Evaluating the Performance of Open-Source LLMs in Sarcasm Detection: A Comparative Analysis

by

Samin Haque
21301628
Syed Shakib Ahmed
24141224
Kazi Fardin Ahmed
21301235
Julker Nayeen
22101120
Khandoker Hamidun Nabi
24141243

A thesis submitted to the Department of Computer Science and Engineering
in partial fulfillment of the requirements for the degree of
B.Sc. in Computer Science and Engineering

Department of Computer Science and Engineering
Brac University
October 2025.

Declaration

It is hereby declared that

1. The thesis submitted is my/our own original work while completing degree at Brac University.
2. The thesis does not contain material previously published or written by a third party, except where this is appropriately cited through full and accurate referencing.
3. The thesis does not contain material which has been accepted, or submitted, for any other degree or diploma at a university or other institution.
4. We have acknowledged all main sources of help.

Student's Full Name & Signature:

Samin Haque
21301628

Syed Shakib Ahmed
24141224

Kazi Fardin Ahmed
21301235

Julker Nayeen
22101120

Khandoker Hamidun Nabi
24141243

Approval

The thesis/project titled "Evaluating the Performance of Open-Source LLMs in Sarcasm Detection: A Comparative Analysis" submitted by

1. Samin Haque (21301628)
2. Syed Shakib Ahmed (24141224)
3. Kazi Fardin Ahmed (21301235)
4. Julker Nayeen (22101120)
5. Khandoker Hamidun Nabi (24141243)

of Fall, 2024 has been accepted as satisfactory in partial fulfillment of the requirement for the degree of B.Sc. in Computer Science on Summer, 2025.

Examining Committee:

Supervisor:
(Member)

Dr. Farig Yousuf Sadeque

Associate Professor
Department of Computer Science and Engineering
Brac University

Thesis Coordinator:
(Member)

Dr. Md. Golam Rabiul Alam

Professor
Department of Computer Science and Engineering
Brac University

Head of Department:
(Chair)

Dr. Sadia Hamid Kazi

Associate Professor & Chairperson
Department of Computer Science and Engineering
Brac University

Abstract

Detecting sarcasm is a nuanced and challenging task in natural language processing (NLP), as it requires understanding subtle contextual cues, symbolic gestures, and timeline of dataset as sarcasm constantly changes through time. This study tends to evaluate the performance of open-source large language models in sarcasm detection, exploring each model's effectiveness in identifying sarcasm across varied contexts. In doing so, it tries to apply various prompt engineering techniques towards establishing an optimized environment; where these open source models will detect sarcasm based on the hyper parameters like context, and the actual comment corresponding to the context. Among the evaluated models, Llama 3.1 8B achieved the highest overall accuracy (69.07%) while DeepSeek R1 14B demonstrated the most balanced performance (accuracy = 66%, precision = 64.29%, recall = 72%). On sarcasm-device level evaluation, the models showed varying strengths: Llama 3.1 8B performed best on Contextual Incongruity (63.15%), Mistral 7B on both Contradiction and Contextual Incongruity (50%), Gemma 3 12B on Contradiction (80%), Qwen 3 8B on Contradiction (50%), and DeepSeek R1 14B on Hyperbole (50%). The findings highlight that while current open-source LLMs exhibit promising capabilities in detecting sarcastic intent, their performance varies by sarcasm device, revealing architectural biases and limitations. This study thus provides benchmark insights and interpretive evidence for advancing future NLP research in sarcasm modeling, architectural evaluation, and pragmatic language understanding.

Keywords: Sarcasm Detection, Large Language Model (LLM), Open-Source, Natural Language Processing (NLP)

Acknowledgement

The thesis represents our own original work while pursuing degree at the Computer Science and Engineering Department, BRAC University.We extend our deepest gratitude to the Department of Computer Science and Engineering at BRAC University for providing us with the opportunity and resources to pursue this research. We are deeply grateful to our supervisor, Dr. Farig Yousuf Sadeque, for his invaluable guidance, continuous support and insightful feedback throughout the process. All sources of information are referenced appropriately and the views expressed in the paper are solely of the authors. As far as our knowledge, this work does not contain any material of any previously published or submitted works of elsewhere except where explicitly cited with proper references.

Table of Contents

List of Tables

Chapter 1

Introduction

In modern times, social media has emerged as the most significant platform for global communication, reshaping how people interact, share information and express themselves. In social media, people express their sentiments and communicate by messaging, texting, sharing their thoughts and expressing concerns through comments, discussing issues through articles- overall writing their thoughts through and through. But the sentiments in the written texts are not always straightforward as they seem, which is why it is important to understand if a message contains sarcasm or irony or not as it changes the whole meaning of a context into something different. It is also necessary to understand and detect the sarcastic meaning for it has become a general application of modern era social media users. Just like any language, with respect to the change of time, these sarcastic and ironic sentiments' form and meaning changes. Furthermore, words and languages adapt to changes, influenced by culture, social factors, popular trends, mass uses, religious and political influence etc. Words that may have had different meanings in previous years can have semantic shifts or gain new connotations through time (eg: "Gay" used to mean happy or carefree before the mid 20th century and now it is used for homosexuality). The rapid evolution of language on social media, driven by cultural trends, technological advancements and shifting societal norms, presents challenges in communication. One of the most pressing challenges is the potential for miscommunication, particularly as words and phrases acquire new meanings, are used sarcastically or are reappropriated by different communities. Detecting contextual sentiments and adapting to these changes with respect to time remains our primary goal. To achieve it,we are using open source LLMs to detect sarcasm and irony in contexts. Although multiple approaches and researches are conducted to detect nonliteral context, detecting sarcasm using semantic incongruity and literal word-context disambiguation, there are hardly any works or researches present that works with open source LLMs to detect the sarcasm of a text. A challenge still stands which is the ever changing nature of words,especially sarcastic and ironic words. Twitter, Reddit and Instagram is a good source of social data that can be used to pool sarcastic and non-sarcastic texts. Non-sarcastic texts have positive and negative sentiments. The sarcasm of a context can be detected using several methods like recognizing contextual incongruity,lexical structure, intensifiers (adjective and adverb), interjections, punctuations, capitalizations etc. Our first objective is to categorize texts into sarcastic and non-sarcastic categories using open source LLMs. Then we create a custom made model which is only defined by a certain

time period and separately another model containing both the timeline and context and feed the testing data to evaluate and compare the three results to conclude our findings into a fact.

1.1 Problem Statement

The study aims to conduct a comparative analysis of open-source large language models in sarcasm detection. As sarcasm is rapidly evolving, the ability to update with linguistic changes across diverse datasets is quite challenging and important. The goal of this research is to investigate and find creative solutions to contribute in improving the performance of these open-source models.

1.2 Objective

Data Extraction and Analysis:

- Analysing and narrowing down extracted datasets from the social media platform Reddit.
- Pre-processing the dataset to structure it with features like contextual incongruity, lexicals, intensifiers (e.g., adverbs, adjectives).

Integration and Evaluation of Models:

- Implement state-of-the-art open-source large language models such as Llama 3.1, Mistral, Gemma, Qwen and Deepseek R1.
- Focus on various "Prompt Engineering" techniques to properly ground the responses and optimize these generative models classification capabilities.
- Evaluate the performance of these models on various sarcasm devices (e.g., hyperbole, contradiction, feigned ignorance, contextual incongruity)

1.3 Methodology in Brief

Firstly, we collected some research papers related to our study that is sarcasm detection and then we thoroughly read these papers to extract the limitations of those papers so that we can work on that. According to these papers we wrote literature reviews and completed our pre- thesis 1 report writing. For pre-thesis 2, we will then collect data from reddit. Following this we will preprocess the data to make it ready for model training. After fine tuning the models we will look into the results. After finishing the result analysis we will make ourselves ready for the final defense. In the below work flow Blue is indicating pre-thesis 1, Orange is indicating pre-thesis 2 and Green is indicating Final defense.

Figure 1.3.1: Workplan gantt chart

1.4 Scopes and Challenges

The challenges are to collect the necessary data from sources (e.g., Twitter, Reddit, Instagram), pre-process the datasets into mandatory features, and fine-tune the LLMs to create the necessary custom models.

Chapter 2

Literature Review

2.1 Preliminaries

The related works on which we have researched are chronologically organized. Below are some important information provided to better understand our studies.

Definitions:

1. **Context Incongruity:** Semantic mismatch between an utterance and its surrounding context; leading to a discrepancy that conveys implicit meaning (e.g., sarcasm, irony, criticism).
2. **Utterance:** In these studies, utterances referred to individual statements within a conversational dialogue on platforms such as Twitter or Reddit.
3. **Inter-sentential incongruity:** Refers to incongruity between utterances within a discourse context.
4. **Verbal Irony:** A figure of speech where the speaker says something but it means something opposite, indicating sarcasm or humor.
5. **Context:** The surrounding text or conversation that helps to interpret the meaning of a statement.
6. **Negation:** The use of negative markers to reverse the meaning of a statement.
7. **Pragmatic Inference:** The process of deriving meaning from context outside the literal interpretation of words.
8. **Gold Standard:** A dataset or annotation considered highly accurate which often used as a benchmark for evaluation.

Concepts:

1. **Sarcasm Detection as Word Sense Disambiguation (WSD):** Treat sarcasm detection as a distinguishing factor between literal and sarcastic senses of specific target words such as 'love', 'brilliant', 'never', etc.
2. **Crowdsourcing and Human Annotation:** Leveraging Amazon Mechanical Turk (MTurk) to collect paraphrases of sarcastic tweets which enabled the identification of semantically opposite word pairs (e.g., "Love" $\longleftrightarrow$ "hate")

3. **Word Embeddings:** Employing word2vec (skip-gram/CBOW), GloVe, and WTMF to create dense vector representations to capture semantic nuances of words in sarcasm vs. literal contexts

4. **Kernel-Based Classification:** Integration of SVM with MVME and word embeddings (e.g., WTMF, word2vec, GloVe) creating a custom SVM kernel to classify sarcasm against literal senses.

5. **Evaluation Metrics:** Use of precision, recall, and F1 score to compare performance across models.

6. **Sentiment Analysis:** Identifying the emotional tone of a text.

7. **Overfitting:** A modelling error where a model performs well on training data but underperforms on unseen data.

8. **Attention Mechanism:** A technique in neural networks that focuses on the important parts of a data annotated by weights to improve performance.

9. **Hierarchical Attention:** A model structure applied with attention mechanisms at multiple levels.

10. **Lexicon-based Features:** Features derived from predefined list of words with specific meanings or sentiments

Theories:

1. **Word Embedding Theories:**

   (a) **word2vec:** Leverages skip-gram and CBOW (Continuous Bag of Words) architectures to learn dense vector representations by predicting neighboring words or the target word from context.

   (b) **GloVe:** Uses global co-occurrence statistics to generate embeddings, emphasizing the importance of word to word co-occurrence patterns.

   (c) **WTMF:** A matrix factorization method tailored for short texts and to handle sparse data by modeling ignored words

2. **Context Incongruity Theory (A Linguistic Theory):** Sarcasm arises from a mismatch between a statement and its context. (e.g., Explicit Incongruity and Implicit Incongruity)conversational dialogue on platforms such as Twitter or Reddit.

3. **Pragmatic Theory:** The study of the influence of context on the interpretation of meaning.

4. **Lexical Semantics:** The study of word meanings and their relationship used to identify the incongruity in a text.

Models: SVM, SVM (RBF Kernel), MVME, WTMF, Word2Vec, GloVe, PPMI, BERT, BiLSTM, LSTM, , Attention-based LSTM, Conditional LSTM, Naive Bayes, Logistic Regression

Timeline: The research articles related to sarcasm detection that we reviewed are from 2009 to 2021.

2.2 Review of Existing Research

Burfoot and Baldwin [1] introduced a novel approach in separating satirical news articles from genuine ones. They leveraged lexical and semantic features to address the challenge of identifying ironic content that mimics real news. The researchers constructed an imbalanced dataset that included 4,000 true newswire articles from the English Gigaword Corpus and 233 satirical articles that were sourced through Google queries to target satire-specific domains, e.g., The Onion. Additionally, they manually filtered and removed metadata and source identifiers to prevent trivial classification. The authors used SVM classifiers with binary and Bi-Normal Separation (BNS) feature scaling, which they augmented by headline-specific unigrams, profanity detection (Regexp::Common::profanity), slang frequency (Wiktionary), and semantic validity checks based on web queries for entity co-occurrence. BNS scaling with validity features achieved the highest F score (0.798), outperforming binary features (F=0.654) and baseline models (F=0.118). BNS with validity features demonstrated high precision (0.945) and improved recall (0.690). Although, recall remained suboptimal due to subtle satire requiring deeper contextual knowledge. Lexical features such as headlines and slangs, provided marginal gains; whereas semantic validity comparatively enhanced performance, highlighting the necessity of entity plausibility. However, the research's reliance on manual curation, dataset imbalance, and third-party web queries for validity imposes scalability and generalizability concerns. Despite having certain limitations, the authors successfully established a foundational framework for satire detection, building on the efficacy of BNS and semantic features over purely lexical approaches.

Muresan et al. [2] worked on the work of Davidov et al. (2010), who tried to identify sarcastic and non sarcastic utterances in Twitter and in Amazon product reviews. Then Muresan et al. worked on addressing the challenge of conveying positive and negative attitudes in distinguishing sarcastic tweets. While approaching this problem, he was inspired by looking at lexical features for identification of sarcasm in the work of Kreuz and Caucci (2007). He also looked for pragmatic features, like establishing a common ground between listener and speaker. The dataset they used is made by them. To construct the dataset they used twitter hashtags like #sarcasm, #angry, #happy, #sarcastic etc for sentiment understanding and then refined the dataset eliminating noise such as URLs, duplicates and non-english tweets. Finally, the resulting dataset is 900 tweets consisting in each of the three categories which are positive, negative and sarcastic. The authors investigated the utility of lexical features for identifying the sarcasm like, unigram and dictionary based word categories and pragmatic features like, emoticons and user interactions in machine learning models. Features including unigram presence, term frequency of the dictionary terms, and pragmatic markers were used to train and test two classifiers: Support Vector Machines with sequential minimal optimization (SMO) and Logistic Regression. The results reveal the fundamental challenge of sarcasm detection: three-way classification's best classification accuracy 57% for SMO with unigram features could only be achieved when sarcasm vs. non-sarcastic was split (accuracy 68.33%). Surprisingly, the human annotators did not fare significantly better than a machine approached, achieving accuracies from 59.44% to 66.85%.

In the proposed framework, pragmatic features such as emoticons and interaction markers were shown to be significant in sarcasm identification and often proved to be the strongest differentiators between sarcastic vs straightforward sentiments. For example, scans found over-the-top compliments or out-of-place emoticons like a smiley face trailing a negative Tweet and tagged them as being sarcastic. These features emphasized the reversed sentiment that is often implicit in sarcasm. The study goes on to explain that tweeters are also limited by the absence of adequate context and the shortness of tweets factors that hinder both human and machine performance. Consequently, it promotes further research that takes more contextual information, such as common ground and world knowledge, into account to achieve better sarcasm detection. "As sarcasm is an intricate linguistic phenomenon, it demands subtle computational models that can incorporate both lexical and contextual signals for effective detection.

Joshi et al. [3] investigated the use of a linguistic theory: context incongruity as the basis for sarcasm detection and also addressed the limitations of prior works that relied on rule-based and statistical approaches that used unigrams and pragmatic features (i.e. emoticons), hashtag-based sentiment. The authors argued that sarcasm arises from incongruity between a statement and its context; which they categorised into two incongruities: explicit (contradictions within overtly expressed sentiment words e.g., "I love being ignored") and implicit (covert contradictions requiring deeper interpretation e.g., "I love this paper so much that I made a dodgy bag out of it"). The study developed a sarcasm classifier that was evaluated on two text forms: tweets and discussion forum posts. The collected datasets were categorized into three categories: Tweet-A (Hashtag-Supervised Tweets), Tweet-B (Manually Labeled Tweets) and Discussion-A (From post from the Internet Argument Corpus, IAC). The size of Tweet-A was of 5208 tweets (4170 sarcastic, 1038 non-sarcastic) which they downloaded using the Twitter API based on hashtags like #sarcasm and #sarcastic for sarcastic tweets and #notsarcasm and #notsarcastic for non-sarcastic tweets. They curated this dataset by performing manual checkups to first remove hashtags from the tweets to treat them as labels and they removed incorrect tweets as well. Tweet-B was of size 2270 tweets (506 sarcastic, 1772 non-sarcastic) which was manually labeled by one of the prior works that the authors researched on. Finally, Discussion-A was of size 1502 discussion forum posts (752 sarcastic, 750 non-sarcastic) which was also manually annotated and extracted from IAC. From this dataset they concatenated "elicitor post" (previous post in the thread) with the target post to capture broader context for inter-sentimental incongruity. However this was only feasible for 15% of the posts due to data sparsity. In addition to these the authors also used a separate dataset of 4000 tweets (50% sarcastic) to extract implicit incongruity features using an iterative algorithm (which the authors modified from a rule-based algorithm used in prior works). The researchers used Support Vector Machines (SVMs) as the primary model for sarcasm detection. Additionally, they employed LibSVM with a Radial Basis Function (RBF) kernel for training and evaluation. The features lexical, pragmatic, explicit and implicit incongruity were used to train this model. The feature engineering, choice of model and modifications in classification algorithms of the prior works allowed the researchers of this paper to achieve F-scores of 0.8876 (Tweet-A), 0.61 (Tweet-B) and 0.640 (Discussion-A)

outperforming rule-based baselines (used on prior works that the authors used for comparison) by 10-20%. In addition to that by integrating elicitor posts and using incongruity features to capture inter-sentential incongruity, the authors observed an improvement of 21.6% in precision for the dataset Discussion-A (discussion forum posts) but got a low recall due to data sparsity. The researchers concluded highlighting the key strengths of their research that included success in detecting sarcasm from both short (tweets) and long (forum posts) and also mentioned exploring more robust incorporation of inter-sentential incongruity in sarcasm detection which was challenging for this paper due to data sparsity.

