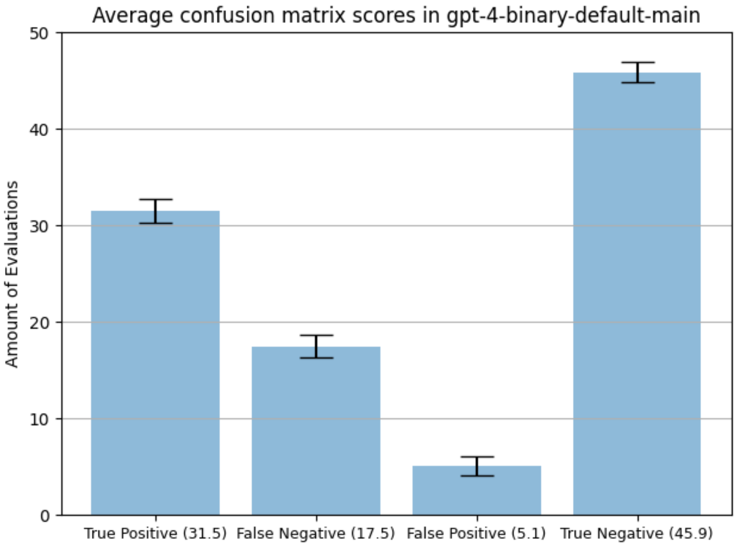

Figure 4: Average confusion matrix scores from run set gpt-4-binary-default-main.

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, as opposed to GPT-3.5 which had lower distribution for *tp*/*fn* than the other pair. For GPT-4 runs, the standard deviation is for all intents and purposes the same for each score (around 1). 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 notable 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, 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 given input.

Examining consistency, the vast majority of posts, 98 out of 100, are classified consistently. 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. 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. 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%) consistently correct ironic rows were absolutely correct, contrasted with only 29 out of 45 (about 64%) being absolutely correct in gpt-3.5-binary-default-main. The number of absolutely correctly interpreted non-ironic posts is even larger at 41 out of 46 (about 89%) consistently correct non-ironic 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 this row’s evaluation in gpt-3.5-binary-default-main as contested. This row shows the existence of evaluations that are 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 runs in gpt-4-binary-default-main 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-main. 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. 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 could be regarded as unrelated, and the post labeled as non-ironic. GPT-3.5 also classified this posting consistently wrong, even though to humans this may be more obvious irony.

The two contested rows of gpt-4-binary-default-main 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. 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, 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, and while it 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 (run set gpt-4-binary-default-reddit, first 100 rows contain 29 ironic and 71 non-ironic posts), average accuracy is 0.73, precision 0.56, recall 0.42 and F_1-Score is 0.48. Immediately an improvement is seen from GPT-3.5 in terms of accuracy and precision. F_1-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 F_1-Scores has remained similar. Whereas the average F_1-Score was 0.70 in run set gpt-3.5-binary-default-main and 0.74 in run set gpt-4-binary-default-main, resulting in a difference of ~0.04, the difference between the GPT-3.5 and GPT-4 runs on the reddit dataset was ~0.06 (GPT-3.5: 0.42; GPT-4: 0.48). This is a comparatively small difference in score compared to the those in 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 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 inherent advantage or disadvantage based on the type of inputs (reddit comments, tweets, etc.).

Running the binary classification on the manual dataset for GPT-4 (gpt-4-binary-default-manual), the results showed an average accuracy of 0.78, precision of 0.75, recall of 0.84 and an F_1-Score of 0.79. This is a measurable increase over the main dataset, at an F_1-Score increase of ~0.05. However, while overall F_1-Score increased, there is an almost switch of precision and recall with similar accuracy compared to gpt-4-binary-default-main, which indicates a change in matrix scores.

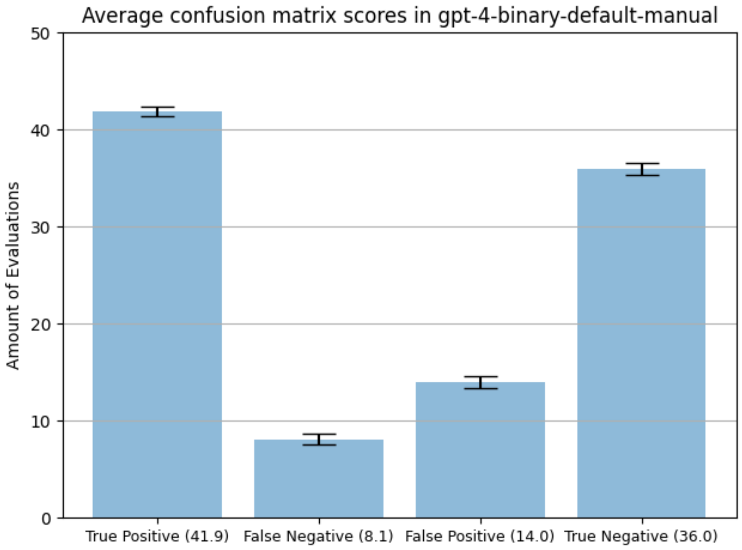

Figure 5: Average confusion matrix scores from run set gpt-4-binary-default-manual.

As seen in Figure 5, matrix scores have changed compared to gpt-4-binary-default-main, now showing better irony detection than non-irony detection. This could indicate actually different engagement methods with the content that were absent when using GPT-3.5, which almost reproduced its results from the main dataset when using the manual dataset. GPT-4 results continue to have exceptionally low standard deviation, with an equally low deviation of less than 1 across all matrix scores, even when compared to already low deviation in the main prompt run set. This indicates that GPT-4 is even more confident in its evaluations, likely due to more clear examples or irony and non-irony in the dataset, which becomes apparent when reviewing consistency. Overall consistency is virtually the same with only one less line (from 2 to 1) being contested. The changes in matrix scores are reflected in the consistency as well, with consistently correct rows going from 32 irony / 46 non-irony to 42/35, due to very low standard deviation these scores are almost perfectly in line with average matrix scores seen in Figure 5. Consistently incorrect classifications have also switched from 16/4 to 8/14, once again showing evaluation scores are more balanced. Reviewing performance comparatively, GPT-3.5's results were about the same when using the manual dataset, which shows that the model's overclassification of irony extends beyond just one dataset of tweets, and also reflects in the reddit dataset (albeit with far more false positives). GPT-4 shows improved irony detection when using a manual dataset, which could be an effect of better and more clear examples of irony. However, paired with decreased non-irony detection, it could be an effect caused by different prevalence of phrases or phrasings, which result in different distributions of scores, which is expected to some degree when using a different dataset. In either case, GPT-3.5's almost stagnant performance is a strong indicator of the model's weaknesses compared to GPT-4, which outperformed GPT-3.5 in all 3 datasets.

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 (gpt-3.5-binary-noDetector-main, set length 20) resulted in an average accuracy score of 0.64, precision of 0.61, recall of 0.71 and F_1-Score of 0.66. While not large, there is a difference in average scores to the binary run using the default prompt, particularly in terms of F_1-Score (from 0.70 to 0.66). Precision and recall are more balanced, while precision was noticeable lower in gpt-3.5-binary-default-main 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 F_1-Score. Accuracy as well has improved, though only slightly from 0.61 to 0.64.

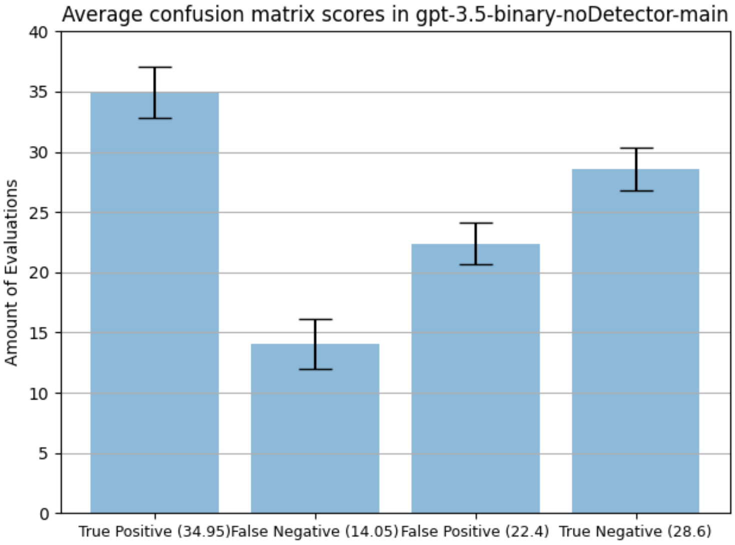

Figure 6: Average confusion matrix scores from run set gpt-3.5-binary-noDetector-main.

Figure 6 shows the results from the confusion matrices of each run. Standard deviation is not low across all scores, leading to more inconsistent scoring (thus an increase of 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 greatly 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 F_1-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-main. 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 irony/non-irony to 13/17. 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 (if not exactly equal).

GPT-4

Looking at run set gpt-4-binary-noDetector-main, the results showed an average accuracy of 0.75, precision of 0.88, recall of 0.57 and F_1-Score of 0.69. Already noticeable is a similar drop in overall F_1-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 gpt-4-binary-default-main to 0.69 in this run set (by ~0.05), reducing by a similar amount for both models when using this prompt, a relation already seen with the reddit dataset for GPT-3.5 and GPT-4. 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 stayed within expected deviations.