Ghosh et al. [4] presents a novel approach to sarcasm detection by reframing it as a word sense disambiguation (WSD) task where the sense of word is either literal or sarcastic. They reframed it as Literal/Sarcastic Sense Disambiguation (LSSD) that aims to classify whether a target word in a given context conveys its literal or sarcastic meaning. The authors first collected a dataset of 70 target words (e.g. "love", "brilliant", "never") i.e. words that exhibit both meanings (literal or sarcastic) depending on the context. The curation process of datasets required downloading more than 2.5 million tweets via Twitter API and crowdsourcing with Amazon Mechanical Turk (Mturk). The crowdworkers rephrased sarcastic tweets (i.e. from #sarcasm hashtags) to their intended meanings. Then they aligned these pairs using unsupervised methods like co-training and SMT alignment to find semantically opposite words which resulted in 70 target words. The collected data included tweets labeled with hashtags (e.g., #sarcasm, #sarcastic) which they referred as S, tweets that are sarcastic but without sarcasm related hashtags, referring those as L and tweets labeled with sentiment-related hashtags (e.g., #happy, #sad) that they are referred to as Lsent. The dataset was split into 80% for training, 10% for development and 10% for testing with a minimum of 400 instances per sense. This finally yielded opposite-word pairs (e.g., "love" $\longleftrightarrow$ "hate") of total 37 target words. Additionally the dataset underwent several preprocessing steps: tokenization using CMU, lowercasing words (except the capitalized ones used for emphasis) and removing target words and hashtags to prevent bias. The study evaluated two approaches: distributional methods and classification methods. The distributional methods included positive pointwise mutual information (PPMI) baseline and word embeddings (e.g., WTMF, word2vec skip-gram and CBOW, GloVe) combined with modified MVME algorithm to compute context similarity. And for classification methods the researchers used Support Vector Machines (SVMs) with a custom kernel integrating word embeddings via MVME and three more word embeddings (WTMF, word2vec, GloVe) which is similar to the distributional method discussed above. These approaches showed that word embeddings outperformed all traditional baselines achieving 96-97% F1 score for S vs L classification and 83-84% F1 for S vs Lsent. With these improvements the researchers successfully indicated that word embeddings were highly effective in terms of words with distinct literal and sarcastic contexts. However, the research did have some limitations, those are, due to the extraction of initial 70 target words from unsupervised alignment techniques the authors had to work with 37 target words leaving remaining words with insufficient training data and also as the research used hashtags as labels; this could lead to noisy data as tweets can be sarcastic but may not include hashtags like #sarcasm or

#sarcastic. So the authors planned their future works to include expanding target words by refining semantic alignment techniques and addressing noise in hashtag-based annotations.

Muresan et al [5] tried to identify sarcasm by identifying nonliteral language in social media. They categorized tweets into 3 parts : Sarcastic, Positive and Negative as a primary goal to find lexical or pragmatic features that distinguish sarcastic sentiments from non-sarcastic utterances. Firstly, they created a corpus containing 900 tweets of each category using Twitter API and hashtags referencing targeted sentiments and considered all of them as non-context tweets. They categorized the three classes into sarcastic S, positive P and negative N. To identify positive and negative tweets, seed words from LIWC (2001) and WordNet-Affect (2004) were used. Retweets, duplicates, quotes, spam and other language tweets were excluded in preprocessing. They used 3 machine-learning algorithms : SVM, Naive Bayes and Logistic Regression and implemented lexical features (n-grams and lexicon-based) ,pragmatic features (emoticons,common ground) and a list of interjections and punctuations. All tweets were converted into feature vector representation with a total feature of 80,among them 10 features were mostly used.The respective percentage of emoticons present were- 3.1% smiley and 1.6% frowning for sarcastic, 11.1% smiley and 0.5% frowning for positive and 0.2% smiley and 5.2% frowning for negative tweets in the corpus.Only 28.7% of sarcastic tweets included a user as recipient of the message while 9.9% of positive and 17.9% of negative tweets were addressed to other users, indicating the possible importance of features directing common ground in sarcasm identification. Three human judges were asked to annotate the 10% of data as evaluation where 90% data were tested through machine-learning algorithms. In the first study, the data was to be categorized into respective 3 categories: S,P and N classes. The human judges achieved 62,59% accuracy where only 135 of 270 tweets were agreed by both of them as the same category, resulting in the gold standard test set falling into 43.33%. Then the data was divided into T1 and T2 sets where T1 was the whole 270 tweets and T2 was the 135 tweets the judges agreed on. For T1, SVM model achieved 59.3% accuracy where 59.90% accuracy was achieved using LogReg for T2. In the second study, instead of three classification, the authors conducted two classification tasks labeling tweets sarcastic and non-sarcastic. Human judges achieved accuracy of 66.85% with 0.37% uncertainty. The gold standard test set fell to 59.44% considering the tweets all judges agreed on (129 out of 180 tweets). The data was divided into T3(180 tweets) and T4 (129) set and all the machine-learning algorithms achieved higher accuracy for T3 (67.78% NB, 71.67% SVM,68.90% LogReg) and T4 (68.80% accuracy).In third study, only tweets with emoticons were used, containing 50 sarcastic and 50 non-sarcastic tweets (25 positive.5 negative). Human judges accuracy was 73% with uncertainty of 10%. The accuracy on the set they agreed on was 83% (83 of 100 tweets). Two datasets were created T5 (100 tweets) and T6 (83 tweets). For T5, SVM achieved 71% accuracy and 83.13% was achieved using SVM for T6 which is significantly lower than the accuracy of human judges. The end results show that to improve sarcasm identification, the context of the utterance needs to be taken into account as well as the context necessary to clarify the intention to convey sarcasm.

Ghosh et al [6] attempted to show that the conversational context is a noteworthy factor for detecting sarcasm in online social media and communication. Firstly, they collected data from Discussion Forums and Twitter. For Discussion Forums, the data was taken from Sarcasm Corpus V2 which used crowdsourcing methods to classify the dataset into two categories- sarcastic and non-sarcastic. The sarcastic discussions were gold labeled when majority annotators labeled them as sarcasm, thus resulting in a total of 4,692 data of balanced sarcasm and non-sarcasm category. For Twitter, data were pooled containing hashtags like #sarcasm, #sarcastic, #irony etc to retrieve essential data. Data were excluded like retweets, duplicates, quotes, tweets that contain only hashtags, URLs, sentences shorter than 3 words and hashtags that were not used at the very end of the tweets. From Twitter, 200,000 data was collected which was reduced by 13% as they were reply of another tweet and through the preprocessing of data, the final number of data were 25,991 where 12,215 sarcastic and 13,776 non-sarcastic data were finalized. In computation, the data were split into categories such as sarcastic S, non- sarcastic as NS, reply in isolation in first classification, considering both reply and context in second classification. Two computational methods were used: Support Vector Machines with linguistically-motivated discrete features and LSTM networks. In SVM, different features were used like bag of words that are derived from unigram, bigram and trigram representation of words, lexicon-based features. Linguistic Inquiry and Word Count (LIWC) was used to identify pragmatic features where Boolean technique is used to separate a LIWC category. "MPQA" and "Opinion Lexicons" were used to model utterance sentiment. Counting negative and positive sentiment tokens, negations and boolean features gives the end result of which sentiment a data belongs to. To identify sarcasm, morpho-syntactic features such interjections , tag questions , exclamation marks, typographic features like capital words,quotation marks, emoticons, tropes like superlative and intensifier words were used. Two LSTM networks were used by the authors, one is Attention-based LSTM Networks and second is Conditional LSTM Networks. In Attention-based networks, sentence level attention was used as it helps to highlight the sentence which triggers the whole data into a sarcastic category and it helps to reduce overfitting. The authors structured the network in a way where the context is read by an LSTM and the response is read by another LSTM and average word embeddings is used for sentence representation. In the Conditional network, two separate LSTMs are used.In Conditional Network, although the structure is the same as the attention-based network, the LSTM which reads the reply is conditioned on the representation of LSTM that reads the context that is built on the context.In computational model, the data were split into 90%-10% keeping balanced distribution for sarcastic and non-sarcastic data. In the parameters and word vectors training, 100-dimension skip-gram was used for twitter based on 2.5 million tweets and 300-dimension n-gram word2vec pretrained model was used for discussion forums. Instead of optimizing, a dropout rate of 0.5 and L2 regularization was used to evaluate and configure on the test set. The minibatch size was 16 for both dataset. For discussion forums, the SVM models underperformed on discrete features and adding context worsened it even more. The LSTM models improve the result for both context and reply. The highest improvement is achieved by the conditional LSTM network (73.32% F1 for S class and 7.56% F1 for NS) along with a 6% and 3% increment for S and NS classes respectively. For sentence

level attention-based LSTM model, F1 score for S class is 73.7%. The recall score for NS class drops for sentence level attention only on reply. Therefore, they experimented with a hierarchical attention model and secured an F1 score of 69.88% for S class. For Twitter, SVM does not perform very well. But for attention-based and conditional LSTM networks, the results are significantly improved. In conditional LSTM. 11% and 4-5% improvement on F1 score of S and NS class were achieved whereas attention-based secured 2% improvement on F1 score for both classes. Ultimately, LSTM networks worked better on both dataset rather than SVM models with better results achieved.

Ghosh and Muresan [7] targeted the social media irony detection automation problem by examining theoretically well-based irony markers that reveal ironic intent. Earlier computational detection methods integrated surface-level linguistic features but lacked a justified theoretical basis that lacked consistency with accepted linguistic theories. Researchers have filled this gap by following a structured assessment process of Attardo's (2000) irony frameworks to identify irony factors as well as optional irony markers. The authors investigate platform-specific communication styles by studying how the markers operate between Twitter and Reddit due to context-aware model requirements. The study drew from two extensive datasets that included 350,000 tweets. Sentiment hashtags like #happy or #sad classified non-ironic content but ironic content received tags through hashtags like #irony or #sarcasm. The preprocessing stage removed retweets along with English tweets and situations when the use of irony tags was ambiguous. Khodak et al. (2018) compiled the Reddit dataset using 50,000 posts containing "/s" markers to identify sarcasm alongside non-sarcastic threaded replies. The researchers excluded brief posts which lacked context by implementing an automatic filter ensuring posts maintained at least two sentences of text. A split-partitioning scheme divided the datasets for a comprehensive evaluation of model performance into partitions like training, development and test. Methodologically, the study employed a Support Vector Machine (SVM) classifier with a linear kernel, trained on three categories of irony markers: The research utilizes different categories of markers which include tropes (metaphors and hyperboles), morpho-syntactic patterns (exclamation marks and tag questions) and typographic elements (emojis and capitalization). Components of metaphors were retrieved from pre-existing lists and digital detectors while hyperboles became accessible from the MPQA sentiment database. Roland and Robinson analyzed rhetorical questions and interjections along with emoticons as examples of morpho-syntactic features yet integrated emoticons and emojis as typographic features which they categorized through sentiment analysis. The research team ran ablation tests that showed specific dependencies among each marker category based on each platform type. For Twitter, typographic markers like angry emojis and sarcastic emoticons (e.g., ":Marking characteristics including emoji ":P " demonstrated the highest discrimination factor leading to an F1 score of 71.75% for identifying ironic tweets. These morpho-syntactic markers consisted of exclamations and tag questions which enabled Reddit to achieve 58.35% F1 accuracy for identifying ironic content. The study's direct verification of platform-specific irony expressions is what makes it successful. Because Twitter is a visual-centric communication platform and emojis and emoticons are obvious irony signals, typographic markers took over there. Par-

ticularly in controversial subreddits like politics or religion, Reddit showed a greater reliance on linguistic structures like arguments and hyperboles, as it favored longer, text-rich discussions. These disparities were further emphasized by frequency analysis: exclamation points were used in 42% of political Reddit posts compared to 9% in technological in nature threads, while emojis were present in 5% of ironic tweets but were hardly used in Reddit. Additionally, the authors showed that connecting theoretically informed features increased detection accuracy, surpassing models that failed to take platform context into consideration. The study admits its drawbacks despite its contributions, including its reliance on self-annotated hashtags, which might ignore implicit irony, and its solely in English focus, which restricts cross-linguistic generalization. In order to capture subtle sarcasm, future research could investigate complicated architectures like transformers, integrate multimodal features such as images in tweets, or extend to multilingual datasets. With implications for sentiment analysis, conversational artificial intelligence, and social media moderation tools, this research offers a strong framework to improve irony detection systems by outlining platform details and strongly establishing computational methods in linguistic theory.

Ghosh et al. [8] explores the limitations of prior work that they researched on, which analyzes utterances in terms of detecting sarcasm; limitations being these utterances in isolation, overlooking the conversational contexts. The authors' investigation yielded three key problems: whether (i) modeling conversation context i.e. both preceding or succeeding turns improve detection, (ii) recognizing context segments triggering sarcasm, and (iii) localizing sarcastic sentences within multi-sentence posts. The researchers collected their datasets from Twitter (self-labeled via hashtags), Reddit (self-labeled with "/s"), and crowdsourced annotated datasets from Internet Argument Corpus (IACv2). The preprocessing steps involved removing duplicates, non-English content, and truncation of lengthy posts. They made a comparative analysis between SVM baselines (lexical, sentiment, and sarcasm-marker features) against LSTM variants including simple LSTMs, attention-based models (word/sentence-level), and conditional LSTMs that encode context-reply relationships. The architecture separated contexts in preceding or succeeding turns and current turns into different LSTMs, with attention mechanisms that highlighted critical sentences. Achieving F1 scores of 73.32% (IACv2) and 76.30% (Twitter) for sarcastic classes, conditional LSTMs and sentence-level attention models outperformed baselines. With F1 scores between 78% to 84%, context-aware models consistently surpassed context-free ones. Again, multiple-LSTMs with F1 score of 83.92% on IACv2+ outperformed concatenated-input LSTMs. However, a drop of 10% on F1 score was observed on crowdsourced IACv2 due to training on self-labeled Reddit data; highlighting annotation-method biases. While attention weights partially aligned with Turkers' annotations where 41% overlap in trigger identification occurred, the authors faced challenges in handling background knowledge, profanity, and multi-sentence sarcasm. The authors concluded by emphasizing the value of context-aware architectures in sarcasm detection, and also called for domain-specific adaptations and richer contextual modeling to address remaining limitations.

Ghosh [9] in his comprehensive and empirical study on verbal irony, addressed three main problems: (i) automatic identification of verbal irony, (ii) interpretation of verbal irony, and (iii) its role in detecting agreement or disagreement relations in online discussion forums. The incongruity between literal and intended meanings is a challenging task for natural language understanding (NLU) systems; particularly social media where brevity, context dependency, and implicit sentiment are prevalent. The researcher deals with these issues through a mix of theoretical frameworks such as irony markers and factors from prior works; and computational models, leveraging datasets from Twitter that are self-labeled via hashtags like #sarcasm and #irony, leveraging datasets from Reddit that are self-annotated with "/s" markers, and finally the crowdsourced labels from the Internet Argument Corpus (IAC). Preprocessing procedures included removal of duplication and non-English tweets using Textblob; managing of typographic markers such as emoticons, emojis, and elongated words. For the identification of irony, the author employed baseline models like SVMs with discrete features such as n-grams, LIWC lexicons, irony markers like hyperbole and rhetorical questions; along with achieving F1 scores of 66.10% to 68.7% on Twitter and 58.35% to 70.35% on Reddit. A dual-CNN model containing advanced methods integrated word embeddings such as Word2Vec and GloVe; and deep learning architectures was used to explicitly model incongruity between sentiment words like "love" and negative contexts like "hospital"; which achieved a F1 score of 89.0% in literal or ironic sense disambiguation (LISD) tasks. LSTMs and attention mechanisms were used for context-aware irony detection. The conditional LSTM; initializing current-turn LSTM states with prior-turn context, achieved better results than baseline LSTMs, scoring a result of 76.30% F1 on Twitter and 73.32% F1 on IAC. Meanwhile, sentence-level attention LSTMs recognized context sentences that triggers irony such as semantic incongruence between prior and current turns. On the contrary due to limited training data; hierarchical attention models that are at word and sentence level underperformed, specifically achieving 69.88% F1 on IAC. Crowdsourced rephrasing experiments showed linguistic strategies like negation and antonym substitution for interpretation, with computational models achieving 84.9% accuracy in strategy classification. There was an improvement in detecting agreement/disagreement in IAC by 5% F1, using irony-based features in argument mining which demonstrated practical utility. The authors found that several adversities such as reliance on noisy self-labeled data like hashtag ambiguity, platform-specific feature generalization like Reddit's typographic markers vs. Twitter's emojis, and scalability of attention mechanisms for long contexts arises; while they studied to find a successful way to integrate theoretical insights such as irony markers, with the help of deep learning. Although there remains a challenge of cross-domain adaptability and real-time processing, the dual-CNN and conditional LSTM architectures appeared as robust solutions.

Ghost et al [10] introduced an idea to detect verbal irony using connection to the type of semantic incongruity. They use the idea of incongruity between literal evaluation and the context to link verbal ironies with a text or data. To begin with, they collected data from Twitter using hashtags like #irony,#sarcastic,#sarcasm which resulted in a total of 1,000 tweets. They use crowdsourcing methods to distinguish the category of irony. For a given message five MTurks (Amazon Mechanical Turk)

were asked to detect sarcastic senses . The definition of verbal irony was taken from the Merriam-Webster dictionary which was used to train the Turkers. A dataset of 5,000 message-sense pairs was obtained where 184 Turkers labeled 5% of pairs as invalid thus making the final dataset contain 4,762 pairs of data. Text containing incongruity via hashtags were considered explicit irony and texts containing terms need to be reconstructed from context were considered implicit irony. Using two expert annotators, among 1,000 ironic messages 38.7% were explicit and 61.3% were implicit ironies with the inter-annotator agreement k=0.7. The dataset was shortened into 500 pairs named dev set and were annotated by a linguist expert using techniques like lexical and phrasal antonyms, negation, weakening the intensity of sentiment, interrogative to declarative transformation, pragmatic inference etc. They ran a test on both the message-sign dataset and SIGN dataset using two annotators who used various linguist techniques. Lexical antonym pair achieved 92.2% F1 on dev dataset. Using a total of 30 negation markers, they try to detect if they appear in either the message or the sense. For both these techniques, they remove the intensifiers like adjectives and adverbs. To detect, rhetorical type questions as a form of irony, they collect 4,000 tweets containing #sarcasm, #irony and "?" and 4,000 information seeking tweets containing "?" and run SVM RBF Kernel achieving 58.6% F1. They also use desiderative constructions with regular expressions and phrasal antonyms and pragmatic inference using language models. For all the strategies used on both message-sign test dataset and SIGN dataset, 90% F1 score was achieved except for AntPhrase+PragInf. The strategy distribution was also similar for both dataset.They concluded that Turkers mostly adopted lexical antonym and negation strategies to detect irony and interpretation are correlated whether ironic messages are implicit or explicit.

Chakrabarty et al. [11] examine in their paper "$\mathbf{R^3}$: Reverse, Retrieve, and Rank for Sarcasm Generation with Commonsense Knowledge" how to produce automatic sarcasms while focusing on the key components of valence reversal and semantic incongruity which require shared commonsense knowledge. The authors demonstrate that unifying contextual incongruity with sentiment reversal through commonsense reasoning enhances the quality of generated sarcastic text. $\mathbf{R^3}$ emerges as an unsupervised framework that utilizes a retrieve-and-edit approach to develop sarcastic statements from input texts without sarcasm which benefits both conversational AI and creative content creation systems. The study addresses the scarcity of labeled data for sarcasm generation, while also the complexity of this task involves several aspects of sarcasm. Unfortunately, existing methods were either based on a rule based system or supervised model that needed parallel data, making it infeasible for the large scale deployment. Two existing sarcasm datasets were merged by the authors: 1) 4,762 sarcastic-crowdsourced interpretation pairs from Ghosh et al. (2020) and 2) 3,000 sarcastic tweets from Peled and Reichart (2017). 150 of these 150 non-sarcastic utterances were selected for testing, and they were ensured to be lexically diverse. For commonsense knowledge generation, ConceptNet was an external resource included, for training contradiction detection models, the Multi Genre NLI Corpus was used, and as a retrieval corpus for context sentences, Sentencedict.com was a resource that we used. The $\mathbf{R^3}$ is a framework that integrates three modules, including Reversal of Valence that finds negative evaluative words

(captured from SentiWordNet) and reverses them using antonyms (from WordNet) and negation removal, for example "not fun" to "fun"; Utilizing COMET, a GPT trained on ConceptNet, this method uses it to associate causal common sense concepts (e.g.: fast food → stomach ache). Sentences with these concepts which are relevant are fetched from Sentencedict.com and are edited for grammatical consistency. Semantic Incongruity Ranking: The most semantically incongruous pair is obtained by Ranking with RoBERTa-large Fine tuned on MultiNLI to score contradiction between valence reversed sentence and retrieved context. Then, the full model (FM) is compared to ablations (e.g., valence reversal only) and the state of the art MTS2019 system (Mishra et al., 2019) that renews the learning without availing common sense integration. A three module pipeline is used to generate sarcastic utterances from non sarcastic inputs in the R3 framework with special concern for valence reversal and the semantic incongruity. Reversal of Valence is the first module which would identify words with negative evaluative patterns and replace them with their antonyms with WordNet or remove negation markers (not fun becomes fun). It is essential to make the quality that the generated utterance flips the literal sentiment, a principal aspect of sarcastic character. The second module implements Commonsense Context Retrieval, which uses a GPT based model, COMET, which has been fine tuned on ConceptNet to retrieve causal Commonsense concepts from the input (e.g., fast food causes stomach ache). Then they are used to retrieve relevant context sentences from Sentencedict.com and then edited to be grammatically consistent. The third module, Semantic Incongruity Ranking, utilizes RoBERTa-large fine-tuned on the Multi Genre NLI Corpus and uses it to figure out how to discover contradiction between the valence reverted sentence and the context retrieved. It selects the context with the highest contradiction score, and generated sarcasm should contain semantic incongruity. With this combination of modules, the framework generates sarcastic utterances which are not only sentiment reversed but also contextually non congruous, improving their quality and humor. To test the contribution of each component, ablations (c.g., valence reversal only) are tested on the full model (FM) which integrates all three modules. $\mathbf{R^3}$ was superior to human evaluations on Amazon Mechanical Turk (900 utterances rated on creativity, sarcasticness and humor, and grammaticality). Across all of the pairwise comparisons, the full model beat MTS2019 by 90%, and our model matched or bested human-generated sarcasm in the creativity and humor aspects (48% win rate with respect to humans). To pick one as an example, the input "I hate getting sick from fast food" was added to "I love getting sick from fast food. Coming in also as more sarcastic and humorous was 'stomach ache' (human-authored examples only), which is just 'an additional side effect.' Results of ablation studies demonstrated that valence reversal combined with commonsense context and incongruity ranking lead to the best results, and Pearson correlations between the contradiction scores and human ratings demonstrated the role of semantic incongruity. However, there were limitations in the form of occasional dependence on COMET's accuracy, though the work lay the foundation for unsupervised sarcasm generation based on creativity and contextual relevance.

Baruah et al. [12] explored the effectiveness of incorporating conversational context in sarcasm detection in online conversations using deep learning models. Their

primary choice of model was BERT but additionally they used BiLSTM and SVM model to produce a comparative analysis against BERT. The datasets were collected from Twitter and Reddit where both contained conversational dialogues. The final responses among these datasets were needed to be classified as sarcastic or not-sarcastic. Both the datasets were balanced (50% sarcastic and 50% non-sarcastic). The authors prepared their datasets into 5 variations (in terms of "final response") to perform training and testing the models mentioned above. The conversational variations were: a) responses without any context, b) response along with last utterance, response with c) two utterances, d) three utterances, and e) response with all utterances. Among the collected datasets, 5000 dialogues (as training set), 1800 dialogues (as test set) was from Twitter where the utterances count per dialogues included 2 to 20 (train) and 2 to 18 (test). Again 4400 dialogues (as training set), 1800 dialogues (as test set) was from Reddit where the utterance count per dialogues were 2 to 8 (train) and 2 to 13 (test). Preprocessing these datasets involved removing "USER" and "URL" tokens, lowercasing the text, concatenating target responses with those 5 varied utterances and preceding the utterances in reverse order. For the study the authors used uncased large versions of BERT which they fine-tuned with the Adam optimizer at a learning rate 2e-5. For the BiLSTM the preprocessed text was represented using pre-trained fastText embeddings. Finally, SVM was trained using TF-IDF features from character n-grams (1 - 6). In addition to these configurations the researchers cross validated the models by retaining 20% of the train set as a development set which had its demerits also as the models did not have 20% of the instances from the train set. To resolve this they performed 5-Fold cross validation. Overall performance of BERT's was superior then other two models achieving highest F1-score of 0.743 for conversational context variance with "response plus last utterance" and for sequence length of 140. Although the F1-score was downgraded to 0.500 and 0.334 for variations of "response plus last two and three" utterances respectively, it made up for the variation "response plus all utterances" achieving F1-score of 0.734, which indicated a substantial increase. Whereas, BiLSTM and SVM achieved F1-score of 0.672 and 0.676 respectively. These results as mentioned above were from the Twitter test dataset. For Reddit, the best performance of BERT's was an F1-score of 0.658 for the contextual variation "response without context" but showed a similar behavior that of Twitter test dataset by performing low for "response with two and three utterances" with F1-scores 0.478 and 0.334 respectively and again with an increase in performance "response plus all utterances".

There is a paper in this domain, namely the paper "A Report on the 2020 Sarcasm Detection Shared Task" by Ghosh et al. [13] which sheds a critical light on how the problem of sarcasm detection in natural language processing (NLP) can be solved by using the contextual knowledge. Sarcasm, a type of disparity between literal and intended meaning, is a challenge for automated systems due to lack of enough cues for reliable detection of isolated utterances. Basically, most of the prior work focused on a single turn text analysis, but according to authors, prior dialogue turns are crucial to disambiguate sarcastic intent. Under ACL 2020 Figurative Language Processing Workshop, the shared task seeks to benchmark state-of-the art models using conversation context on the two social media platforms: Twitter and

Reddit. There was a distinct bias towards using transformer architectures in the task at hand, with BERT, ROBERTa, and ALBERT receiving the most number of submissions. Different models were previously trained and then developed for their target use through a variety of techniques aimed at enhancing the conversational aspect, such as mixing previous dialogues together with the final response or using hierarchical attention methods. The winning system, microblog, most notably combined BERT with BiLSTM layers and NeXtVLAD pooling to use unlabelled data and pseudo-labeling at the same time using BERT's next-sentence prediction. Other approaches attempted to consider the previous context as an "aspect" of the context in aspect sentiment classification (e.g., LCF-BERT) or supplemented several transformers to enhance the varied context while trying to satisfy the multi-parameters. The shared task attracted 655 submissions for Reddit and 1,070 for Twitter, with microblog achieving top results including F1 score which was 0.834 for Reddit, and 0.931 for Twitter. The key components of Context ensembling and data augmentation helped the team achieve victories in F1 by outperforming competing systems by 7% to 14%. Nevertheless, on models that analyzed deeper context (3+ turns), diminishing returns suggested that there is a different appropriate context length for different platforms. Despite the progress, several limitations emerged. Systems struggled with sarcasm reliant on subtle humor or positive sentiment markers (e.g., "haha" or emoticons), suggesting a gap in detecting witty or contextually implicit sarcasm. Additionally, self-annotated hashtags introduced potential label noise, particularly in Twitter data, where sarcastic tweets might inadvertently use sentiment hashtags. Generalization remained challenging for niche topics (e.g., sports or politics), indicating domain adaptation as a critical hurdle. Future research could address these gaps by integrating multimodal cues (e.g., visual or acoustic signals in social media posts) to disambiguate intent, or adopting few-shot learning techniques to handle rare sarcasm types. Improving pseudo-labeling strategies for unlabeled conversational data and refining dynamic context aggregation mechanisms—such as adaptive attention over variable-length threads—could further enhance robustness. Lastly, developing frameworks to disentangle author-specific sarcastic styles from contextual patterns may improve cross-domain generalization, paving the way for more human-like figurative language understanding. It remained challenging for topics like sports or politics. Future research can bridge these gaps by combining multimodal cues (nature of visual or acoustic signals of social media posts) to disambiguate the intent, or to use a few short learning methods when handling infrequent sarcasm types. In the end, the making of frameworks, which is the work of a party to pull differently practiced author-specific sarcastic styles, may have an impact more on cross-domain generalization being facilitated, and this will give the language a more human-like figurative understanding. Although the progress was made, several limitations arose. It turned out that systems were challenged by sarcasm built upon lighter humor or positive sentiment markers (e.g., 'haha' or emoticons), indicating a deficit for detecting 'witty' or implied sarcasm. Furthermore, hashtags in the self annotated sample included potential noise in the form of new labels, notably in tweets from Twitter data, as sarcastic tweets could have used sentiment hashtags.

Ghosh et al. [14] research explores the use of sarcasm and its effects on argumentative relations in online discussions, specifically, agreeing, disagreeing and none.

The posted research uses a large annotated dataset for argumentative discussions on social and political topics, the Internet Argument Corpus (IAC). In each of these examples, sarcasm is being used to attack an opponent or register disagreement, in argumentation that is often more sophisticated than that which could be achieved through straightforward linguistic signalling. Integrating sarcasm detection leads probing the understanding of what kind of scans is responsible for shaping the disagreement space the interactive domain encompassing the emergence of differing opinions and resolution of disputes. The paper uses various computational frameworks such as logistic regression with discrete sarcasm-relevant features, dual LSTM architectures with hierarchical attention and pretrained transformer-based models like BERT with multitask learning (MTL). Our results show a significant performance boost (2-3%) received from multi-task learning, where sarcasm detection aids the argumentative relation classifier across all models. Meanwhile, the best performance which achieved the score of 65.3% F1-micro was the BERT-MTL model with the dynamic loss weights, which outperformed the simple setups. The qualitative analysis of the study shows significant sarcasm cues (like incongruity, sentiment mismatch and hyperbole) as the main factors that allow the models to disambiguate sarcastic vs argument turn.For example, incongruity of the previous turn with the current one was consistently identified as a main signal of sarcasm and is frequently embodied by oppositional sentiment or extreme compliments (INCONGRUENCE). Hyperbole was also used when something out of the ordinary is done to demonstrate difference, albeit in an indirect manner. Beyond just the quantitative results, the article goes on to offer extensive qualitative analysis, looking into cases of instances where the detection of sarcasm has had an impact on the decision of argumentative categorization. By investigating attention weights and feature contributions, authors show that sarcasm creates and deepens disagreement and provides a silent form of underpinning argumentative actions in general, but through most of the time implicitly. They find that sarcasm plays a powerful role in disagreements happening online, one that affects not only the tone, but also the structure of arguments. This paper paves the way to a more interesting research for more knowledge of the relationship among sarcasm, persuasion, and argumentation fallacies to more accurately model the online discourse.

2.3 Summary of Key Findings

From the research timeline of 2020-2021 each of the papers are related to sarcasm detection and every paper focuses on different aspects. So, they used various dataset and each model performs differently. We found BERT based models improved by 2-3% with an F-1 score of 65.3%. On another paper we found three-way classification performance dropped to 57% where two-way classification was 68.33%. Across all the papers we found out some common patterns such as context is crucial and BERT model consistently outperformed machine learning models like SVMs. By analysing these papers we can talk about some significant implications like social media monitoring and in virtual assistance such as chatbots etc. The studies from 2015-2020 showed a gradual and significant improvement in sarcasm detection using word embedding (e.g., Word2Vec, WTMF, GloVe) along with trending deep learning and machine learning models (e.g., BERT, SVM). On top of that, pre-processing techniques like concatenating contextual incongruity and in some other cases prepending

or appending utterances with targets boosted the models performance. Contextual awareness varied, as the dataset from Twitter binded properly and performed well but didn't perform quite well with Reddit. This revealed dataset-dependent effects hinting the need for refining annotation, better contextual understandings and improving sarcasm modeling beyond word opposites. Proposed future work included incorporating inter-sential features and contextual enhancements to improve sarcasm detection across varied platforms. The papers investigate the use of context, linguistic markers, and basic knowledge about the world in detection and generation of sarcasm and irony in social media. The paper shows that transformer-based models such as BERT and RoBERTa are better at sarcasm detection than traditional techniques once context from multi-turn Twitter or Reddit dialogues is used. Also it explores irony markers and shows that while typological cues (emojis, emoticons), are good discriminative on Twitter, morpho syntactic ones (tag questions, exclamations) prevail on Reddit as SVM classifiers are good at capturing these patterns. Again, it brings an unsupervised sarcasm generation framework, R3, that leverages the two control signals (valence reversal and semantic incongruity) with commonsense knowledge of ConceptNet, and achieves human-like performance in creativity and humor. Together, these studies indicate the necessity of such platform specific features, as well as the contextual understanding and theoretical grounding when trying to further NLP tasks such as figurative language, and suggest such points of view to enable more advanced conversational AI and sentiment analysis systems. The papers from 2016 to 2020 included some traditional approaches besides modern ones in detecting sarcasm and irony by identifying the conversational context that triggers irony, semantic incongruity between literal evaluation and context and recognizing pragmatic features. The data were pooled from social medias like Twitter and Discussion Form and preprocessed as necessary. For conversation context detection, LSTM works better than SVM. Among LSTM networks, Conditional LSTM achieved a 73.32% F1 score for sarcastic (S) class and a 7.56% F1 score for non-sarcastic (NS) class in Discussion Forums. For Twitter, Conditional LSTM improved F1 scores by 11% for S class and 4-5% for NS class. Using different linguistic strategies to find semantic incongruity, lexical antonym pairs achieved better results with a 92.21% F1 on the dev dataset and SVM works better among all the machine learning algorithms with a 58.6% F1 score. Turkers uses lexical antonyms to detect sarcasm and the interpretation of contextual irony is correlated with the sarcasm being implicit or explicit. In the pragmatic feature identification process, Human judges worked better when it is annotated by emoticons. Machine learning models achieved better results in binary classification with SVM achieving best results (71.67% accuracy) identifying sarcastic and non-sarcastic data.