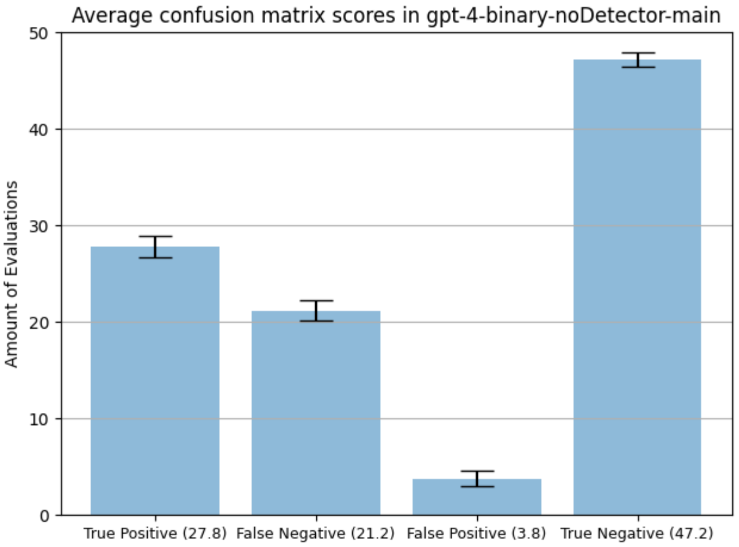

Figure 7: Average confusion matrix scores from run set gpt-4-binary-noDetector-main.

As seen in Figure 7 and if compared directly to Figure 4, there is no noticeable difference in standard deviation or the general distribution of matrix scores. 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-main. 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-main, 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, every one of the 4 contested rows in the “no detector” run is a row that was consistent in gpt-4-binary-default-main, 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 F_1-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 (gpt-3.5-binary-yesNo-main, set length 20) resulted in an average accuracy score of 0.66, precision of 0.65, recall of 0.65 and F_1-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, compared to gpt-3.5-binary-default-main, 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). F_1-Score also dropped (0.70 to 0.65).

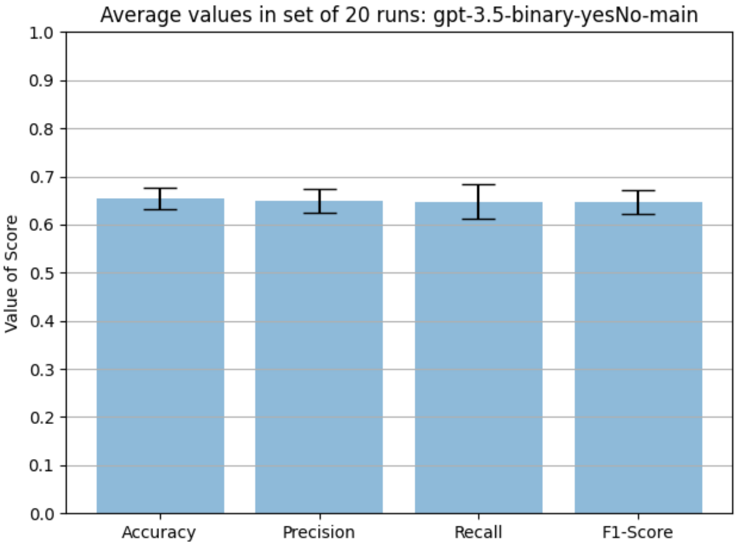

Figure 8: The score averages with distributions for run set gtp-3.5-binary-yesNo-main

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 further evidenced by the average matrix scores, seen in Figure 9.

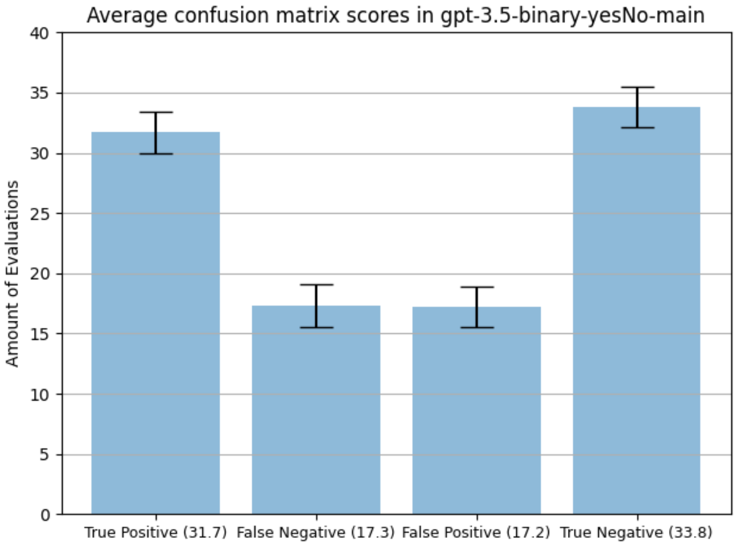

Figure 9: Average confusion matrix scores from run set gpt-3.5-binary-yesNo-main.

The standard deviation is about equal for each metric at ~2, whereas in gpt-3.5-binary-default-main, deviation was higher for the actual non-irony labels. This shift indicates that there is less 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. 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, fewer incorrect irony classifications and a model overall less likely to classify the majority of rows as ironic.

GPT-4

When run using GPT-4 and the main dataset (gpt-4-binary-yesNo-main), the yes/no prompt resulted in an average accuracy score of 0.75, precision of 0.86, recall of 0.58 and an average F_1-Score of 0.69. While average accuracy and precision are the similar to gpt-4-binary-default-main, recall and F_1-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.

When looking at distribution of matrix scores, the standard deviation again is quite low. The scores themselves do not significantly deviate from the results obtained in a binary run on the main dataset using the default prompt (results seen in Figure 4), which is interesting given GPT-3.5's stark deviation from its previous results. Overall, the one of only two observable differences appears in the deviation, which is remarkably low for the pair of *fp*/*tn* compared to the main GPT-4 run, indicating far more consistency in its evaluations for non-irony labelings.

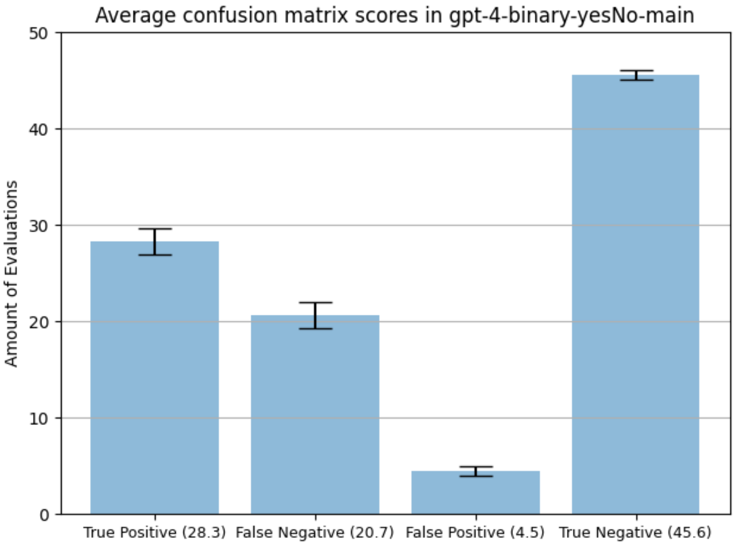

Figure 10: Average confusion matrix scores from run set gpt-4-binary-yesNo-main.

The other difference is the small decrease in average true positives (with deviation for *tp*/*fn* staying similar at ~1.35 compared to ~1.20 in gpt-4-binary-default-main). Consistency reflects this almost perfectly due to low standard deviation, with 97 out of 100 rows consistent as well as 28 ironic and 45 non-ironic consistently correct classifications. Similarly, 19 ironic and 5 non-ironic classifications were consistently incorrect with 3 contested rows separated into 2 ironic and 1 non-ironic. These numbers are quite similar to the results obtained in gpt-4-binary-default-main. An important note is that the contested rows once again do not overlap between these two sets, again indicating some deviation in terms of classifications. Absolute consistency is similar, with this run set showing 20 ironic and 44 non-ironic absolutely correct as well as 16 ironic and 3 non-ironic absolutely correct evaluations. Once again, GPT-4 continues the trend of having the majority of its evaluations be absolute, with absolute non-irony evaluations being larger in proportion to its consistent classifications than ironic ones.

Overall, the Yes/No answer prompt causes significant changes in GPT-3.5's behavior, while GPT-4's behavior was almost unaffected, save for a decrease in standard deviation of one score pair and an overall decrease in most scores due to a slight decrease in true positives. Compounded with previously obtained results, it is thus safe to assume that GPT-3.5 is far more prone to changing the way it evaluates inputs based on the instruction phrasing and answer requirements, whereas GPT-4 results remain largely the same in most aspects. GPT-4 thus seems to be not only better in raw performance, but also has a good capability to extract meaning from the instructions and be uninfluenced by slight prompt engineering changes.

4.1.1.4 Sub prompt 3: One-shot

This sub prompt is based on giving GPT an example of irony or non-irony along with the base prompt. For this purpose, the experiment is divided into two categories: oneshot-0, which gives GPT an example of non-irony, and oneshot-1, which gives GPT an example of irony. The irony and non-irony examples are also from the main dataset, however from far later in the dataset (rows 2685 and 2624 respectively), meaning they are not being evaluated in any of the run sets discussed in this paper. The base prompts are thus:

You are an irony detector. Respond with '1' (for yes) or '0' (for no) depending on whether you think the following statements are ironic. An example of a non-ironic statement: "@user No! I rarely drink at all. Got a stomach bug :-("

for oneshot-0, and

You are an irony detector. Respond with '1' (for yes) or '0' (for no) depending on whether you think the following statements are ironic. An example of an ironic statement: "Always fun when buses don't turn up! It's my favorite waiting outside in the freezing cold for them for like half an hour"

for oneshot-1.

4.1.1.4.1 Oneshot-0

GPT-3.5

The run set for this run (gpt-3.5-binary-oneshot0-main) resulted in an average accuracy of 0.65, precision of 0.59, recall of 0.89 and F_1-Score of 0.71. These results are similar to the regular GPT-3.5 results using the main prompt, with an increase in average accuracy (by ~0.03) being the only notable difference.

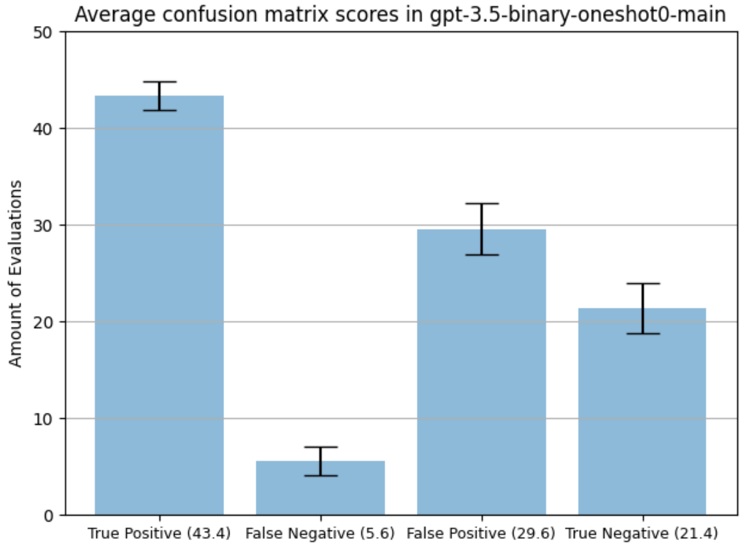

Figure 11: Average confusion matrix scores from run set gpt-3.5-binary-oneshot0-main.

The average matrix scores also show strong similarities in terms of distribution to the main dataset results in Figure 2, with only a slight increase in true negatives and similar standard deviation. It is thus possible that the slight increase in true negatives is due to the specific instructions clarifying the content of non-irony posts, marginally reducing false positives. Consistency shows 84 out of 100 rows as consistent, with 57 consistently correct rows (42 irony, 15 non-irony), 27 consistently incorrect rows (3 irony, 24 non-irony) and 16 rows contested (4 irony, 12 non-irony). These consistency numbers also barely differ from the set gpt-3.5-binary-default-main, which had 52 consistently correct (45 irony, 7 non-irony), 29 consistently incorrect (1 irony, 28 non-irony) and 19 contested rows (3 irony, 16 non-irony). Overall, the theme of increased consistently incorrect non-irony is again present in this set, with marginal changes in non-irony detection. The largest consistency differences are seen, as expected from the results comparing Figure 11 to Figure 2, in consistently correct non-irony, going from 7 to 15, and thus decreasing consistently incorrect irony (from 28 to 24) and contested non-irony (from 16 to 12). This again indicates improved non-irony detection through the given example of non-irony, whereas the number of consistently correct rows barely changed, if anything reduced by a small (but statistically insignificant) amount.

GPT-4

Running the same oneshot-0 prompt on GPT-4 (gpt-4-binary-oneshot0-main) resulted in an average accuracy of 0.79, precision of 0.84, recall of 0.71 and F_1-Score of 0.77. These scores also show, as previously seen for GPT-4, remarkably low standard deviation in comparison to the average GPT-3.5 run sets. Compared to gpt-4-binary-default-main, this set had an increase in recall of ~0.07 and F_1-Score of ~0.03 with accuracy and precision staying within expected deviations of ~0.02. Apart from the increase in recall and F_1-Score, these remain, like for GPT-3.5 oneshot-0 runs, largely the same as their main prompt counterparts, with GPT-4 showing a marginally better result in terms of scoring. The matrix also shows similar results to the one seen in gpt-4-binary-default-main, with true negatives still being the highest metric, followed by true positives, with very similar deviation to the main prompt run. Consistency also shows no significant differences, with (results from gpt-4-binary-default-main in parentheses) 99 (98) rows consistent, separated into 34 (32) consistently correct ironic and 45 (46) consistently correct non-ironic classifications. 15 (16) rows were consistently incorrect irony and 5 (4) rows consistently incorrect non-irony evaluations. 1 non-ironic row was contested, compared to 2 rows in gpt-4-binary-default-main, separated into 1 ironic and 1 non-ironic row each. Overall, the sub prompt had no large effect on results compared to its main prompt counterpart, but a small change in F_1-Score indicate a slightly noticeable performance increase in some metrics, however reflecting in scores strong than the performance increase for GPT-3.5 in this prompt.

4.1.1.4.2 Oneshot-1

GPT-3.5

Results from run set gpt-3.5-binary-oneshot1-main show an average accuracy of 0.69, precision of 0.64, recall of 0.84 and F_1-Score of 0.73. Compared to gpt-3.5-binary-default-main, these results show an immediate increase in accuracy by ~0.07, precision by ~0.07 and F_1-Score by ~0.03. While recall dropped by ~0.06, the overall score improved measurably. This already indicates that the performance of GPT-3.5 increased when being shown an example of irony, more so especially compared to being shown an example of non-irony, where scores had only slightly improved, if at all. Looking at matrix scores, the most notable difference to the main prompt is the fact that true negatives now slightly outnumber the false positives, meaning that the irony example has actually dramatically improved non-irony detection for GPT-3.5.

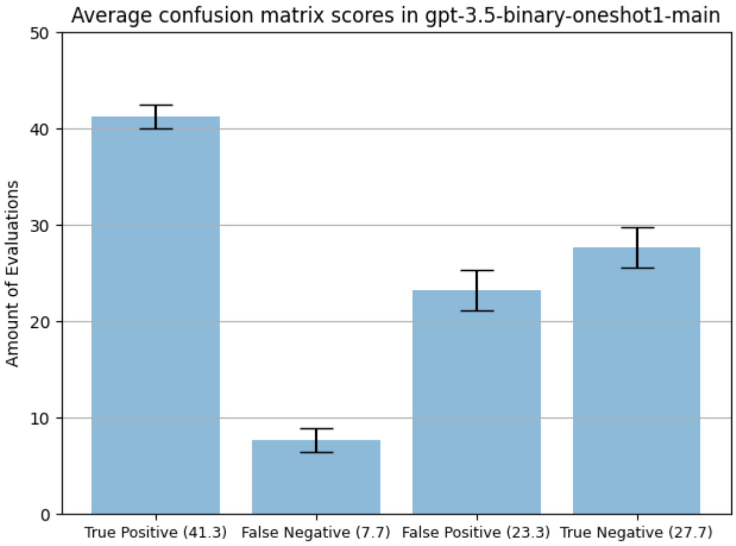

Figure 12: Average confusion matrix scores from run set gpt-3.5-binary-oneshot1-main.

The results also show a lower standard deviation for each score, especially the pair *tp*/*fn*. Consistency for this set also improved compared to the main prompt, with 93 out of 100 rows being consistent compared to 81 out of 100 in gpt-3.5-binary-default-main. The improvement in average true negatives is also reflected, with only 19 consistently incorrect non-irony classifications compared to 28 in the main prompt run set. However, consistently incorrect irony has increased from 1 to 7, and while consistently correct non-irony improved from 7 to 27, consistently correct irony has dropped from 45 to 40. In addition, the already indicated smaller number of contested rows (7 compared to 19) separate into 2 irony and 5 non-ironic classifications, a stark difference compared to the 3 irony and 16 non-ironic contested rows in gpt-3.5-binary-default-main. Overall, there is an observable increase in performance, especially when looking at consistency in primarily non-irony detection, which is as previously discussed a weak point for GPT-3.5.

GPT-4

Looking at the run set gpt-4-binary-oneshot1-main, the results show an average accuracy of 0.78, precision of 0.84, recall of 0.67 and F_1-Score of 0.75. Except for a slight increase in recall (by ~0.03), all other results do not differ from gpt-4-binary-default-main by more than ~0.02, showing practically no real difference to the results obtained using the main prompt.

Matrix scores also show no changes to the main prompt run set and similar standard deviation. Consistency as well shows no discernable differences compared to gpt-4-binary-default-main, strongly indicating that irony and non-irony detection are not affected by one-shot with an ironic given example, further suggesting that GPT-4's irony detection is solid and largely unaffected by prompt changes, a pattern which has been seen in all sub prompts tried in this experiment.