Chapter 3

Requirements, Impacts and Constraints

3.1 Final Specifications and Requirements

In order to create an empirical ground to understand how to recognise the presence of sarcasm in a text on an open-source Large Language Model (LLM), a few technical and methodological requirements were formulated. Specifically, the benchmark requires having a publicly available and adequate dataset of Reddit posts that contains a balanced number of sarcastic and non-sarcastic comments. All the models studied in this research, implementing LLaMA-3.1-7b and Mistral-7b were open-source and available in the HuggingFace Transformer libraries. Considering the computational requirements of these large models, their analysis was performed using Google Colab Pro platforms and personal systems capable of running GPU. The entire programming was done in Python using libraries including Scikit-learn, Pandas and Matplotlib. In order to guarantee the uniformity of model results, the prompts enforced by JSON were used during the research. The metrics computed throughout the whole dataset were evaluation based on accuracy, precision, recall, and F1-score.

3.2 Societal Impact

The issue of sarcasm recognition in the context of Natural Language Processing (NLP) has gained scholastic attention due to the direct implications of such societal concerns in the electronic communication processes and sentiment-analysis platforms. Indeed, when AI systems use a proper sarcasm detector, they would acquire more information about how people express linguistically, and, as such, user experience in chatbots, virtual assistants, and customer-service situations will be better. In addition, such an ability is essential to use in content-moderation policies, since the misunderstanding of sarcasm may lead to totally incorrect censorship of the right of free expression or to the continuance of destructive discussion. In mental-health diagnosis and cyber-safety surveillance even the ability to detect sarcastic tones supports denser interpretations of user intent. Nonetheless, the usage of these technologies brings into the forefront the question of the possible excess of oversight and reduction of privacy, especially in the case of their misuse in surveil-

lance or profiling applications.

3.3 Environmental Impact

Even though the scope of the current project does not presuppose any direct environmental changes, one cannot but mention the ecological print of large-scale LLMs. These models require monstrous amounts of computational capabilities and energy consumption follows as a result of this aspect. To keep such an impact to a minimum, the study did not train a model on the scratch but used pre-trained models, which significantly saved resources. Furthermore, to allow controlled execution, minimized unnecessary computation was achieved by reducing computational burden by using platforms with computational overtime constraints like Google Colab and local GPUs. The cost of the environment in this small scenario is relatively low but is still an important factor toward scaling up the project to the real world.

3.4 Ethical Issues

The design and implementation of the present study considered ethics when designing and conducting it. The utilized data set was composed of Reddit posts that were publicly available, which confirmed that no confidential and personal information of users was engaged in the use and, thus, the privacy and anonymity of the users were upheld. One of the most important ethical issues in sarcasm detection involves the cultural and linguistic bias because sarcasm has been found to be situation-specific and it also has differences between demographics and language types. To counter this, the team reassured that the dataset was balanced as well as diverse in its scope. Moreover, the created models were not implemented into the real-time applications with the consequent minimization of the risks connected with misunderstanding or excessive trust to the automated forecasts. Interpretability of model outputs was preserved by structured prompting, and there was no chance of misusing outputs in the form of a black-box decision framework.

3.5 Standards

Several best practices and standards were followed in this study to ensure consistency and reliability. JSON was chosen for flattening and unflattening model outputs due to both readability and structural consistency. This study adhered to the FAIR data principles that used data should be Findable, Accessible, Interoperable and Reusable. Further, the work followed a set of ethical guidelines for developing artificial intelligence as recommended by the IEEE Global Initiative on Ethics of Autonomous and Intelligent Systems, for responsible and transparent conduct of artificial intelligence tools.

3.6 Project Management Plan

The project was controlled by a formal plan, including scheduling, budgeting, and resourcing. Throughout a three-month pre-thesis 2 period, the team researched and experimented with a variety of open-source LLMs to examinetheir ability to detect sarcasm. This stage concerned optimising training, benchmarking and code organisation that suitedthe project goals. Budget: Team spent $10 by subscribing Google Colab Proto facilitate training and testing the models. Other hardware needs weresupplied by BRAC University, such as a RTX 3070 Ti GPU with 8GB of V-RAM, that allowed for effective benchmarking of the chosen models. Resource Management: The work was divided among five team members to attract evenly load and specialization. Two of the members were in charge of programming andimplementation; one member was responsible with the testing and the evaluation and the other two members were responsible for the documentation, report and poster. This joint work was a help from the technicalpoint of view, as well as the public defence presentation.

3.7 Risk Management

Few potential risks were anticipated in the design of this study. One of the biggest issues was the varying LLMoutputs, this was solved using structured prompt formats and output parsing with JSON. Another high-risk here was dataset imbalance or having culturally biased examplesof sarcasm, for which we tackled them by handcrafting and validating a balanced set of training and testing data. There are also concerns about computation overhead (specifically, model size), however, this wasaddressed with optimizations and cloud-based GPU systems. Lastly, we considered that the prompts designed may not work universally across all the models or for different type oftexts, and that is why we experimented with various different types of prompts through out the evaluation.

3.8 Economic Analysis

The cost of theproject is low, considering the economic burden, given that most of the tools used are open-source and the data is publicly available. The principal costwas incurred by cloud computing. Subscriptions to Google Colab Pro and their corresponding GPU compute hourswere needed to execute and measure large language models. The total cost was estimated to be in the range of USD $30–50 by theteam over the life of the project. This price was including some premiumaccess and occasional consumption of local machines' electricity only. In summary, the project was cost-effective to the end, and managed to produce academically useful results that may be extendedfurther in subsequent commercial or research projects.

Chapter 4

Proposed Methodology

4.1 Methodology Overview

This work leverages a prompt-based generative inference with Large Language Models (LLMs) forthe detection of sarcastic comments in Reddit. The approach relies on generative language models, which are a type of transformer-based model that istrained to predict the next word or token in a sequence, given some previously observed context. These models (e.g., LLaMA-3.1, Mistral) leverage self-attention and billions of parameters to unpackthe deep semantics, tone, context, and intent of natural language. The generative method is also compared to conventional machine learning classifiers which are based on feature processing and labeled data. Instead of learning rules or patterns explicitly from training examples, generative models condition on natural language prompts and extrapolate information from a large and general purpose pretraining. This enables flexible adaptive deployment across a diversity of tasks as well as sarcasm detection without task-specific fine-tuning. One of the best things about this method is that it supports zero-shot or few-shot learning environments where a model can be tested on new tasks with very few or no training data. The generative modelsin this study are not retrained or fine-tuned. Butrather than being shown a Reddit post, they're given a sentence prompt that asks if a particular Reddit post is sarcastic and are asked to label it as such, along with explaining their decision. The interaction is demonstrative of howa human would reason: the model reads the post, infers the sentiment, and votes along with a reason. This approach leverages the internal language modeling and context reasoning capability of the generative model, whichis suitable for fine-grained classification problems such as sarcasm detection.

Model Loading: For establishing our workflow we locally installed Ollama and then pulled the LLM models using that. As we also used Colab; so to simulate that workflow; we began by installing Ollama directly into Colab (using !curl -fsSL https://ollama.com/install.sh — sh) at the beginning of all cells. Then Ollama was run in the background so that the following cell to this cell had the local model inferences.

Environment Setup for Ollama Server

The Ollama server was initialized in the background using a simple Python script to allow concurrent execution of evaluation tasks. The following configuration ensures

that the model can be accessed via the local host on port 11434. The code used for this setup is shown in Listing 4.1.

```
import os
import subprocess
import threading
import time

def run_ollama_serve():
    os.environ['OLLAMA_HOST'] = '0.0.0.0:11434'
    os.environ['OLLAMA_ORIGINS'] = '*'
    subprocess.run(["ollama", "serve"])

ollama_thread = threading.Thread(target=run_ollama_serve)
ollama_thread.daemon = True
ollama_thread.start()

print("Ollama server started in background...")
time.sleep(5)
```

Listing 4.1: Python script used to initialize the Ollama server in the background

After this cell block each model was installed using """!ollama pull mistral:7b""".
Initializing the prompt: One of the most crucial steps of our workflow was generating the right prompt. The following code snippet shows the "system_message_prompt" and "user_prompt_template" that was initialized here, which will include data from the contextual columns (i.e., subreddit, parent_comment and comment). This code also includes initializing the model name (i.e., to be invoked locally)

```
system_message_prompt = """
You are a highly accurate sarcasm detection AI.
Your task is to analyze a comment using its full context: the subreddit
    it was posted in, the parent comment, and the comment itself. The
    subreddit provides crucial cultural context.

Your response MUST be a valid JSON object containing a single key:
    "sarcasm_classification".
The value for this key must be an integer: 1 if the comment is sarcastic,
    0 if it is not.
Do not output any text before or after the JSON object.

Example of a valid response:
{"sarcasm_classification": 1}
"""

user_prompt_template = """
Analyze the following content:
- Subreddit: r/{subreddit}
- Parent Comment: "{parent_comment}"
- Comment: "{comment}"
"""
```

Listing 4.2: Baseline system message and user prompt used for sarcasm classification

Iterate, Predict and Store: Once the right prompt is engineered, the dataset was iterated using the following code. In each iteration, data from these contextual columns (subreddit, parent_comment and comment) were extracted and replaced with the placeholders initialized in the "user_prompt_template". The model response was then parsed to convert the responses into integer type and then to be appended (as 'predicted labels') to a newly created "model response dataset" (as the response of the model was instructed to be in JSON format). A safety measure was ensured to make the predicted responses be '-1' should the model fail to respond with the desired response.

Algorithm 1 Sarcasm Detection Comment Analysis Procedure

Require: *df* ▷ DataFrame containing subreddit, comment, parent_comment, and label

model_name ▷ Name of the evaluated model
system_message_prompt ▷ System instruction for the model
user_prompt_template ▷ Template for formatting user prompts
Initialize *results* ← empty list
for each *row* in *df* **do**
    Extract *subreddit_text*, *comment_text*, *parent_comment_text*, and *original_label* from *row*
    user_prompt ← Format(*user_prompt_template*, subreddit=*subreddit_text*, parent_comment=*parent_comment_text*, comment=*comment_text*)
    response ← ollama.chat(model=*model_name*, messages=[{role: "system", content: *system_message_prompt*}, {role: "user", content: *user_prompt*}])
    model_answer_text ← Clean(response.message.content)
    parsed_json ← $JSON\_PARSE(model\_answer\_text)$
    predicted_label ← Integer($parsed\_json$["$sarcasm\_classification$"])
    explanation ← $parsed\_json$["$explanation$"]
    if *predicted_label* $\notin [0, 1]$ **then**
        predicted_label ← -1
    end if
    Append result to *results* with all extracted and predicted values
     Exception *e*
    Display "Error at index", $row.Index$
    Append error entry to *results* with *predicted_label* $= -1$
end for
results_df ← Convert_to_DataFrame(*results*)
Save *results_df* as CSV($output\_filename$)

4.2 Preliminary Design or Design (Model) Specification

A systematic approach used in the study is divided into four major steps: dataset selection and preparation, the engineering of the prompts, model inference, and evaluation. The aim of each phase was to create a thorough and impartial comparison of

two open-source LLMs LLaMA-3.1 and Mistral-7b in an experimental environment.

Dataset Selection and Preprocessing:
This paper uses the Self-Annotated Reddit Corpus (SARC), a large-scale benchmark proposed by Khodak et al. [15] and made publicly accessible on Kaggle (train-balanced-sarcasm.csv). The corpus is made up of millions of Reddit comments that are annotated as either sarcastic or non-sarcastic by community participants and their feedback. The data source used is Reddit since the platform is informal and discussion oriented that naturally incorporates sarcasm, humour, and irony, and it is therefore a realistic setting in which to test sarcasm-detection models. Every line in the dataset has contextual and textual aspects of the subreddit, parent_comment (context), comment (target text), and label (ground truth). These characteristics enable models to comprehend sarcasm in conversational context, as opposed to outside conversational situations. The dataset was balanced, and emojis were already converted to text, thus minimal pre-processing was required at our end.

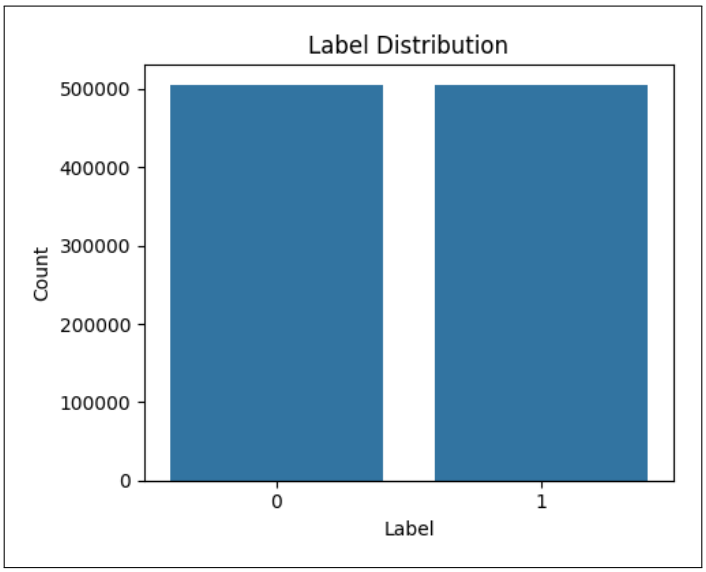


Figure 4.1: Balanced Labels

Prompt Engineering and Query Format:
The current exploration is focusing on early engineering, namely early construction of input prompts to guide a model toward performing a preselected task, in the given case, sarcasm detection. A well-organized statement prompt was developed in the following way: system_message_prompt = """ You are a highly accurate sarcasm detection AI. Your task is to analyze a comment using its full context: the subreddit it was posted in, the parent comment, and the comment itself. The subreddit provides crucial cultural context. Your response MUST be a valid JSON object containing a single key: "sarcasm_classification". The value for this key must be an integer: 1 if the comment is sarcastic, 0 if it is not. Do not output any text before or after the JSON object. Example of a valid response: "sarcasm_classification": 1 """ user_prompt_template = """ Analyze the following content:Subreddit: r/subreddit,Parent Comment: "parent_comment",Comment: "comment","""
The selection of this template was also done to force one to have a formatted output schema, which affords an efficient mechanical parsing and extraction of the results.

The deposition of the phrase brief explanation encourages the model to produce a succinct rationale on its forecast which then it is simpler to apply qualitative analysis. More importantly, even the prompt itself does not carry any exemplar data, and therefore the experimental scenario may be deemed as representing a zero-shot learning setting, where the model must generalize based solely on the model parameters it is pre-trained on.
In the case of every Reddit post to the collection being examined, the plaintext was coded in a consistent template and forwarded to two current open-source LLMs, whereupon it was run through some inference. Same format was adhered to in each model so that the subsequent performance comparison would be based on the same input conditions.

Model Selection and Inference Procedure:
Three LLMs were chosen to benchmark: LLaMA-3.1 by Meta that gained popularity due to its high performance in reasoning and wide language coverage; Mistral-7b, lighter and efficient type of model and capable of outstanding performance in a low-resource environment. The above-mentioned models were loaded into a Python environment through HuggingFace Transformers library and run on a GPU-powered computing architecture Google Colab Pro instances and local NVIDIA environments. The pre-trained models were used to run in their pre-training forms without fine-tuning them specifically on a given task.
The template was filled in and finally presented to the chosen model/s according to each Reddit post. There was automated data extraction of the label field and reason field contained in each of the responses received to each post. The reason field was set aside to be used in terms of qualitative analysis but excluded in later classification measurements.

4.2.1 Result and Analysis

Our research's systematic evaluation for open-source LLMs in sarcasm detection begins with prompt engineering and culminates in an analysis of model scale. Our experimentation revealed distinct behavioral biases in smaller models (7-8B parameters) and ultimately demonstrated that model scale, and the right prompt is the most critical factor for achieving nuanced, reliable performance on this complex linguistic task.
Comprehensive Performance Comparison: Table 4.1 summarizes the key experiments, tracking the evolution from initial Chain-of-Thought (CoT) prompts to the robust and optimal candidate "JSON-based prompt", and comparing models across different architectures and sizes.

Key Insights

- **Prompt Engineering Exposes Model Biases:** While advanced prompting (CoT, JSON formatting) improves the models instruction-following, it also reveals the inherent biases of smaller models. Llama 3.1 8B consistently defaulted to an aggressive "always sarcastic" strategy (i.e. High Recall, Low Precision), while Mistral 7B initially showed an opposite "overlay cautious" bias (High Precision, Low Recall).

Table 4.1: Model Performance Comparison on Sarcasm Detection

Model	Prompt Strategy	Accuracy	Precision	Recall	F1-Score
LLaMA 3.1 8B	CoT	51.0%	50.5%	100.0%	67.1%
Mistral 7B	CoT	58.3%	**72.7%**	22.9%	34.8%
Mistral 7B	JSON + Context	59.6%	55.3%	95.9%	70.2%
LLaMA 3.1 8B	JSON + Context	52.1%	51.1%	99.0%	67.3%
LLaMA 3.1 70B	CoT	**64.0%**	58.3%	98.0%	**73.1%**

- **Smaller Models "Reasoning Ceiling":** The 7-8B models, regardless of prompt sophistication, ultimately failed to achieve a balanced performance. Adding more context (subreddit) caused Mistral to abandon its "cautiousness" and adopt the same aggressive bias as Llama. This indicated that these models over-rely on simple heuristics rather than performing complex reasoning.
- **Model Scale:** The most significant finding is that the Llama 3.1 70B model, even with an earlier prompt version, delivered the best performance. Its ability to significantly improve Precision while keeping Recall high demonstrates a superior capacity for linguistic nuance. It was successful in using complex prompt instruction to correctly identify more non-sarcastic comments, which the smaller models consistently failed.