4.1.1.5 Insights from the main prompt experiments

The experiments on the main prompt and its sub prompts indicate that while GPT-3.5 is not terrible at irony detection, this is due to its overarching tendency for ironic labeling, as well as a higher standard deviation on average and worse consistency scores than GPT-4. GPT-3.5 is decent at irony detection, however due to the consistently high recall and lower precision scores, one can infer that GPT-3.5 is not very selective with its labelings, and abysmal in non-irony detection. Overall, due to its lack of non-irony detection, it's difficult to say whether GPT-

3.5 is actually capable of intelligently separating irony from non-irony. Thus, looking at the somewhat adequate F_1-Scores of 0.69-0.71, this might through a naïve analysis seem to indicate GPT-3.5 is only slightly behind GPT-4 when it comes to irony detection, when in fact the methods, processes and results paint a different picture, showing GPT-3.5 having these scores mostly as a result of low sensitivity and an excessive amount of irony labelings. GPT-4 on the other hand appears to excel in non-irony detection and is in fact better at it than detecting irony in most cases, indicated by higher average scores and better consistency for non-irony detection. GPT-4 also displays more confidence in its evaluations, shown by high consistency scores in almost all metrics as well as the fact that most of its matrix scores are very closely reflected in its consistency, indicating low deviation and more importantly few contested rows. GPT-4 does not only produce better evaluations on average, but a more confident and intelligent separation of irony and non-irony, increasing both the number of absolute and consistent rows in comparison to GPT-3.5 in every examined experiment thus far.

Prompt engineering had an impact on both models, however one of the most notable results is that throughout every prompt, the changes applied to it affected the results for GPT-3.5 far more than GPT-4. In fact, a changing of the answer format alone caused GPT-3.5 to have significantly improved non-irony detection, whereas the changes for GPT-4 in this prompt were minimal. This once again highlights the low confidence of GPT-3.5, enabling one to cause massive changes to results by minimally changing the input prompt. In addition, there is no clear or obvious reason for these changes (as detailed in the respective sections), making it unclear why changing the input format sometimes has these significant effects on results. However, also of note is that when experiment results indicated a decline in effectiveness for GPT-3.5, a similar (though sometimes not as stark) decline in overall performance could most often be seen for GPT-4 as well. This indicates that there is a level of similarity between these models that cause a negative effect on one to be reflected in the other to some degree as well.

GPT-3.5	Accuracy	Precision	Recall	F_1-Score	TP	FN	FP	TN
Main prompt	0.62	0.57	**0.90**	0.70	**44.3**	4.7	**33.7**	17.3
No detector	0.64	0.61	0.71	0.66	34.95	14.05	22.4	28.6
Yes/No answer	0.66	**0.65**	0.65	0.65	31.7	**17.3**	17.2	**33.8**
Oneshot-0	0.65	0.59	0.89	0.71	43.4	5.6	29.6	21.4
Oneshot-1	**0.69**	0.64	0.84	**0.73**	41.3	7.7	23.3	27.7

Table 2: Result values from binary prompt run sets using GPT-3.5. Bold highlights the highest value in a column, red the lowest.

Table 4 shows aggregated results from the GPT-3.5 binary prompt runs. The main prompt actually had some of the worst results overall, with accuracy and precision as the lowest scores out of all sub prompts, alongside the lowest amount of negative labelings. However, in terms of F_1-Score, the main prompt ranked squarely in the upper middle at 0.70. The best non-irony detection was seen with the yes/no answer prompt which also almost entirely equalized irony and non-irony detection, but also had the lowest F_1-Score due to far fewer true positives. However, it also showed the highest precision due to the equally decent detection of irony and non-irony. Oneshot-1, as discussed earlier, showed most improvements overall and had the highest F_1-Score and accuracy for GPT-3.5. While this run didn't have any highlights in terms of matrix scores, it still resulted in decent detection of non-irony compared to the main prompt run without really losing any true positives, which is the reason for its higher scores. Overall, GPT-3.5 results are typically high in recall, resulting from high true positives and low false negatives, but low in precision, resulting from high false positives along high true positives. These results underline the tendency for GPT-3.5 to drastically overestimate irony in shown results, with the notable exception of the yes/no answer prompt.

GPT-4	Accuracy	Precision	Recall	F_1-Score	TP	FN	FP	TN
Main prompt	0.77	0.86	0.64	0.74	31.5	17.5	4.1	45.9
No detector	0.75	**0.88**	0.57	0.69	27.8	**21.2**	3.8	**47.2**
Yes/No answer	0.75	0.86	0.58	0.69	28.3	20.7	4.5	45.6
Oneshot-0	**0.79**	0.85	**0.71**	**0.77**	**34.7**	14.3	**6.4**	44.6
Oneshot-1	0.78	0.84	0.67	0.75	33.0	16.0	**6.4**	44.6

Table 3: Result values from binary prompt run sets using GPT-4. Bold highlights the highest value in a column, red the lowest.

Table 3 shows aggregated results from the GPT-4 binary prompt runs. Of note is the series of lowest scores for the no detector prompt, which however had the highest true negatives and precision. Compared to the rest, the no detector prompt may have had similar, if only slightly noticeable effects to GPT-3.5, lowering correct irony detection while boosting correct non-irony detection. It could be the case that due to GPT-4's already high non-irony detection, this had little effect except lowering true positives, resulting in worse scores across the board, barring precision (due to low false positives). Also noticeable is the series of highest scores for oneshot-0, which are mostly still within error (~0.02) of scores for oneshot-1, but still indicate a marginally better result overall. This is likely due to increased true positives, and while true negatives are the lowest out of all sub prompts, they only show a difference of 1.3 to the main prompt set and a difference of 2.6 to the highest true negative score, as well as the lowest number of false negatives overall. Generally, the sub prompt runs for the main prompt show remarkably consistent scores for GPT-4, whereas GPT-3.5's scores were far more malleable throughout the sub prompts.

4.1.2 Run type: Confidence

This run type consists of asking GPT to provide a binary evaluation, while also adding a measure of how confident it is in its classification as a percentage value. The complete prompt for this run type is thus:

> *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]'*

It is necessary to add the format requirement, as otherwise the responses are not parseable due to GPT answering with different sentence structures.

4.1.2.1 Confidence run main prompt

GPT-3.5

The results from run set gpt-3.5-confidence-default-main show an average accuracy of 0.54, precision of 0.51, recall of 0.99 and F_1-Score of 0.68. Compared to the main prompt run gpt-3.5-binary-default-main, this constitutes a decrease in accuracy by ~0.08, precision by ~0.06, an increase in recall by ~0.09 and an F_1-Score within margin of error. The stark differences in accuracy and recall indicate a problem of low sensitivity that is even more pronounced here than in the base prompt GPT-3.5 runs. However, all scores have exceptionally low deviation at less than 0.01 for every score.

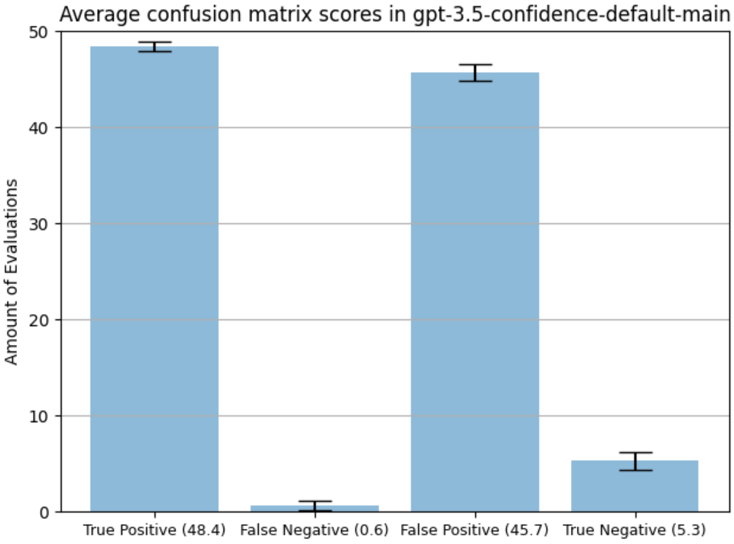

Figure 13: Average confusion matrix scores from run set gpt-3.5-confidence-default-main.

The problem is further underlined by looking at matrix scores in Figure 13. While the main prompt run sets already had a strongly increased false positive rate, the problem is exacerbated in the confidence run set, with an exceptionally low true negative rate. Also noteworthy is the standard deviation for *tp*/*fn* being only ~0.49, with deviation for *fp*/*tn* being ~0.9, indicating that true positives are a little bit more consistent, but both pairs have very low standard deviation. This indicates GPT-3.5 has unusually strong confidence in its evaluations.

Looking at the consistency of this run set, 98 out of 100 rows are consistent, a hitherto unseen number for GPT-3.5 run sets. Furthermore, out of the 52 consistently correct rows, 48 are ironic with 4 non-ironic classifications. The remaining 46 consistently incorrect rows all have a non-ironic label, as well as 1 contested irony and non-irony row each. Another interesting metric is absolute consistency, which shows 46 out of 48 consistently correct irony classifications as absolute (difference of 2), with none of the consistently correct non-irony evaluations being absolute. 39 out of the 46 consistently incorrect non-irony rows were absolutely incorrect (difference of 7). This shows that while GPT-3.5 is very, mostly even absolutely consistent with correct irony, there is still some marginally higher difference for consistently incorrect non-irony.

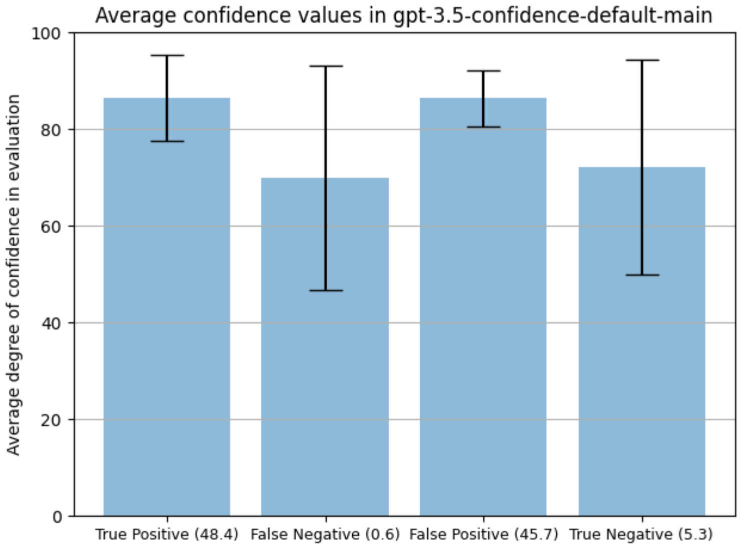

Figure 14: Averaged confidence values for the run set on a percentage scale from 0-100%.

Figure 14 shows the confidence percentages for the run set. Immediately noticeable is that the average confidence in evaluations is equal to or greater than 70% for all values. However, large standard deviations for false negatives and true negatives indicate that these deviate strongly, meaning the actual values are more spread out across the scale from 0-100. In the positive labelings however, standard deviation is comparatively low, which is also due to the higher number of evaluations, but nevertheless indicates a stronger degree of confidence in positive labelings. Also notable is the similar average confidence value in true and false positives, with the only difference being slightly higher deviation for true positives (~8.84 compared to ~5.82). Overall, GPT-3.5 gives fairly high confidence values, especially for positive labelings, even if about half of these labelings are actually incorrect (false positives). The problem of incorrect positive labelings not only continues but is far more pronounced in this run type. This indicates that asking GPT-3.5 to give confidence values in its evaluations leads to more actual positive evaluations as well, most of them with very high confidence percentages given. As such, it is fair to say that this prompt actually reduces the effectivity of GPT-3.5 in irony detection and basically nullifies any semblance of non-irony detection it had using the main prompt.

GPT-4

The results from run set gpt-4-confidence-default-main show an average accuracy of 0.79, precision of 0.86, recall 0.68 and F_1-Score of 0.76. Compared to gpt-4-binary-default-main, this constitutes a small increase in recall (by ~0.04), however all other scores are within ~0.02 of each other, already indicating no large change in performance.

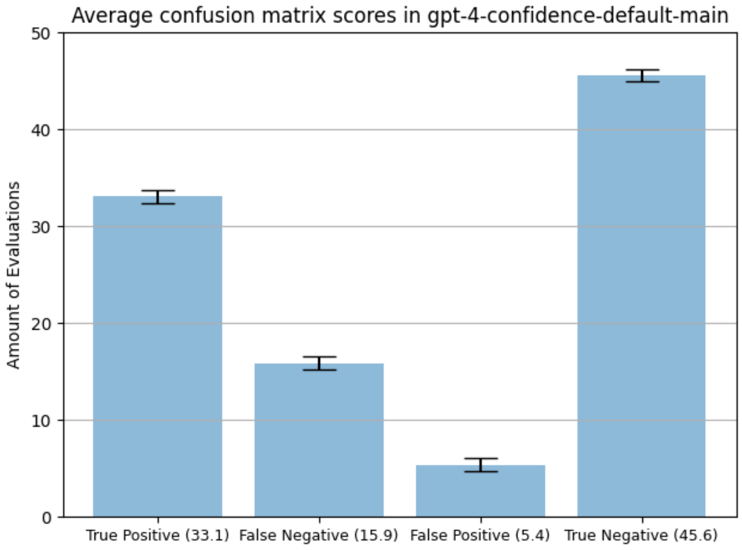

Figure 15: Average confusion matrix scores from run set gpt-4-confidence-default-main.

As seen when comparing Figure 15 and Figure 4, there is no noticeable difference between GPT-4 sets when using the main prompt or the confidence prompt in absolute average values, however a small decrease in deviation is noticed for the two score pairs. Comparing consistency, no score shows any noteworthy difference.

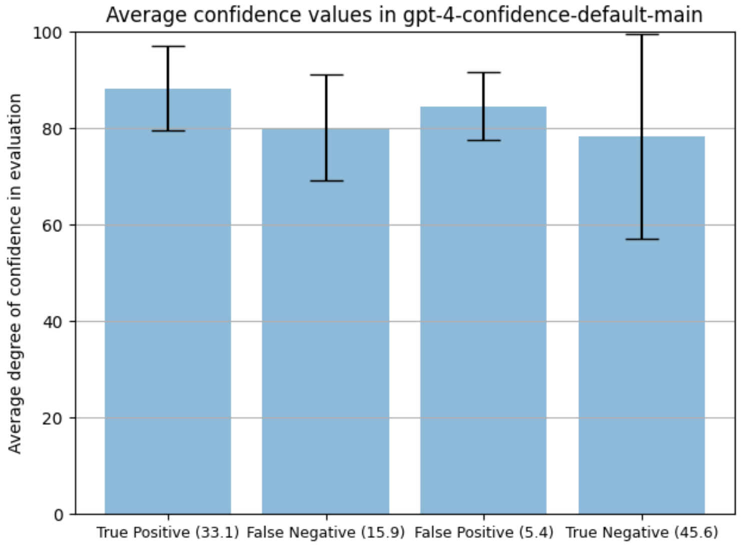

Figure 16: Averaged confidence values for the run set on a percentage scale from 0-100%.

Figure 16 shows the average confidence values for each metric across the 10 runs. Immediately noticeable is the comparatively higher average confidence and also the fact that all 4 metrics are far closer to each other than in the GPT-3.5 run seen in Figure 13. Another notable fact is the higher confidence deviation for negative labelings than positive labelings (similar to the results for GPT-3.5), especially seen in the true negative category. However, with true negative classifications being the highest of all 4 metrics, this means that GPT-4 actually has the lowest average score and the highest variation in confidence values for the score that it is most correct in. Though it is unknown what exactly causes this to be the case, it is possible that due to the irony detection instruction, GPT is more unsure about detecting non-ironic posts. On the other hand, it could simply be due to non-ironic posts possibly being interpretable as ironic, whereas ironic posts could be more unambiguously ironic.

4.1.2.2 Insights from confidence prompt experiments

As seen in previous run types, GPT-3.5 once again shows the largest difference to its main prompt runs with significant changes in both absolute scores and deviations. This again supports the supposition that GPT-3.5 is a far more malleable model than GPT-4, which showed no real difference to its main prompt counterpart run sets. This leads to the conclusion that not only is GPT-4 (as seen in other run set comparisons on the main prompt) more consistent, but it would also in fact appear that for its evaluations, GPT-4 was not influenced by having to give confidence percentages in its responses, as the rest of the prompt was exactly the same as the main prompt. Overall, this run type sees the same pattern of differing GPT-3.5 to GPT-4 behavior as expected from previous results. The behavior in regards to confidence however is similar between both models, showing high confidence and low deviation for positive labels and a lower confidence and higher deviation for negative labelings (regardless of how they are labeled in the dataset).

4.1.3 Run type: Percentage

The concept of this run type is to have GPT evaluate a post with a percentage value of irony instead of a binary evaluation. The base prompt for this run is:

> *You are an irony detector. Respond to messages with your evaluation of how ironic the message is, given only as a percentage, such as '50%'.*

As these results are not classified into irony and non-irony, in order to evaluate scores, a result of 50% or greater is counter as an irony classification, whereas any other value is counted as a non-ironic classification.

4.1.3.1 Percentage run main prompt

GPT-3.5

The results of run set gpt-3.5-percentage-default-main show an average accuracy of 0.69, precision of 0.67, recall of 0.74 and F_1-Score of 0.70. While F_1-Score is the same as in gpt-3.5-binary-default-main, this run actually shows an improvement of ~0.07 in terms of accuracy and ~0.1 in precision, making both of these scores some of the largest differences to other GPT-3.5 runs. Recall on the other hand has dropped by ~0.16, which along with higher precision indicates a more selective result with lower sensitivity. Standard deviation shows no large change for this run set, meaning the distribution of scores around the average stayed roughly the same even if the absolute value of the score changed.

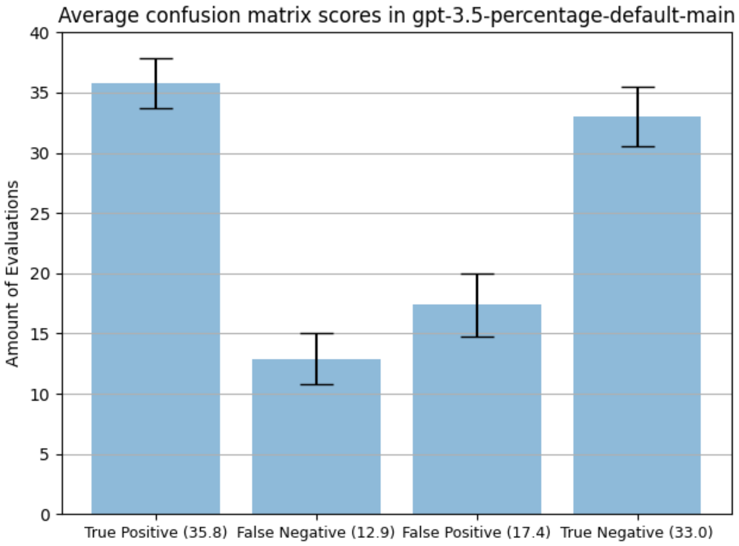

Figure 17: Average confusion matrix scores from run set gpt-3.5-percentage-default-main.