4.2.2 Research Gap

While proprietary models like GPT, Grok excel, their closed nature limits research transparency. A significant gap exists in the systematic benchmarking of open-source LLMs zero-shot classification performance on dynamic, context-dependent tasks like sarcasm detection. Much of the current focus remains on generative capabilities, leaving room for improvements in prompting strategies for classification un-explored. This study addresses this gap by:

1. Providing a direct comparative analysis of major open-source models (Llama vs Mistral).
2. Quantifying how model scale (8B vs 70B parameters) directly impacts reasoning, bias, and adherence to complex instructions.
3. Evaluating the practical ceiling of advanced prompt engineering techniques like CoT and especially JSON enforcer for eliciting reliable classification without task-specific training.

4.2.3 Limitations

This study's scope lays the foundation for a clear path for future research. Firstly, the primary limitation was our reliance on prompt engineering due to computational constraints preventing fine-tuning. Consequently, we could not fully correct the models inherent biases, and future work should compare these results against fine-tuned counterparts to quantify the performance differences.
Secondly, our analysis was limited to the provided text (comment, parent comment,

subreddit) and did not programmatically incorporate the full conversation history from external links. In future, studies could leverage this deeper contextual awareness to test model performance in a more information-rich, real-world context and particularly for resolving highly ambiguous cases of sarcasm.

Chapter 5

Result Analysis

5.1 Performance Evaluation

5.1.1 Llama 3.1-8B

Overview

In this report, the results of two balanced evaluation data sets (Dataset 28 and Dataset 29, which have a total of 50 samples each) are combined to provide a detailed architectural overview for the Llama 3.1-8B open-source Large Language Model (LLM).The main aspects of analysis are structural bias in the model, stability of its performance and the chains of thought (CoT) reasoning limits for the classification of sarcasm devices

Model Overview

Llama 3.1-8B [16] is a decoder-only transformer that serves as an incremental update to the Llama 3 base architecture.
Key technical characteristics include:

- Parameters: 8 B
- Layers: 32
- Hidden size: 4096
- Attention heads: 32
- Context length: 128k tokens
- Core Components: Rotary Position Embedding (RoPE), SwiGLU activation, and RMS Normalization.
- Origin: Developed by Meta as an improved and more capable version of the Llama 3 model, optimised for a balance of performance and efficiency.

Overall Performance and Architectural Bias (Combined)

Metric	Score	Implementation
Accuracy	0.6907	Correctly classified 69.07% of all comments.
Precision (sarcastic = 1)	0.6406	Prediction was correct 64.06% of the time.
Recall (sarcastic = 1)	0.8542	Identified 85.42% of all genuinely sarcastic comments.
F1-Score (sarcastic = 1)	0.7321	Overall balanced perfomance.

Table 5.1: Performance of combined samples

The data in the table is a summary of the combined classification metrics of Llama 3.1-8B (N = 100).The difference between the high Recall (0.8542) and moderate Precision (0.6406) shows a characteristic architectural bias: Llama 3.1-8B prefers to over-detect sarcasm rather than miss it. Its detection mechanism prioritises coverage and contextual sensitivity, thus resulting in a large number of True Positives but also a fair number of False Positives.This suggests that the model's architecture is a high-sensitivity incongruity detector, as opposed to a precision-oriented classifier.

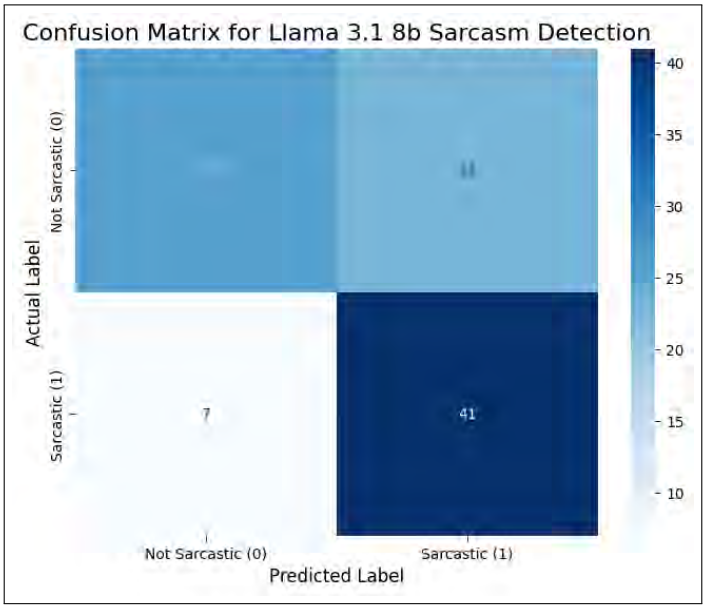


Figure 5.1: Confusion Matrix (28 & 29 Combined)

Per-Device Performance

Device	Precision	Recall	F1-score	Support
Hyperbole	0.17	0.30	0.21	10
Contradiction	0.30	0.19	0.23	16
Contextual Incongruity	0.43	0.63	0.51	18
Feigned Ignorance	0.00	0.00	0.00	6
Non-sarcastic	0.82	0.54	0.65	50
Noise	0.00	0.00	0.00	0

Table 5.2: Per-device classification report

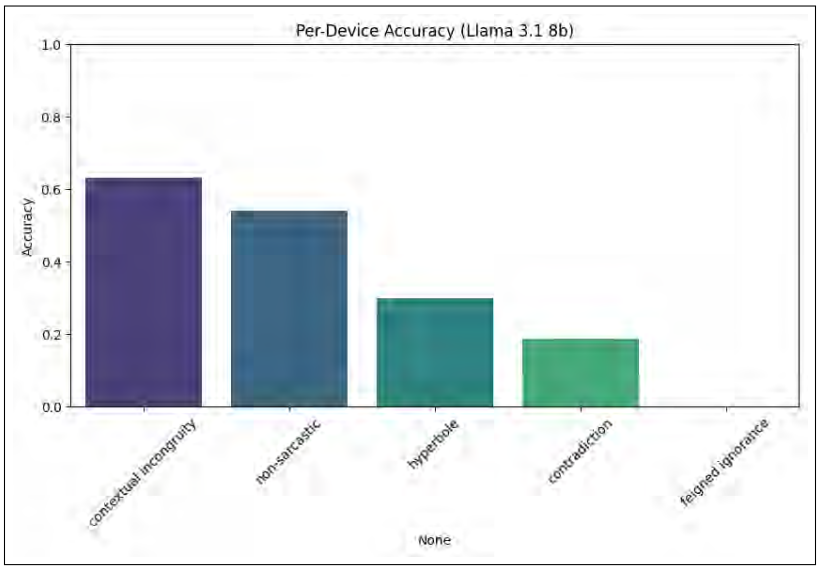
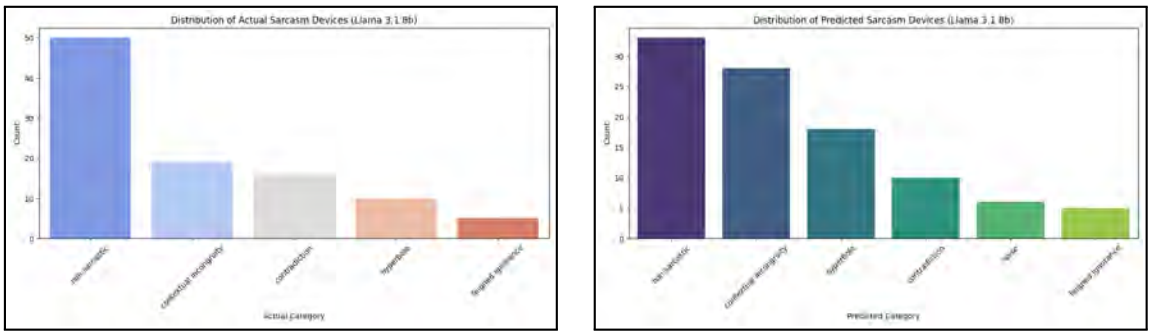


Figure 5.2: Per-Device Accuracy

Sarcasm Device	Accuracy	Architectural Influence
Contextual Incongruity	0.6315	Relies on **RoPE** and **multi-head attention** to process long-range context in comment threads.
Non-sarcastic	0.540	High sensitivity to **sentiment polarity** and **token embeddings** creates a bias for sarcasm detection.
Hyperbole	0.30	The **SwiGLU network** helps detect obvious exaggeration by using intense lexical cues.
Contradiction	0.1875	Uses a **"Contradiction Heuristic"**, simplifying complex irony into basic logical checks.
Feigned Ignorance	0.00	The architecture **lacks a mechanism for intent modeling**, making it unable to understand cognitive irony.

Table 5.3: Architectural Influence on Accuracy

Mean sarcasm-device accuracy (excluding "Other"): 0.3318.
This quantifies the gap between recognising sarcasm's presence (overall 0.6907 accuracy) and identifying its type (0.3318 accuracy).



(a) Distribution of Actual Sarcasm Devices **(b)** Distribution of Predicted Sarcasm Devices

Figure 5.3: Comparison between Distribution of Actual & Predicted Sarcasm Devices

5.1.2 Mistral 7B

Overview

The performance of the Mistral 7B model in sarcasm detection (results of two separate evaluation datasets Prediction-31 and Prediction-32, each containing 50), assessed in terms of its classification measures, and Chain-of-Thought (CoT) predictive accuracy, points to a very effective operational profile as characterized by an efficient balance between expediency/efficiency and predictive accuracy. The specified behavior is closely associated with its novel architectural elements, Sliding Window Attention (SWA) and Grouped-Query Attention (GQA).

Model Overview

Mistral 7B [17] is a 7.3-billion-parameter decoder-only transformer language model designed for efficiency and high performance across a wide range of tasks.
Key technical characteristics include:

- Parameters: 7.3 B
- Layers: 32
- Hidden size: 4096
- Key-Value heads: 8
- Sliding window size in local attention: 4096 tokens
- Core Components: Grouped-Query Attention (GQA), Sliding Window Attention (SWA), SiLU activation, and RMS Normalization
- Attention mechanisms: Grouped-Query Attention (GQA) for faster inference and reduced memory during decoding; Sliding Window Attention (SWA) for effective handling of longer sequences with lower inference cost

Overall Performance and Architectural Bias (Combined)

Metric	Score	Implementation
Accuracy	0.5521	Correctly classified 55.21% of all comments.
Precision (sarcastic = 1)	0.5376	53.76% of predicted sarcastic comments were actually sarcastic
Recall (sarcastic = 1)	1.0000	High detection sensitivity
F1-Score (sarcastic = 1)	0.6999	Overall performance

Table 5.4: Performance of combined samples

The performance of the model on the combined data (N=100) reveals a strategic architectural bias. It is optimized for high sensitivity (Recall), successfully identifying all true sarcastic instances, but this comes at the cost of lower prediction certainty (Precision). This is evident from the model's tendency to generate a high number of false positives.

The architecture of Mistral 7B highlights a significant gap between the high Recall (1.0000) and lower Precision (0.5376). Which indicates that the semantic detection unit is sound but at the expense of lowering precision. As it captures a high volume of sarcastic instances (true positives) it also generates a higher number of false positives, thus trading off precision.

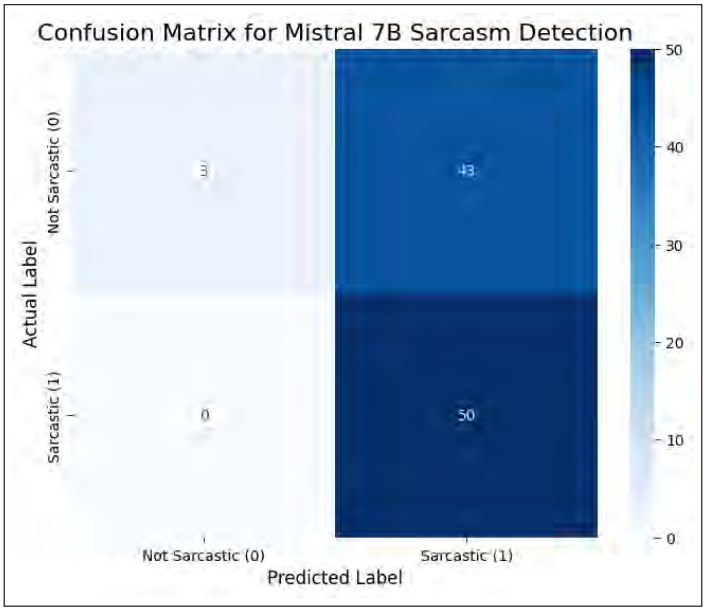


Figure 5.4: Confusion Matrix (31 & 32 Combined)

Per-Device Performance

Device	Precision	Recall	F1-score	Support
Hyperbole	0.00	0.00	0.00	10
Contradiction	0.24	0.50	0.32	16
Contextual Incongruity	0.24	0.50	0.32	18
Feigned Ignorance	0.25	0.17	0.20	6
Non-sarcastic	1.00	0.10	0.18	50
Noise	0.00	0.00	0.00	0

Table 5.5: Per-device classification report

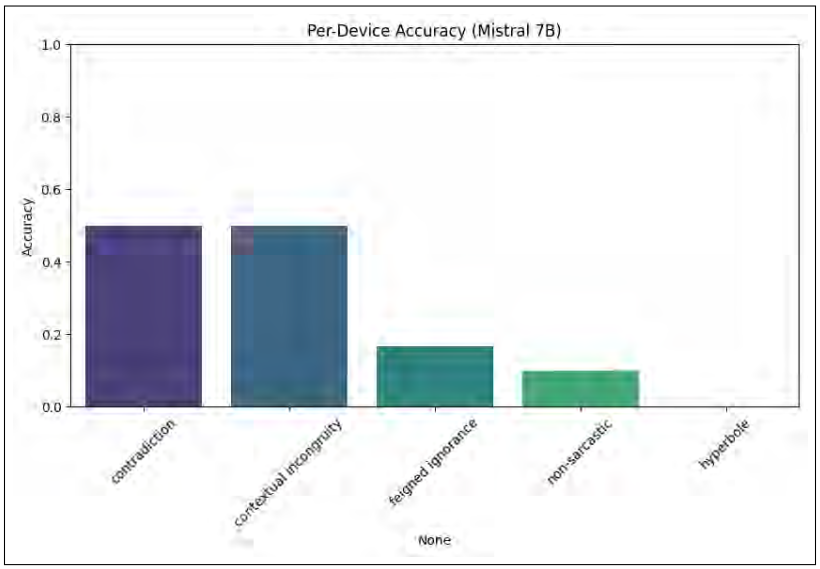
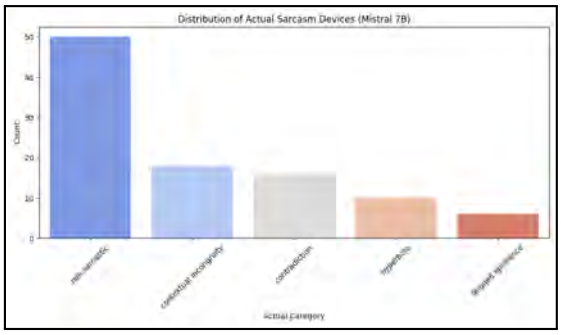


Figure 5.5: Per-Device Accuracy

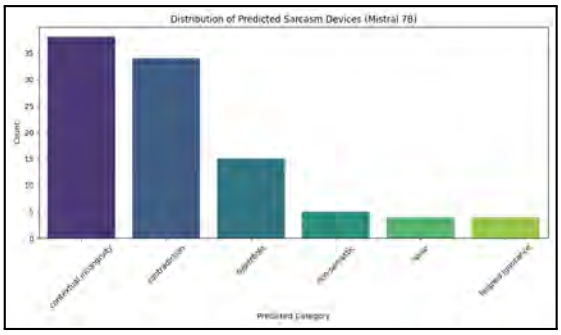
Sarcasm Device	Accuracy	Architectural Influence
Contradiction	0.50	The model's foundational logic, supported by Sliding Window Attention (SWA) for contextual comparison, allows for moderate detection of direct logical conflicts.
Contextual Incongruity	0.50	Performance is directly tied to SWA, which enables the model to process long-range context. Its moderate success suggests a decent but imperfect ability to grasp situational irony.
Feigned Ignorance	0.167	This is identified as a structural "Blind Spot." The architecture fails to process this nuanced social cue, causing it to be misclassified as a more basic logical flaw like contradiction.
Non-sarcastic	0.10	The extremely low accuracy reflects the model's overall architectural bias. It aggressively classifies comments as sarcastic to achieve high recall, leading to a very high number of false positives.
Hyperbole	0.0	While the model fails to recall any true instances, it suffers from an "Over-Projection" bias, frequently mislabeling other inconsistencies as hyperbole. This indicates a failure in classification, not in the detection of incongruity itself.

Table 5.6: Per-Device Architectural Influence on Accuracy

Mean sarcasm-device accuracy (excluding "Other"): 0.2534.



(a) Distribution of Actual Sarcasm Devices



(b) Distribution of Predicted Sarcasm Devices

Figure 5.6: Comparison between Distribution of Actual & Predicted Sarcasm Devices

5.1.3 Gemma 3 12B

Overview

The results from both evaluation datasets Dataset 35 and Dataset 36 (each containing 50 samples) are combined to provide a comprehensive view of the operational characteristics of the Gemma12B model.