Figure 17 shows the averaged matrix scores for the run set. Immediately noticeable is a shift in actual non-irony labels, whereas the main prompt run sets (as well as alternate prompts and run types) continuously had higher false positives than true negatives, this run set shows a large increase in true negatives, making them almost as high as true positives, meaning that this prompt actually caused non-irony detection to drastically improve for GPT-3.5. This could be due to the fact that having to assign an actual value to the score forces GPT-3.5 to evaluate actual irony and non-irony content in detail instead of overall sentiment of a post, causing it to more accurately assess non-irony. However, this increase in true negatives also comes with a small decrease in true positives, making the model more balanced in its results but not show a large increase in for example F_1-Score. Both pairs also have gotten closer in terms of standard deviation, whereas in the main prompt run set the actual irony rows showed far less deviation than the actual non-irony rows. This again leads to the conclusion that the tendency for overconfident irony labelings is far more reduced, if not eliminated using this prompt. Consistency compared to gpt-3.5-binary-default-main (results in parentheses) shows a larger amount of consistently correct rows at 65 (52), and less consistently incorrect rows at 21 (29), making 14 (19) rows contested. As seen in the matrix scores, consistently correct rows have become far more balanced at 35 (45) irony and 30 (7) non-irony, marking a massive increase in consistently correct non-irony. Contested irony has increased to 6 (3), but contested non-irony has halved to 8 (16). Overall, the results can be considered better than runs done on the main prompt and many of its sub prompts, as there is a clear increase in non-irony detection, leading to the conclusion that the model stops overinterpreting most rows as ironic when asked to give percentages as answers.

GPT-4

Run set gpt-4-percentage-default-main resulted in an average accuracy of 0.76, precision of 0.91, recall of 0.58 and F_1-Score of 0.70. Compared to gpt-4-binary-default-main, precision has increased by ~0.04 while recall has dropped by ~0.06. Of note is also the decrease in F_1-Score by ~0.04 while F_1-Score has remained the same for this run type using GPT-3.5, making this prompt cause a decrease in score for GPT-4 and not GPT-3.5, a pattern not yet seen with any other prompt. The comparatively small increase in precision and drop in recall also indicates a slightly more sensitive and selective model.

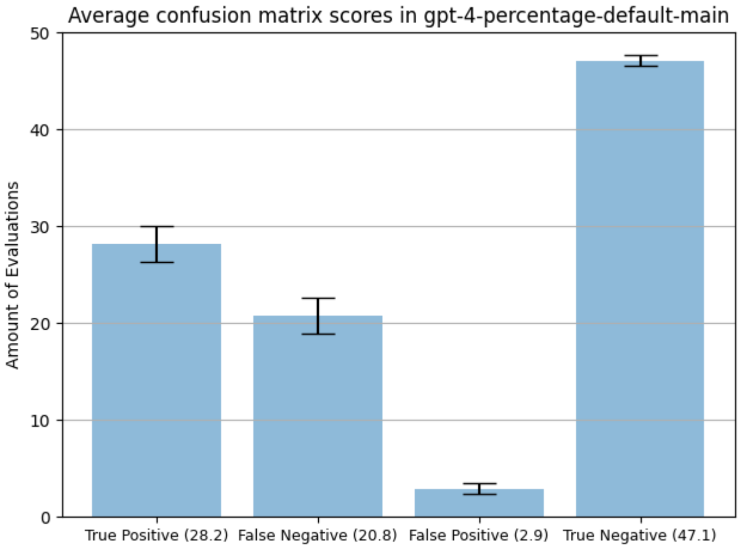

Figure 18: Average confusion matrix scores from run set gpt-4-percentage-default-main.

Figure 18 shows a similar pattern to the one usually seen in GPT-4 runs, having high true negatives with true positives as the second highest score. However, true positives have slightly decreased and become more balanced with false negatives, whereas true negatives have become even larger in number compared to gpt-4-binary-default-main. This experiment also has higher standard deviation for actual irony labels at ~1.83 compared to ~1.20 in the main prompt run, as well as lower standard deviation for actual non-irony labels at ~0.54 compared to ~1.04. This prompt has thus shown a more confident and correct evaluation of non-irony for GPT-4, while at the same time causing a more scattered and less effective irony detection. It's possible that the already better non-irony detection has been improved by a more thorough analysis of sentiment in order to give a better evaluation of irony as a percentage, however also making GPT-4 less sure about the irony content of posts. Except for a small decrease in consistently correct irony detection from 32 to 28 and a slight increase in contested rows from 1 irony and 1 non-irony to 4 irony and 2 non-irony, consistency shows no large differences to gpt-4-binary-default-main. Absolute consistency shows a more significant drop in absolutely correct irony from 26 to 15, further supporting less confidence when evaluating irony for this prompt using GPT-4.

4.1.3.2 Insights from percentage prompt experiments

This run type showed a far better ability for GPT-3.5 to detect non-irony and seemed to almost eliminate its tendencies to incorrectly interpret a plurality of the rows as ironic, a tendency that has been observed not only in the main prompt runs and many of its sub prompts, but also in the confidence run type. This indicates that the phrasing of the main prompt once again has a very significant impact on performance using GPT-3.5, causing what is almost a flip in true negatives and false positives. The same cannot be said for GPT-4, which for the first time saw a performance decrease in terms of F_1-Score whereas GPT-3.5's score remained the same compared to the main prompt run sets. Non-irony detection slightly improved while irony detection saw a drop in performance, reducing the overall F_1-Score for this prompt. It is not exactly clear what causes these differences. However, while acknowledging that the performance did drop somewhat, no stark differences in terms of expected GPT-4 classifications are seen in this run, once again showing that the phrasing and type of prompt has a larger impact on GPT-3.5 than on GPT-4 compared to the main prompt and dataset.

4.1.4 Run type: Sentiment choice

The concept of this run type is to instead of having GPT respond with a binary evaluation, it should assign posts a sentiment from a given list, which includes irony as one of the options. The goal of this run type is to further test irony detection when not specifically instructed to detect irony, similar to sub prompt "no detector" for the main run type. However, this run fully removes any indication that the experiment is about irony specifically. The full prompt is:

> *You are a sentiment detector. Assign posts a sentiment from the following list depending on which you consider most appropriate: angry, sad, ironic, happy, neutral, confused. Respond only with one word.*

While of course not every possible sentiment is given in such a short list, it is expected due to the phrasing that GPT will assign the *closest* sentiment, such that "frustrated" would fall within "angry", or "excited" within "happy". For the purpose of this run, any evaluation that is not "irony" will be counted as a classification of "0" or "non-irony".

4.1.4.1 Sentiment choice run main prompt

GPT-3.5

Run set gpt-3.5-sentChoice-default-main resulted in an average accuracy of 0.62, precision of 0.75, recall of 0.34 and F_1-Score of 0.47. Only 4 rows out of 10 runs on 100 rows (meaning 4 out of 1.000 evaluations in total) were errors, which is in line with most other run types, meaning the instructions did not cause GPT-3.5 any confusion as to which sentiments to assign. All errors are single sentiments that are not contained within the given list, such as *"hopeful"*. These results, especially in terms of F_1-Score show a stark difference to gpt-3.5-binary-default-main, with a strong decrease in both recall (by ~0.56) and F_1-Score (by ~0.23), putting them among the largest differences to the main run recorded in the entire experiment. However, while accuracy stayed the same at 0.62, precision actually increased by ~0.18. The increase in precision and stark drop in recall indicate that GPT-3.5 made more selective decisions, only labeling irony when very confident in that evaluation. Standard deviation shows no notable deviations from expected GPT-3.5 values.

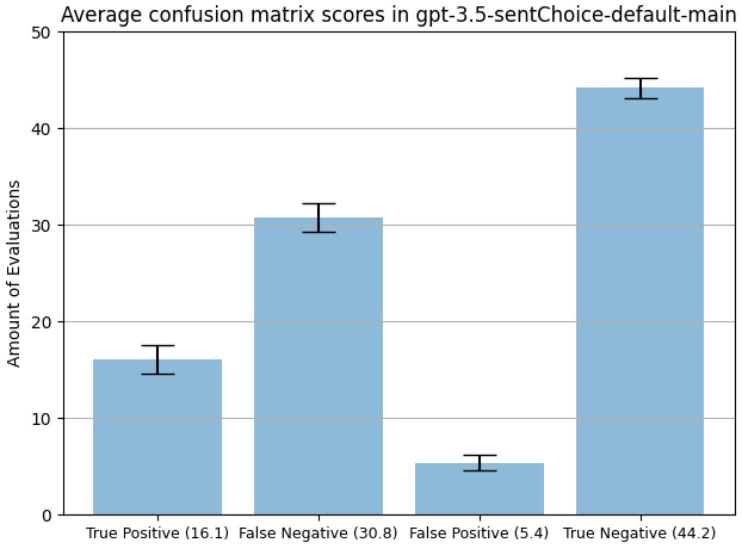

Figure 19: Average confusion matrix scores from run set gpt-3.5-sentChoice-default-main.

The average matrix scores also show very different results from the main prompt runs, with true positives being the second lowest score. True negatives have drastically increased, which is not unexpected given that every not "irony" classification is counted towards "non-irony". However, with the drastic increase in false negatives, there is an almost complete flip of scores in favor of non-irony detection. The scores from Figure 2 were 44.3 (tp), 4.7 (fn), 33.7 (fp), 17.3 (tn), which are a close to perfect mirror of these from Figure 19 scores in the other direction. This would indicate that instead of over-evaluating irony, GPT-3.5 now drastically under-evaluates irony, only giving the irony label when it is very obviously correct (which is backed up by the average scores from precision and recall, suggesting more selectiveness). Consistency compared to gpt-3.5-binary-default-main shows an increase of 9 in terms of overall consistency at 90 out of 100 rows, and also show the switch in scores seen in the matrix scores, albeit slightly less obviously. The main prompt run set had 45 consistently correct ironic and 7 consistently correct non-irony evaluations, while the sentiment choice run set had 12 and 43 respectively. A similar change can be seen in consistent incorrect rows at 1 ironic and 28 non-ironic for the main prompt run set and 29 ironic and 6 non-ironic for the sentiment choice run set. Of 10 contested rows, 8 were ironic and 2 non-ironic, whereas the main prompt run had 19 contested with 3 ironic and 16 non-ironic. While the absolute values are not a perfect match, the proportions and overall distribution support the idea of a reversing in terms of result scores. It is very possible that GPT-3.5 considering irony only as one of its choices and bundling all other sentiments to count as "non-ironic" created the exact effect seen in the main prompt run, where now the other sentiments made up the vast majority of GPT-3.5's possible evaluations, making what would count as the "non-ironic" category far larger and posts more likely to be evaluated as such, causing the overclassification in comparison to ironic lines.

GPT-4

The results from run set gpt-4-sentChoice-default-main showed an average accuracy of 0.77, precision of 0.83, recall of 0.67 and F_1-Score of 0.74. Barring a small decrease in precision by ~0.03 and an equally small increase in recall by ~0.03, accuracy and F_1-Score are the same as in gpt-4-binary-default-main, with no notable change in standard deviation for any score. Overall, especially given the drastic changes seen in GPT-3.5, this result is remarkably consistent with the default prompt despite completely different phrasing and classification instructions.

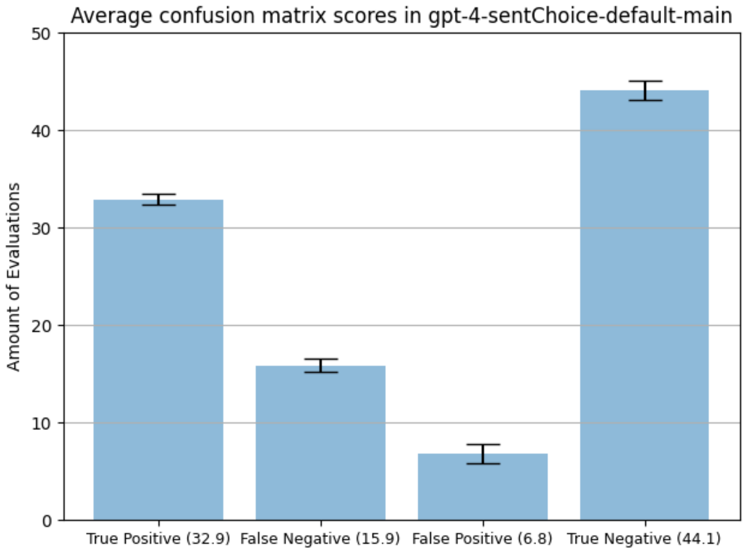

Figure 20: Average confusion matrix scores from run set gpt-4-sentChoice-default-main.

Just as with the aforementioned scores, the average matrix scores show no significant differences to those obtained in gpt-4-binary-default-main, except for a minute decrease in standard deviation in *tp*/*fn* from ~1.20 (both) to ~0.54 (tp) and ~0.7 (fn) (different due to 3 errors overall in the run set). Consistency, as expected from matrix results, shows no significant difference in any metric.

4.1.4.2 Insights from sentiment choice prompt experiments

GPT-3.5 showed entirely different results when using this prompt, almost entirely switching scores for actual irony and non-irony detection, once again showing a drastic change in results for GPT-3.5 when using a different prompt. GPT-4 on the other hand once again showed almost no differences from the main prompt run sets, despite receiving an entirely different task in the prompt. This result shows, once again, remarkable consistency for GPT-4 with great irony and non-irony detection overall, whereas most GPT-3.5 runs resulted in either good irony or good non-irony detection capabilities.

4.1.5 Insights from GPT run set experiments

GPT-3.5	Accuracy	Precision	Recall	F_1-Score	TP	FN	FP	TN
binary	0.62	0.57	**0.90**	**0.70**	44.3	4.7	33.7	17.3
confidence	0.54	0.51	0.74	**0.70**	**48.4**	0.6	**45.7**	5.3
percentage	**0.66**	0.65	0.65	0.65	35.8	12.9	17.4	33.0
sentiment choice	0.62	**0.75**	0.34	0.47	16.1	**30.8**	5.4	**44.2**

Table 4: The average values from the GPT-3.5 run sets seen in each run type analysis. Bold highlights the highest value of the column while red highlights the lowest value of the column.

Table 4 shows aggregated results from the different run types for GPT-3.5. Immediately noticeable are the comparatively good scores for the main run, with the highest F_1-Score and highest recall, as well as no lowest scores. Throughout all run types apart from sentiment choice (which is an outlier in almost all scores) the pattern of labeling most posts as ironic continues, making it a pattern not just in sub prompts of the main prompt, but for GPT-3.5 in general, which was further seen taken to an extreme in the confidence run type, which had the highest true and false positives of all recorded run sets, resulting in the lowest precision for the run types and, due to extremely low true negatives, also the lowest accuracy of all recorded run sets. The sentiment choice run type resulted in a surprising flip of values, as discussed in the respective section. This total flip of scores also resulted in the lowest recorded F$_1$-Score and recall of all run sets, while having the highest precision of all GPT-3.5 run sets. Overall, most of these results (barring the sentiment choice run type) generally fall in line with the expected GPT-3.5 behavior also seen in the sub prompts for the binary run type.

GPT-4	Accuracy	Precision	Recall	F$_1$-Score	TP	FN	FP	TN
binary	0.77	0.86	0.64	0.74	31.5	17.5	5.1	45.9
confidence	**0.79**	0.86	**0.68**	**0.76**	**33.1**	15.9	5.4	45.6
percentage	0.76	**0.91**	0.58	0.70	28.2	**20.8**	2.9	**47.1**
sentiment choice	0.77	0.83	0.67	0.74	32.9	15.9	**6.8**	44.1

Table 5: The average values from the GPT-4 run sets seen in each run type analysis. Bold highlights the highest value of the column while red highlights the lowest value of the column.

Table 4 shows aggregated results from the different run types for GPT-4. Once again, as for the sub prompts of the binary run type, scores are generally consistent with little deviation. Scores remain similar or the same across run types, with the confidence run type resulting on some of the best scores, contrary to GPT-3.5's confidence run set. The percentage run type saw some of the lowest scores, which are a result of its lowered irony detection, having the lowest true positives and highest false negatives of all run types. Overall, the GPT-4 results are all within generally the same ranges as the binary main prompt and sub prompts, further showing great consistency and confidence for GPT-4 for different prompts.

4.2 Other Large Language Models

As discussed earlier, two other irony detection models will be examined in this paper, namely the TweetNLP (TweetNLP, 2024) (Jose Camacho-Collados, 2022) and pysentimiento (pysentimiento, 2024) libraries for Python. Both of these libraries contain mechanisms and methods for multiple different NLP applications such as hate speech detection, emotion analysis or other sentiment detection tasks. Both of these models are based on a pretrained version of Google's BERT, called roBERTa (Yinhan Liu, 2019), with further (separate) pretraining done using tweets. The main dataset, tweet_eval_irony_train, is some of the same SemEval-2018 data used by TweetNLP and pysentimiento to train irony detection, and as such results in an unfair advantage (reflected in almost perfect scores) if comparatively used. Due to this, all comparisons done between LLMs will use the manual dataset formed from a different corpus of tweets so as to not give an advantage to any LLM. As TweetNLP and pysentimiento irony detection was trained on tweets, using a set of different tweets from training data is an appropriate way to maintain performance without unfair disadvantage. The main comparison will be between the models and the default prompt run sets on the manual dataset (Section

4.1.1.1) for GPT-3.5 and GPT-4, as the results are stable and display the abilities of each model.

4.2.1 Direct Comparison

TweetNLP and pysentimiento irony detection results are given in the form of a probability split into irony and non-irony which add up to 1. As such, analogous to the percentage run type examined in Section 4.1.3, if a result contains an irony evaluation of 0.5 or greater, it is counted as an irony classification for the purposes of this analysis. Unlike GPT as used in these experiments, TweetNLP and pysentimiento irony detection always returns the same values for the same inputs, thus eliminating the necessity for multiple runs to gather a coherent average and allowing the use of one run of the dataset for comparison.

	Accuracy	Precision	Recall	F_1-Score	TP	FN	FP	TN
GPT-3.5	0.59	0.55	**0.92**	0.69	**45.9**	4.1	**36.9**	13.1
GPT-4	**0.78**	**0.75**	0.84	**0.79**	41.9	8.1	14.0	**36.0**
TweetNLP	0.61	0.60	0.68	0.64	34.0	**16.0**	23.0	27.0
pysentimiento	0.71	0.66	0.86	0.75	43.0	7.0	22.0	28.0

Table 6: The values from GPT-3.5 and GPT-4 using the manual dataset as well as results obtained from the TweetNLP and pysentimiento libraries. Bold highlights the highest value of the column while red highlights the lowest value of the column.