Model Overview

Gemma 3 12B [18] is a 12-billion-parameter multimodal decoder-only transformer designed by Google, capable of processing both text and images.
Key technical characteristics include:

- Parameters: 12 B
- Layers: 42
- Hidden size: 3072
- Attention heads: 48
- Context length: 128k tokens
- Attention structure: Grouped-Query Attention (GQA) with a pattern of local sliding window self-attention layers interleaved with global self-attention layers.
- Core Components: Rotary Position Embedding (RoPE) with increased base frequency for extended context, RMS Normalization, and a tailored SigLIP vision encoder for images.

Overall Performance and Architectural Bias (Combined)

This analysis reveals that the model demonstrates a notable architectural bias towards **maximizing Recall**, prioritizing the detection of sarcasm even at the expense of **Precision**.

Metric	Score	Implementation
Accuracy	0.5758	Correctly classified 57.58% of all comments.
Precision (sarcastic = 1)	0.5435	Prediction was correct 54.35% of the time.
Recall (sarcastic = 1)	1.000	High detection sensitivity (100%).
F1-Score (sarcastic = 1)	0.7042	Overall performance.

Table 5.7: Performance of combined samples

The data from both datasets highlights a significant gap between the **high Recall** (1.0000) and lower **Precision** (0.5435). This discrepancy indicates that **Gemma3 12B** is optimized to cast a broad net for sarcasm detection, capturing a high volume of sarcastic instances but generating a higher number of **false positives**, thus affecting **Precision**. This operational focus on **Recall** enables the model to be more sensitive to detecting sarcasm at the cost of **predictive certainty.**

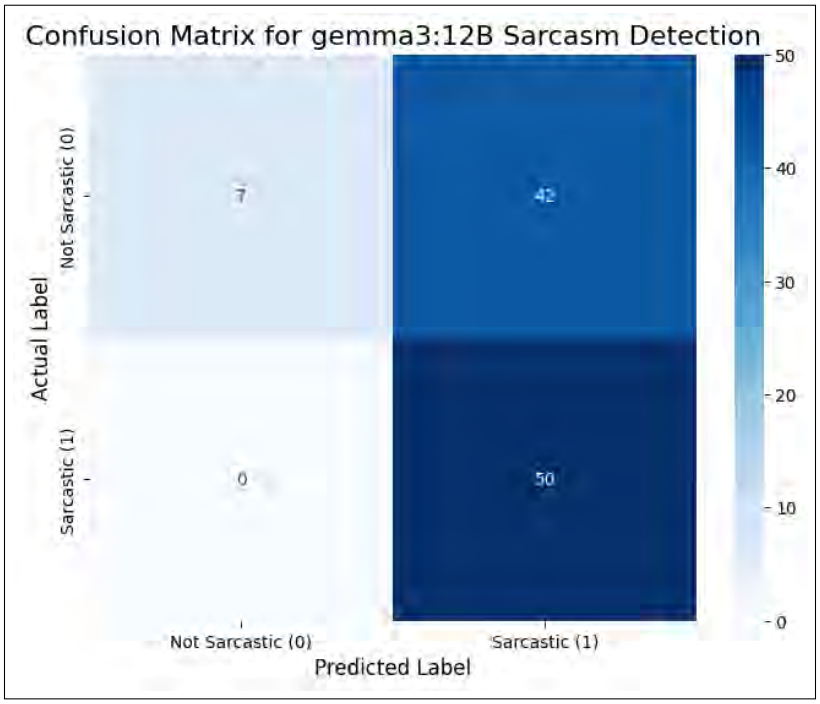


Figure 5.7: Confusion Matrix (35 & 36 Combined)

Per-Device Performance

Device	Precision	Recall	F1-score	Support
Hyperbole	0.27	0.58	0.37	10
Contradiction	0.55	0.80	0.65	16
Contextual Incongruity	0.32	0.56	0.41	18
Feigned Ignorance	0.21	0.60	0.32	6
Non-sarcastic	1.00	0.14	0.25	50
Noise	0.00	0.00	0.00	0

Table 5.8: Per-device classification report

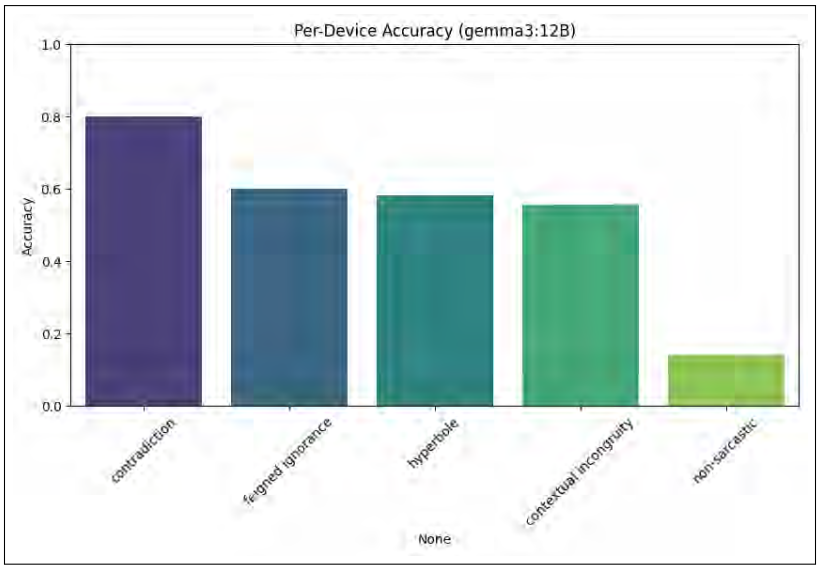


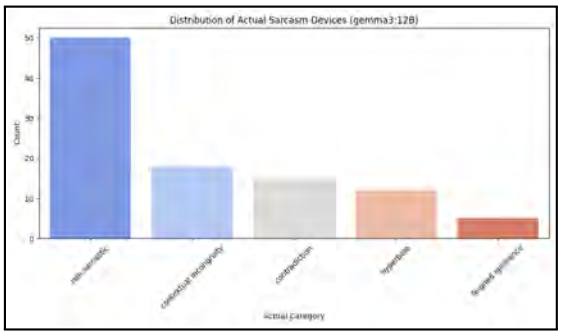
Figure 5.8: Per-Device Accuracy

Sarcasm Device	Accuracy	Architectural Influence
Contradiction	0.800	High recall bias aligns with explicit polarity shifts; architecture favors clear logical contradictions where context and literal meaning diverge sharply.
Feigned Ignorance	0.600	Moderate success due to reliance on linguistic pattern cues ("pretending not to know"); however, lacks deeper pragmatic inference.
Hyperbole	0.583	Chain-of-Thought focuses on exaggeration signals but overweights lexical intensity; lacks calibration for intended absurdity vs. genuine emphasis.
Contextual Incongruity	0.556	Architecture struggles with multi-sentence context integration; transformer attention insufficiently models pragmatic mismatch across comment threads.
Non-sarcastic	0.140	Recall-dominant bias causes overflagging of sarcasm; architecture lacks an inhibitory mechanism for negative sarcasm classification.

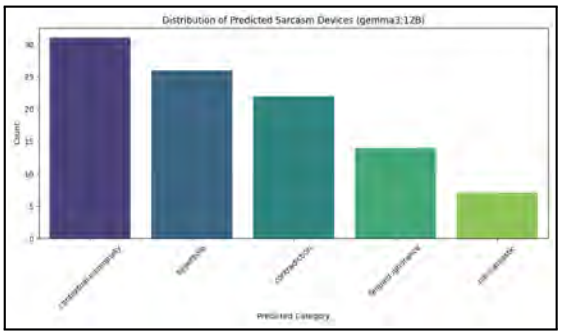
Table 5.9: Per-Device Performance

Mean sarcasm-device accuracy (excluding "Other"): 0.5358.
This quantifies the gap between recognising sarcasm's presence (overall 0.5758 accu-

racy) and identifying its type (0.5358 accuracy). **In other words, Gemma3 12B knows that something sounds ironic and almost why.**



(a) Distribution of Actual Sarcasm Devices



(b) Distribution of Predicted Sarcasm Devices

Figure 5.9: Comparison between Distribution of Actual & Predicted Sarcasm Devices

5.1.4 Qwen3 8B

Overview

In this report, the results of two separate evaluation datasets (Dataset 38 and Dataset 39, with 100 samples each) are combined to give a detailed architectural overview of the Qwen3-8B Large Language Model (LLM). It is analyzed in terms of structural bias in the model, stability of its performance, and the limitations involved in its Chain-of-Thought (CoT) reasoning to classify sarcasm devices.

Model Overview

Qwen3 8B [19] is an 8-billion-parameter decoder-only transformer developed by Alibaba, optimized for multilingual understanding and reasoning across diverse text-based tasks. **Key technical characteristics include:**

- Parameters: 8.2 B
- Layers: 40
- Hidden size: 4096
- Attention heads: 64
- Context length: 40k tokens
- Attention structure: Grouped Query Attention (GQA) for optimization of inference speed and memory usage
- Core Components: Rotary Position Embedding (RoPE). It uses Grouped Query Attention (GQA) which separates the attention heads into query heads and key-value heads, with 32 query heads and 8 key-value heads. The QKV projections keep bias terms for improved model performance.

Overall Performance and Architectural Bias (Combined)

The combination of performance indicators with Qwen3-8B model sets the clear operation profile which was characterized by a significant architectural bias toward maximizing Recall.

Metric	Score	Implementation
Accuracy	0.6100	Correctly classified 61% of all comments.
Precision (sarcastic = 1)	0.5775	57.75% of predicted sarcastic comments were actually sarcastic.
Recall (sarcastic = 1)	0.8200	Detected 82% of all genuinely sarcastic comments.
F1-Score (sarcastic = 1)	0.6777	Overall performance

Table 5.10: Performance of combined samples

The data in the table includes a combination of Classification Metrics of Qwen3-8B (N=100) . The huge difference between the high Recall (0.8200) and lower Precision (0.5775) is a characteristic feature of the architecture of Qwen3-8B used in this task. This Recall is high, which means that the algorithm by which contextual signals are detected within the model is very sensitive. This architecture focuses more on making a broad net to encompass every possible sarcasm that is possible and it can effectively reduce False Negatives. The consequence of this operational choice, however, is that it generates many False Positives, and this lowers the Precision score. The Qwen3-8B is thus structurally-optimized to be a sarcasm finder that prefers coverage of detection and not predictive certainty.

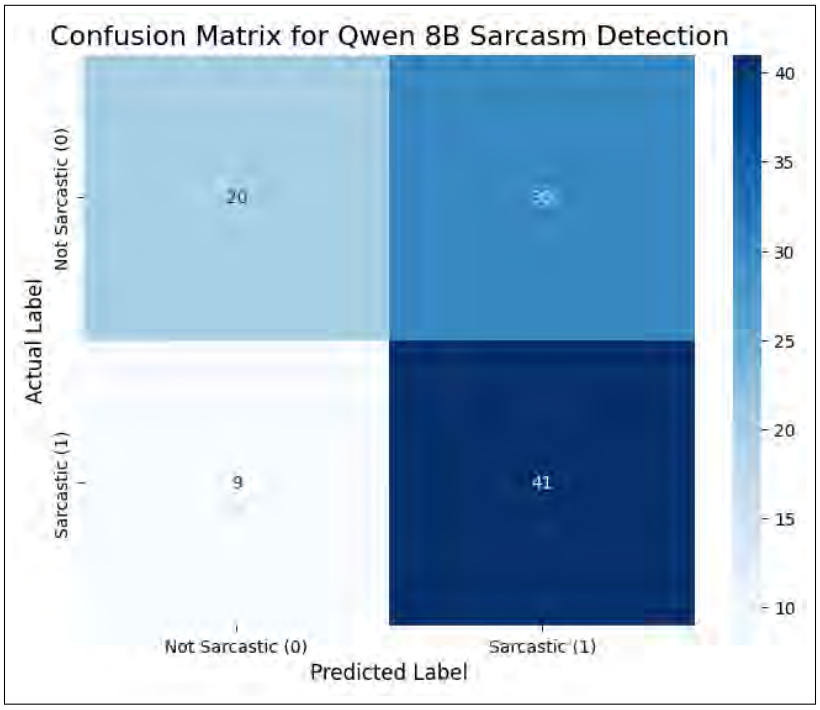


Figure 5.10: Confusion Matrix (38 & 39 Combined)

Device	Precision	Recall	F1-score	Support
Hyperbole	0.00	0.00	0.00	10
Contradiction	0.36	0.50	0.42	16
Contextual Incongruity	0.33	0.39	0.36	18
Feigned Ignorance	0.11	0.17	0.13	6
Non-sarcastic	0.69	0.40	0.51	50
Noise	0.00	0.00	0.00	0

Table 5.11: Per-device classification report

Per-Device Performance

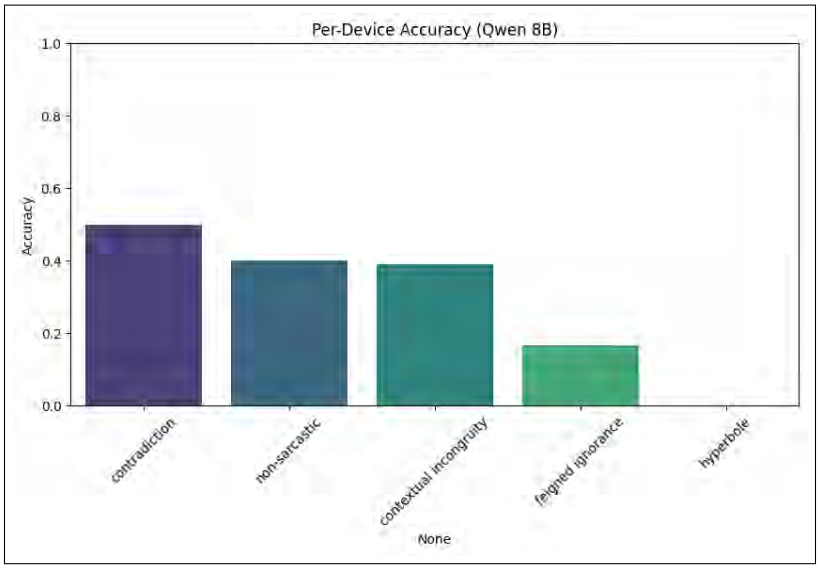
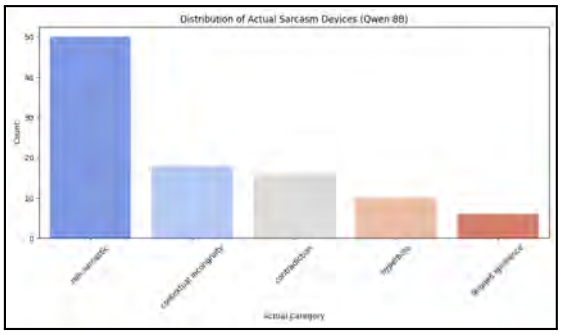


Figure 5.11: Per-Device Accuracy

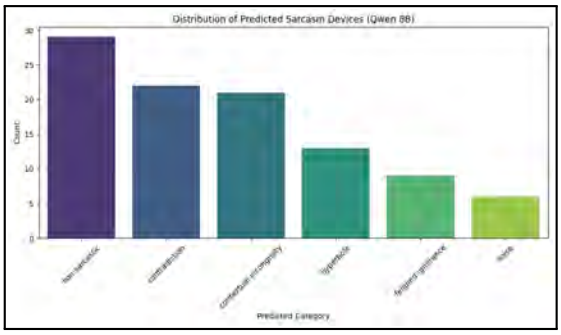
Sarcasm Device	Accuracy	Architectural Influence
Contradiction	0.50	The model is architecturally biased by a "Contradiction Heuristic." Its Transformer design is optimized to process text based on its most immediate and evident logical structure.
Non-sarcastic	0.40	The architecture prioritizes high Recall through a "wide net" design intended to minimize False Negatives. This is combined with instability from the QKV bias, which causes the model to over-extrapolate when faced with novel data.
Contextual Incongruity	0.389	The architecture functions as a "generalized incongruity detector." The use of RMSNorm and SwiGLU activation creates high sensitivity to semantic distance and contextual cues, allowing it to detect that something is "off."
Feigned Ignorance	0.167	The model's reasoning is limited by the "Contradiction Heuristic." Its architecture lacks the semantic detail required to distinguish subtle rhetorical devices like false naivete from a basic logical inconsistency.
Hyperbole	0.00	This device is the most significant victim of the "Contradiction Heuristic" and the model's "semantic constraint." The model is structurally unable to differentiate exaggeration from a direct contradiction.

Table 5.12: Architectural Influence on Accuracy

Mean sarcasm-device accuracy (excluding "Noise"): 0.2912.
This quantifies the gap between recognising sarcasm's presence (overall 0.61 accuracy) and identifying its type (0.2912 accuracy).
In other words, Qwen3 8B knows that something sounds ironic but rarely why.



(a) Distribution of Actual Sarcasm Devices



(b) Distribution of Predicted Sarcasm Devices

Figure 5.12: Comparison between Distribution of Actual & Predicted Sarcasm Devices

5.1.5 DeepSeek-R1-14B

Overview

In this section, the results of two separate evaluation datasets (i.e. coding sheets) Prediction-40 and Prediction-41, each containing 50 samples, are combined to provide a detailed architectural overview of the DeepSeek-R1-14B model.
This analysis is made in terms of model's structural bias, stability of its performance, and the coherence between its internal reasoning (Chain-of-Thought) and true sarcasm devices in relation to its architectural design and distillation origin (Qwen2.5-14B).