Table 6 shows the comparative results in average scores from GPT and the scores given by TweetNLP and pysentimiento. Immediately noticeable is the high performance in most scores by GPT-4 compared to all other models. With the highest F_1-Score and Accuracy values as well has the best non-irony detection (as well as irony detection very close to the best results), it's clear that GPT-4 delivered the best performance, even compared to models specifically trained for tweet irony detection. Perhaps unsurprisingly, GPT-3.5 delivered some of the overall worst results, and its tendency to overlabel irony is not reflected within the other two models, as true negatives are higher than false positives in all other LLM results. With the lowest precision and accuracy, GPT-3.5 still ended up with the third best F_1-Score overall, due to high true positives.

TweetNLP's irony detection on the other hand resulted in the second worst accuracy, with only about 2/3 of all actual irony and about half of non-irony being detected as such, whereas all other models were upwards of 4/5 for actual irony. With the second to lowest precision and the by far lowest recall value, the model did not manage to sufficiently detect irony or non-irony in any notable capacity compared to some other models. As a result of these overall comparatively low scores, TweetNLP ended up with a predictably low F_1-Score of 0.64, which is on the lower end of all recorded run sets.

Pysentimiento's results are measurably better than TweetNLP's, showing better accuracy and precision due to more accurate irony detection at over 40 out of 50 correctly identified irony rows without sacrificing non-irony detection, which was similar to TweetNLP at a little more than half of actual non-irony correctly identified. Due to this comparative increase in true positives on-par with GPT-4's irony detection, pysentimiento ended up with a relatively high F_1-Score of 0.75.

4.2.2 Insights from direct comparison

Results show that GPT occupies both the lowest and the highest score places, with GPT-3.5's poor performance in correctly identifying actual irony and non-irony placing it lower than other models in most scores. GPT-4 on the other hand managed to outshine both libraries with great detection for both actual irony and non-irony, resulting in some of the highest scores (especially F_1-Score) of all recorded run sets. Both pretrained roBERTa models thus place in between the GPT models, with pysentimiento slightly outperforming TweetNLP due to better irony detection. An interesting note however is that while these two models showed differences in irony detection, their non-irony detection was almost exactly the same, potentially showing a difference in the training material or priorities in detection. It also might indicate progress in LLM performance over time, as out of all examined models, GPT-4 is the most recent model, being released in 2023, whereas both GPT-3.5 and roBERTa are older, with GPT-3.5 being released in 2022 but based on GPT-3 which was released in 2020. The pre-trained models from TweetNLP and pysentimiento were both released in 2022, possibly showing the advantage of an older model (roBERTa from 2019) being pretrained for specifically irony detection, a minute effect of this was also seen in GPT when using one-shot prompts, which increased scoring and irony detection. Thus, perhaps expectedly, even though GPT-3.5 and both libraries were first released in the same year, GPT-3.5 as a model not specifically trained for irony detection had worse overall results than TweetNLP and pysentimiento.

5. Future & Conclusion

5.1 Future

More experiments can be done using the frameworks discussed in Sections 3 and 4. Mainly, the experiments could be expanded to include run sets on all included datasets, such as the reddit set or the manual dataset which were only used in very few instances due to brevity. Additionally, datasets could be further altered and preprocessed to check if differences such as the removal of all hashtags, mentions or other linguistic features measurably changes the effect on results for each run type and sub prompt. In addition, since only tweets and reddit comments were used, valuable results could be obtained by acquiring more types of data, such as social media posts from different platforms or ironic/non-ironic statements from other sources. It could also be possible to create multiple different manual datasets from the same overall dataset with no overlaps between them and experiment with how the consistency and overall scores change between subsets of the same dataset. More run types could have been included, such as asking “Are you sure?” after an irony classification from GPT and recording/evaluating the responses. Since no run type except the main binary runs had any sub prompts, the same, similar and entirely different sub prompts could be created for each run type beyond binary in order to evaluate scoring differences between sub prompts. Comparisons could also be done between different prompts and run types beyond just comparisons to the main set by comparing different result prompt performances to one another. Future experiments may also include new GPT or other LLM versions, such as the eventual GPT-5 or other more advanced LLMs.

5.2 Conclusion

Valuable insights were gained during these experiments. For one, GPT-3.5 performed measurably worse than GPT-4 in almost every experiment and metric, showing a rudimentary to missing ability to separate irony from non-irony paired with sometimes very inconsistent scores, resulting in a need to increase set lengths from 10 to 20, which was never required for GPT-4. In addition, the sometimes massive changes in behavior when altering the prompt even slightly or between prompts indicates very poor internal consistency, leading to the conclusion that GPT-3.5 is not effective or fit for irony detection purposes on a general scale. GPT-4's decent to good performance on the other hand indicates that if this model was further pre-trained with irony and non-irony inputs, it is likely that, based on the consistent performance of GPT-4's basic model, such a fine-tuned LLM would excel at irony detection and consistently produce useful results. The other LLM's performances show limitations of the models they are based on, which are not as performant as GPT-4, but due to fine-tuning and pre-training still manage to outperform GPT-3.5. Overall, while these tools were decent, one can conclude they are not necessarily fit for general irony detection purposes and probably would benefit from using more modern, advanced models.

Table of Figures

Table of Tables

References

Aytekin, M. U. (19. January 2024). *Generative Pre-trained Transformer (GPT) Models for Irony Detection and Classification.* Von IEEEXplore: https://www.doi.org/10.1109/IISEC59749.2023.10391005 abgerufen

Barbiery, F., Comacho-Collados, J., Neves, L., & Espinosa-Anke, L. (2020). TweetEval: Unified Benchmark and Comparative Evaluation for Tweet Classifiaction. *Findings of the Association for Computational Linguistics: EMNLP 2020* (S. 1644--1650). Association for Computational Linguistics.

Barth, J. (17. June 2024). *bachelor.* Von GitHub: https://github.com/Jonas-Barth/bachelor abgerufen

Bushwick, S. (16. March 2023). *What the New GPT-4 AI Can Do*. Von Scientific American: https://www.scientificamerican.com/article/what-the-new-gpt-4-ai-can-do/ abgerufen

Heaven, W. D. (14. March 2023). *GPT-4 is bigger and better than ChatGPT—but OpenAI won't say why*. Von MIT Technology Review:

https://www.technologyreview.com/2023/03/14/1069823/gpt-4-is-bigger-and-better-chatgpt-openai/ abgerufen

International Workshop on Semantic Evaluation. (2018). *SemEval-2018 International Workshop on Semantic Evaluation*. Von SemEval-2018: https://alt.qcri.org/semeval2018/index.php?id=tasks abgerufen

John, N. (2020). *Tweets with Sarcasm and Irony.* Von Kaggle: https://www.kaggle.com/datasets/nikhiljohnk/tweets-with-sarcasm-and-irony/ abgerufen

Jose Camacho-Collados, K. R.-A.-C. (29. June 2022). *TweetNLP: Cutting-Edge Natural Language Processing for Social Media.* Von Arxiv: https://doi.org/10.48550/arXiv.2206.14774 abgerufen

Meer, D. V. (12. July 2024). *Number of ChatGPT Users and Key Stats (September 2024)*. Von NamePepper: https://www.namepepper.com/chatgpt-users abgerufen

Montgomery Gole, W.-P. N. (7. Dec 2023). *On Sarcasm Detection with OpenAI GPT-based Models.* Von arxiv: https://doi.org/10.48550/arXiv.2312.04642 abgerufen

OpenAI. (2024). GPT-4 Technical Report. *arxiv*, https://doi.org/10.48550/arXiv.2303.08774.

Piper, K. (13. Aug 2020). *GPT-3, explained: This new language AI is uncanny, funny — and a big deal*. Von Vox: https://www.vox.com/future-perfect/21355768/gpt-3-ai-openai-turing-test-language abgerufen

pysentimiento. (29. August 2024). *pysentimiento: A Python toolkit for Sentiment Analysis and Social NLP tasks*. Von github: https://github.com/pysentimiento/pysentimiento abgerufen

Quach, K. (14. 2 2019). *Roses are red, this is sublime: We fed OpenAI's latest chat bot a classic Reg headline*. Von The Register: https://www.theregister.com/2019/02/14/open_ai_language_bot/ abgerufen

Tatman, R. (2017). *Ironic Corpus.* Von Kaggle: https://www.kaggle.com/datasets/rtatman/ironic-corpus abgerufen

TweetNLP. (29. August 2024). *TweetNLP*. Von TweetNLP: https://www.tweetnlp.org/ abgerufen

Vincent, J. (14. Feb 2019). *OpenAI's new multitalented AI writes, translates, and slanders*. Von Vox: https://www.theverge.com/2019/2/14/18224704/ai-machine-learning-language-models-read-write-openai-gpt2 abgerufen

Yida Mu, B. P. (23. May 2023). *Navigating Prompt Complexity for Zero-Shot Classification: A Study of Large Language Models in Computational Social Science.* Von arxiv: https://doi.org/10.48550/arXiv.2305.14310 abgerufen

Yinhan Liu, M. O. (26. July 2019). *RoBERTa: A Robustly Optimized BERT Pretraining Approach.* Von Arxiv: https://doi.org/10.48550/arXiv.1907.11692 abgerufen