Model Overview

DeepSeek-R1-14B [20] is a dense transformer derived from the Qwen 2.5-14B base architecture.
Key technical characteristics include:

- Parameters: 14 B
- Layers: 40
- Hidden size: 5120
- Attention heads: 80
- Core Components: Rotary Position Embedding (RoPE), SwiGLU activation, RMS Normalization, and QKV bias attention
- Origin: Distilled from the large Mixture-of-Experts (MoE) DeepSeek-R1 model into a dense variant optimized for general reasoning.

This architecture aims to preserve reasoning coherence while improving efficiency, but the distillation from an MoE topology also eliminates expert specialization an element crucial to pragmatic reasoning such as sarcasm comprehension.

Overall Performance and Architectural Bias (Combined)

Metric	Score	Implementation
Accuracy	0.6600	Correctly classified 66% of all comments.
Precision (sarcastic = 1)	0.6429	64.29% of predicted sarcastic comments were actually sarcastic.
Recall (sarcastic = 1)	0.7200	Detected 72% of all genuinely sarcastic comments.
F1-Score (sarcastic = 1)	0.6792	Balanced measure of precision and recall.

Table 5.13: Performance of combined samples

These results indicate that DeepSeek-R1-14B maintains a moderate precision-recall trade-off. The high recall reflects a structural bias towards detecting any form of semantic incongruity, while its moderate precision (0.6429) suggests overgeneralization, a byproduct of its RMSNorm + SwiGLU configuration, which enhances sensitivity but reduces selectivity.
Architecturally, the model acts as a high-recall semantic detector rather than a narrowly focused classifier. This behavior is consistent with its dense transformer design (based on Qwen2.5-14B) that favors robust contextual capture through Rotary Position Embeddings (RoPE) and RMS Normalization.

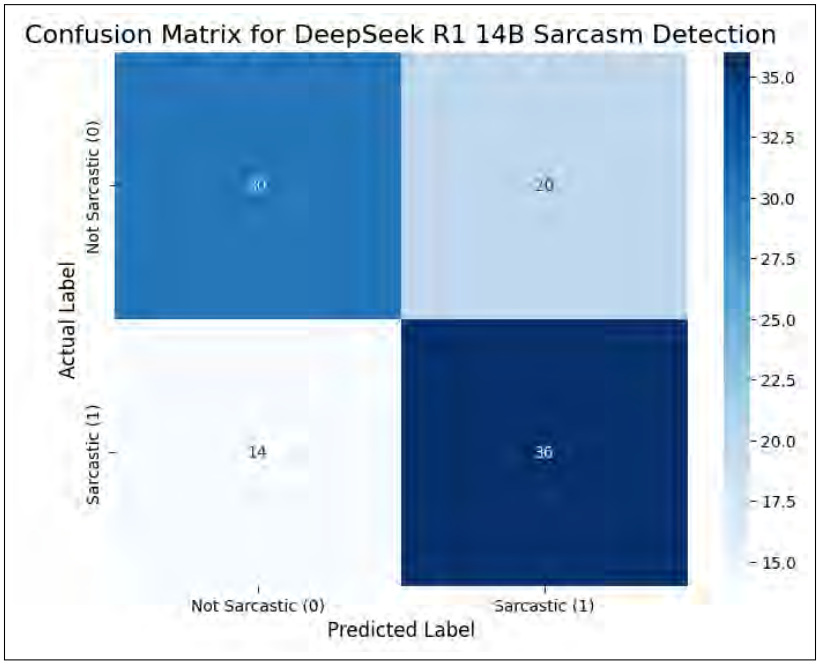


Figure 5.13: Confusion Matrix (40 & 41 Combined)

Device	Precision	Recall	F1-score	Support
Hyperbole	0.25	0.50	0.33	10
Contradiction	0.20	0.12	0.15	16
Contextual Incongruity	0.38	0.17	0.23	18
Feigned Ignorance	0.14	0.33	0.20	6
Non-sarcastic	0.68	0.60	0.64	50
Noise	0.00	0.00	0.00	0

Table 5.14: Per-device classification report

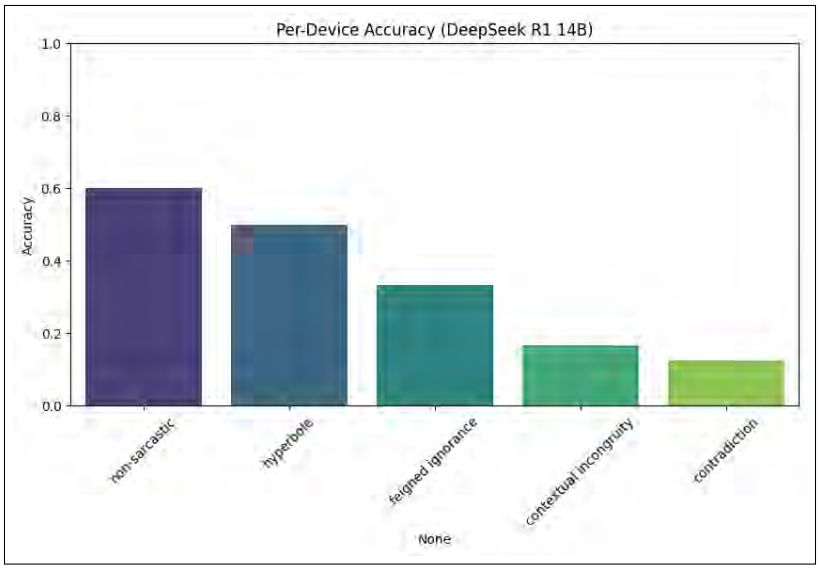


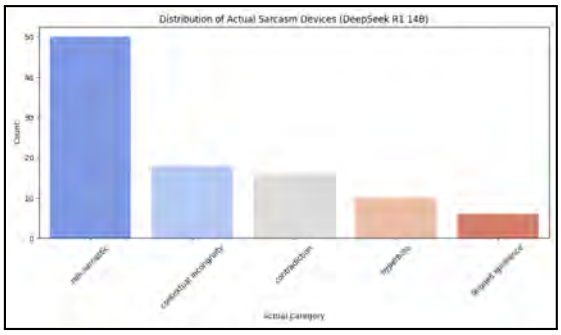
Figure 5.14: Per-Device Accuracy

Per-Device Performance

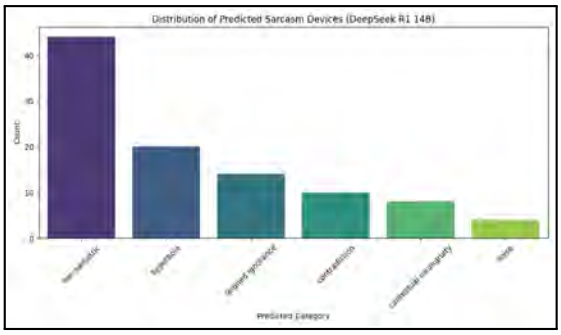
Sarcasm Device	Accuracy	Architectural Influence
Hyperbole	0.50	SwiGLU + RoPE
Contradiction	0.13	RMSNorm + Dense Distillation
Contextual Incongruity	0.17	RoPE + Attention Bias
Feigned Ignorance	0.33	Loss of MoE Specialisation
Non-sarcastic	0.60	RMSNorm + RoPE

Table 5.15: Architectural Influence on Accuracy

Mean sarcasm-device accuracy (excluding "Other"): 0.346.
This quantifies the gap between recognising sarcasm's presence (overall 0.66 accuracy) and identifying its type (0.346 accuracy). **In other words, DeepSeek-R1-14B knows that something sounds ironic but rarely why.**



(a) Distribution of Actual Sarcasm Devices



(b) Distribution of Predicted Sarcasm Devices

Figure 5.15: Comparison between Distribution of Actual & Predicted Sarcasm Devices

5.2 Analysis of Design Solutions

5.2.1 Llama 3.1 8B

Architectural Findings and Impact

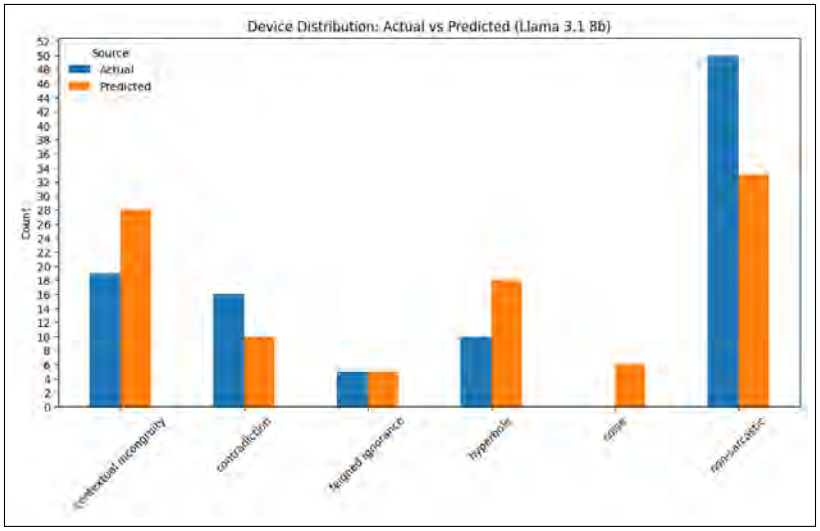


Figure 5.16: Device Distribution: Actual vs Predicted

- **Balanced Detection Architecture**
  Llama 3.1-8B exhibits a balanced relationship between Recall and Precision of 0.8542 and 0.6406 respectively, indicating that the architecture is not only high-Recall or high-Precision oriented. Its decoder-only transformer with multi-head attention and Rotary Positional Embeddings (RoPE) allows reliable comparison of the context of parent and child comments, and can effectively detect incongruity in context.

  Structural Cause (RoPE + SwiGLU): RoPE gives positional consistency across long contexts (very useful for Reddit-style comment threads), while the SwiGLU feed-forward design is useful for modeling high-intensity lexical patterns (another big clue to detecting hyperbole).The result is a well-rounded contextual detector which is very good at detecting exaggeration and contradiction.

Impact: Llama 3.1-8B is an architectural experiment in a context-aware sarcasm recognizer with good generalization and moderate sensitivity to rhetorical tone. Sarcasm is best when it has distinct linguistic indicators.

- **CoT-Driven Pragmatic Reasoning:**
  The CoT structure enhances the interpretability but not necessarily the fidelity. Reasoning was found to be nearly identical to human categorisation in about 41% of cases; in the other cases post-hoc rationalisation was observed. This is a practical division between the detection layers of the model (vector-based decision) and its generator of textual reasoning (language head).

  Structural Cause (Functional Dichotomy): Although the internal attention layer correctly detects semantic incongruity, the text-generation layer often builds its explanation according to common rhetorical terms (e.g., "contradiction", "hyperbole") without establishing a strong connection to the underlying real cause.

  Impact: The CoT justification, while partially authentic - better than Qwen3-8B, for instance - is still not a transparent reflection of internal reasoning.Thus, examining its CoT content reveals patterns of reasoning rather than exact correspondence of cognition.

- **Generalization and Stability:**
  Compared to other models, Llama 3.1-8B is quite stable in terms of performance across datasets (Δ Accuracy=+0.08). This implies increased robustness to sample variability and contextual diversity.

  Structural Cause (Optimization of Transformer Decoder): The pre-norm formulation with RMSNorm ensures that the gradient will propagate more smoothly and that the model made will sustain the same interpretation ability across different subreddit topics.

  Impact: The model's predictions are held consistent even in the face of differing rhetorical styles - confirming the ability of its attention layers to moderate cross-domain sarcasm.

- **Semantic Constraint:**
  Despite Llama in general being a competent model, Llama 3.1-8B still has the Transformer's inherent logical bias - to reduce the many rhetorical tools available in sarcasm to the detection of contradiction, when more complex understanding of its pragmatics is required.

  Structural Cause (Contradiction Heuristic): The model represents textual relationships based mostly on logical consistency checks in the attention space. Subtle irony (feigned ignorance) requires intent modeling, which does not exist in the architecture.

  Impact: Where sarcasm is through cognitive or social irony rather than textual polarity, the model simplifies the model. This accounts for its moderate congruence rate, and the prevalence of "contradiction" reasoning across categories.

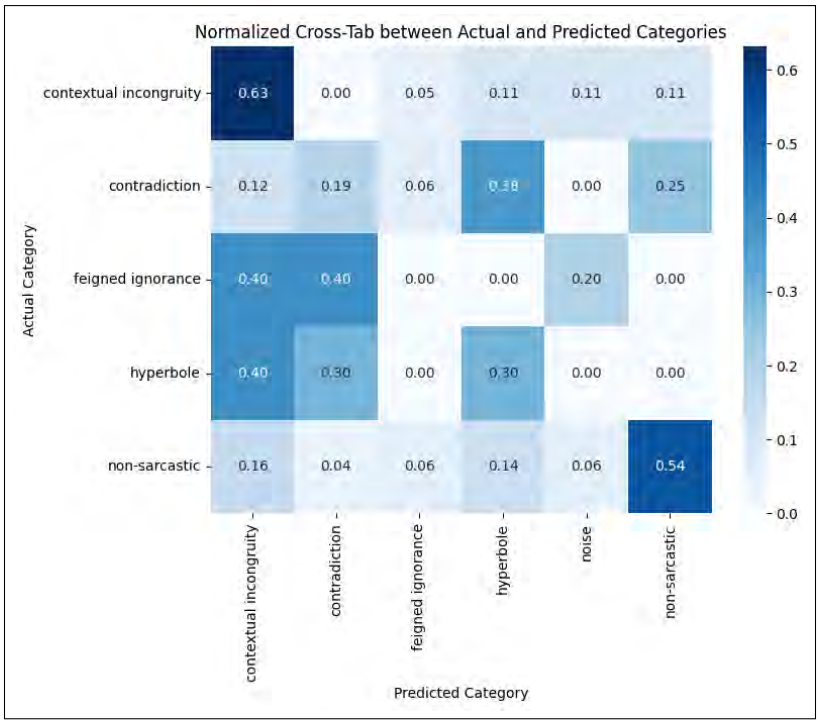


Figure 5.17: Normalized Cross-Tab between Actual and Predicted Categories.

5.2.2 Mistral 7B

Architectural Findings and Impact

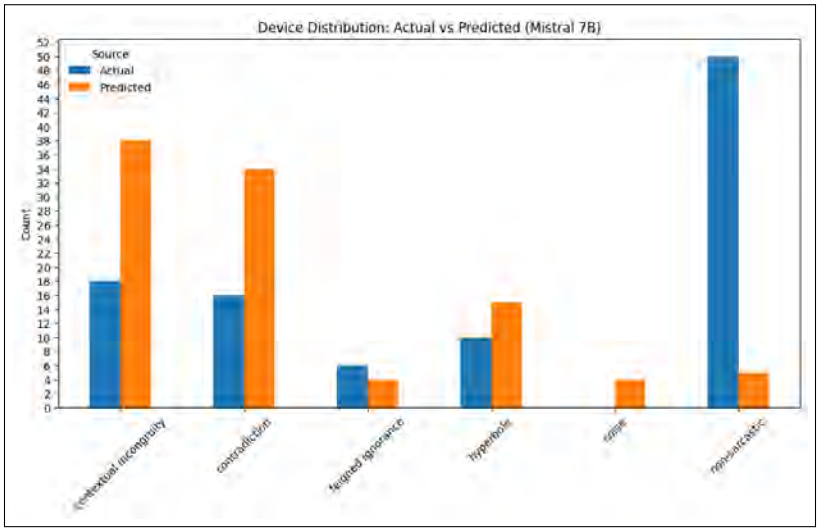


Figure 5.18: Device Distribution: Actual vs Predicted

The success of Mistral 7B is that it uses the advanced attention mechanisms to attain the performance normally enjoyed by models that are twice the size. This enables it to deal with the two-fold challenge of contextual awareness and high-speed inference required to deal with real-time analysis.

Architectural Feature	Functional Role	Trade-off/Impact on task
Grouped-Query Attention (GQA)	Inference Optimization	Inference Efficiency
Sliding Window Attention (SWA)	Contextual Durability	High Recall.
RoPE, SwiGLU, RMSNorm	Foundational Stability	High quality signal flow.

Table 5.16: Feature Specific Impacts

GQA and Inference Optimization: The implementation of Grouped-Query Attention (GQA) being one of the key features for optimizing inference speed and reducing memory usage. As it has 32 Query heads and shares 8 Key-Value heads, Mistral significantly conserves memory bandwidth during decoding. However, in this sarcasm detection task, this efficiency does not translate into high precision. The model's low precision score (0.5376) and high number of false positives indicate that its fine-grained semantic checks are not sufficiently accurate, which lead to many incorrect sarcastic classifications.
SWA and Contextual Recall: SWA and Contextual Recall: Sliding Window Attention (SWA) algorithm works in the sense that the model is capable of long sequence processing, i.e. the full Parent Comment/Subreddit context, at a linear (instead of quadratic) cost of computing it. This kind of design ensures that there is availability of context and that they can be integrated at the cross-layers, which is required in any contextual task such as sarcasm detection. SWA does not affect the structural integrity of the input, and consequently, the high Recall is feasible since it enables one to make reliable long-range contextual comparisons.
Structural Limitation (Rhetorical Collapse): While Mistral 7B can detect incongruity, its rhetorical analysis is limited by a simplified and biased classification system. This leads to a "rhetorical collapse" where the model defaults to certain categories, failing to capture the nuance of human communication. This is evident in two aspects:

- **Hyperbole Over-Projection:** The model shows a strong bias for using hyperbole as a label, despite never correctly identifying a single true instance of it. The confusion matrix shows 0 true positives for hyperbole. Instead, the model frequently mislabels other categories as hyperbole, including 11 non-sarcastic comments. This pattern suggests that the model uses "hyperbole" as a default classification when it detects a high degree of conflict, incorrectly converting more nuanced inconsistencies into simple exaggeration.

- **Feigned Ignorance Blind Spot:** The model does not have a complete blind spot for feigned ignorance, but it represents a significant structural weakness. The data shows its performance is extremely poor, correctly identifying this device in only 1 out of 6 actual instances (a 17% success rate). More often, the architecture fails to process this intricate social signal and devolves it into more basic logical problems, misclassifying it as contradiction (33% of the time) or contextual incongruity (33% of the time).

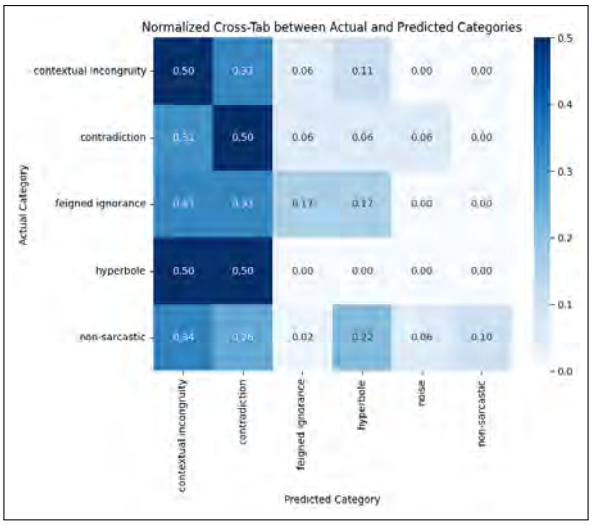


Figure 5.19: Normalized Cross-Tab between Actual and Predicted Categories

5.2.3 Gemma3 12B

Architectural Findings and Impact

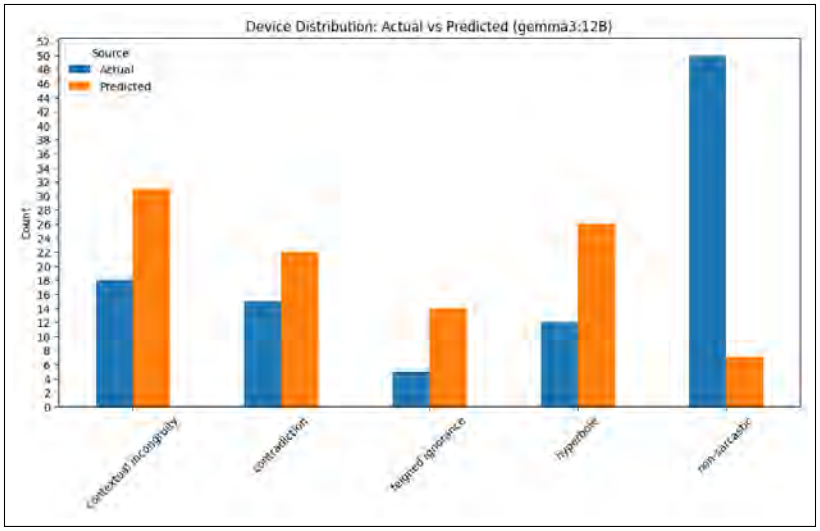


Figure 5.20: Device Distribution: Actual vs Predicted

High-Recall Prioritization:
Gemma12B, as a 12-billion-parameter multimodal transformer, is optimized for high Recall, meaning it prioritizes the detection of sarcasm at the expense of Precision. This bias results in a sensitive model that can detect a broad range of sarcastic comments but also increases the rate of false positives. This sensitivity is a direct result of the model's Group-Query Attention (GQA) architecture, which allows it to capture local and global patterns in both text and images. The long context window of 128K tokens also plays a crucial role in enabling the model to analyze larger chunks of text, giving it a broader context to detect sarcasm, but it also increases the likelihood of misclassifying non-sarcastic comments as sarcastic due to a lack of fine-tuning.

The high Recall (1.0000) reflects that the model successfully identifies all sarcastic comments, but the low Precision (0.5435) reveals that it often flags non-sarcastic comments as sarcastic. This is likely a consequence of the Grouped-Query Attention mechanism and the model's emphasis on capturing broad patterns. While the long context length enables the model to analyze more extended sequences, it may also cause it to overfit to larger patterns of sarcasm that are not always present.
Furthermore, the model's performance on sarcasm devices shows a bias toward detecting Contradiction (80%) and Contextual Incongruity (55.6%), which aligns with its strong ability to capture logical inconsistencies. However, its performance on Hyperbole (58.3%) and Feigned Ignorance (60%) suggests a reductionist approach, where the model oversimplifies complex sarcasm devices into categories it can more easily identify. This may stem from the Rotary Position Embedding (RoPE) mechanism, which enhances the model's ability to process long sequences but also emphasizes certain forms of sarcasm over others.

Impact: While the model is effective at detecting sarcasm, it may flag many non-sarcastic comments as sarcastic, which affects its accuracy and reliability in practical use.

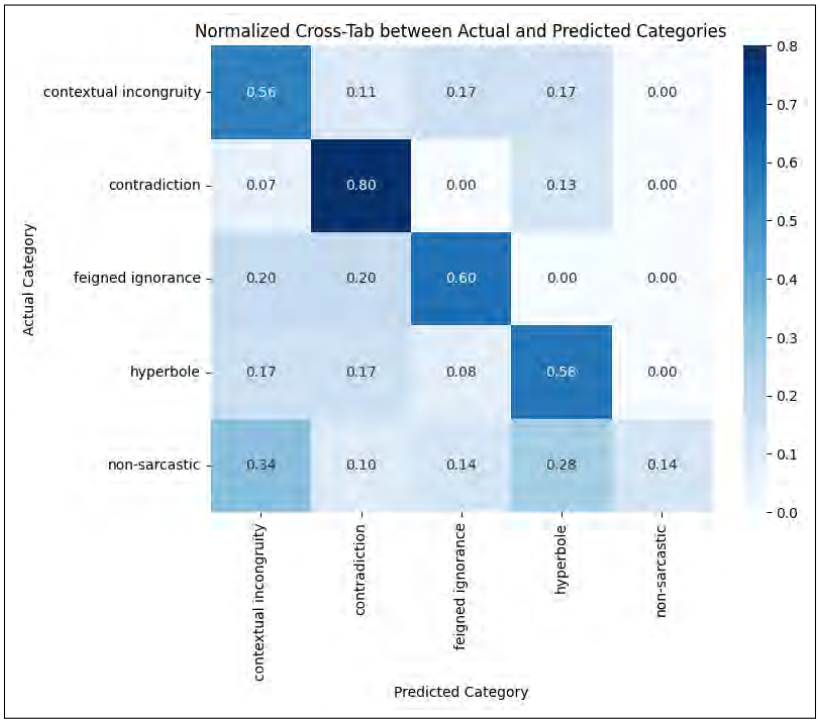


Figure 5.21: Device Distribution: Actual vs Predicted

5.2.4 Qwen3 8B

Architectural Findings and Impact:

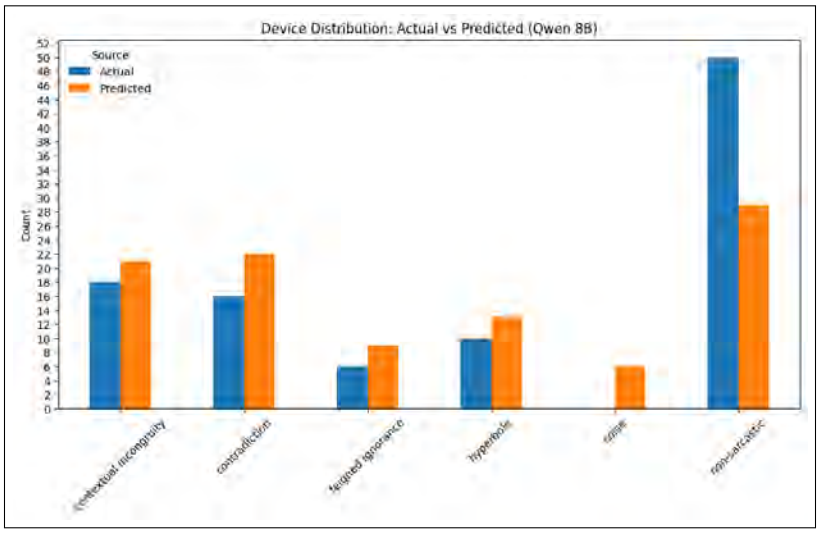


Figure 5.22: Device Distribution: Actual vs Predicted

A high-Recall, generalized incongruity detector is demonstrable in the Qwen3-8B model which follows this dual-dataset analysis. Its architecture:

- **High-Recall Prioritization:** Responds to the information by giving priority to the occurrence of semantic distance (high sensitivity - high Recall). Qwen3-8B architecture has a clear operation requirement to optimize the coverage of detection. The significant difference between its Recall (0.8200) and its Precision (0.5775) proves this.

  Structural Cause (RMSNorm & SwiGLU): RMSNorm used in Pre-Norm formulation boosts training stability and gradient flow enabling the model to be able to learn and remember fine grained contextual clues on deep layers reliably. In addition, the Feed-Forward Networks involve the use of the extremely efficient SwiGLU activation, which offers a more detailed gated signal process.

  Impact: The model is constructed in such a way that it is a highly sensitive detector of incongruity. It is wide net design, so as to minimize False Negatives (not detecting actual sarcasm). This consideration lends credibility to the fact that the basic role of the model is that of generalized sarcasm finding, achieved at the expense of predictive certainty, resulting in high False Positives. One of the general principles to follow is being aware of the potential sarcastic materials, but the Qwen3-8B must be perused by a very qualified individual to remove the non-sarcastic content which the rule wrongly identifies as sarcastic. It is a good rule to recognize possible sarcastic content, but the Qwen3-8B needs to be reviewed by a highly skilled person to eliminate non-sarcastic material the rule incorrectly labels as sarcastic.

- **CoT Congruence Collapse:** Operates upon the data in terms of CoT framework as a justification in post-facto. The Chain-of-Thought (CoT) mechanism

analysis the model is trying to make, the model rhetorical reasoning, shows a structural failure to explain the right reason why it was predicting.

Structural Cause (Functional Dichotomy): The failure is due to the functional dichotomy between the prediction process of the model and the generation process of the model. The Transformer layers (Detection Component) are working effectively to compute a complex vector that represents semantic distance, but the text generation process (Reasoning Component) is unable to convert this vector to the appropriate symbolic rhetorical word.

Impact: According to the model, the corrected congruence rate was only 39.02% (16 of 41 True Positives). This is a low rate and it only goes to show that CoT output is mostly a post-hoc rationalization and not an accurate reflection of the internal decision process. Lower associative layers make the prediction (0 or 1) and then the CoT (Reasoning Component) is forced to provide a justification that is plausible. Evaluating the Qwen3-8B on its numerical measures only (Accuracy, F1) is justified by the fact that its qualitative accounts of why it predicts are architecturally deceptive and cannot be invoked to indicate deep learning. The CoT production is probably an architectural invention that is a post-hoc rationalization. This prediction is carried out in the deepest layers of the association and the language model proceeds to produce the most plausible-sounding but erroneous justification to satisfy the needs of the prompt.

- **Generalization Failure and Structural Instability:** The model has poor ability to generalize as can be seen in the decay in performance between the two test subsets.

  Structural Cause (QKV Bias & RoPE): The design choice of introduction of bias in QKV attention layer is directed to boost extrapolation capability. But this is also what makes it unstable; the model being over-enthusiastic about extrapolation causes it to classify new non-sarcastic incongruity as sarcasm (Killing Precision). Although the RoPE positional encoding allows stable handling of long-range dependencies, the overall threshold of decision making of the model is volatile, demonstrating that the final prediction of this model is unstable in the case the rhetorical nature of the input varies.

  Impact: This model is very vulnerable to the variance of input data which we can call highly data-variant. This fluctuation implies that the acquired habits of sarcasm detection are not completely abstract and applicable to different rhetorical styles. Although the Core Transformer (Detection Component) is mighty, its in-house thresholds of classification are unstable to modifications in the complexity or subtlety of the surrounding information of the input data. The performance of the Qwen3-8B model reported is context-dependent and can be assumed only with a low level of confidence in new or out-of-distribution settings.

- **Semantic Constraint:** The most significant structural bottleneck is the systematic conflation of structural devices in the model, in which complicated rhetorical strategies are reduced to a primitive logical process.

  Structural Cause (Contradiction Heuristic): The model is designed in such a way that it is structurally restricted to the one dominating heuristic: "Is there

a contradiction? This is a direct result of the design of the Transformer, which treats text through its closest logical structure. Such devices as Hyperbole are computationally hard to tell apart with a simple Contradiction.

Impact: In case of successful uncovering sarcasm which is attained through subtle tools, say Hyperbole or Feigned Ignorance, the symbolic reasoning aspect of the model always narrows down to the overall tool of Contradiction. It automatically reduces to Contradiction due to its structure as an architecture that is internally a consistency checker, degrading all fine-grained rhetorical devices to the lowest common denominator of logical contradiction. This simplistic state of affairs is made possible by the architecture being able to trace this effect of the sarcasm (semantic inconsistency) but not the fine-grained rhetorical intelligence which recognizes the precise cause.

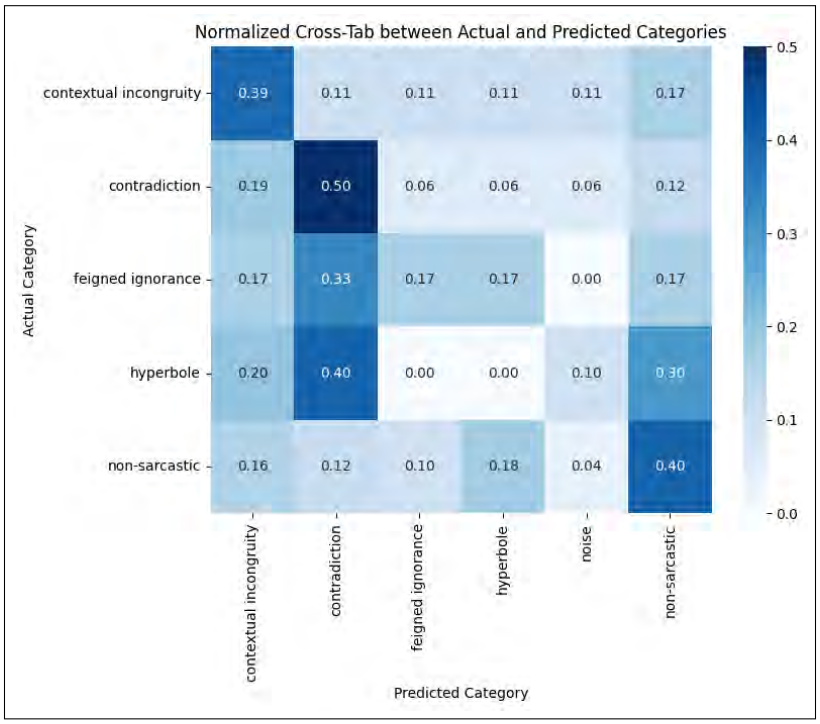


Figure 5.23: Normalized Cross-Tab between Actual and Predicted Categories

5.2.5 DeepSeek-R1-14B

Architectural Findings and Impact

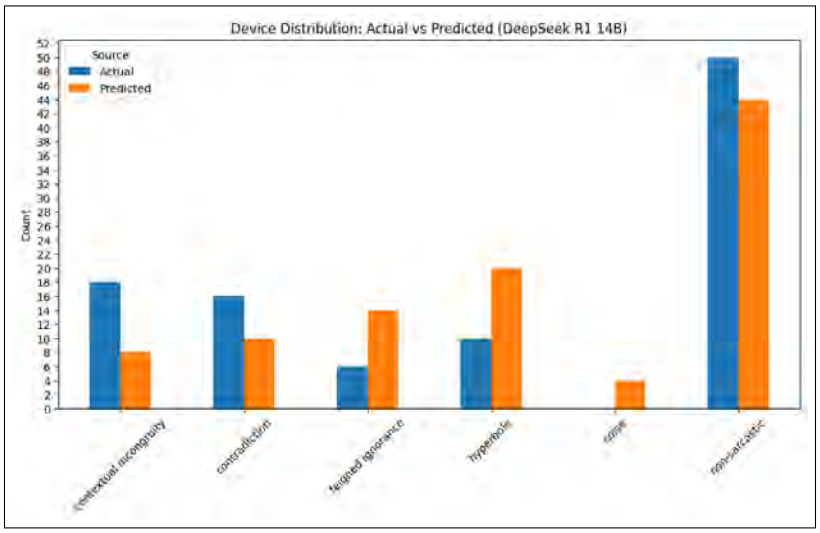


Figure 5.24: Device Distribution: Actual vs Predicted

SwiGLU Activation (Polarity Sensitivity)

SwiGLU's dual-gated structure enhances nonlinear representation of sentiment intensity, allowing simultaneous encoding of positive and negative effects. This underlies the model's relatively strong hyperbole accuracy (0.50), where exaggeration hinges on polarity contrast. However, its emotional bias drives false positives, as enthusiastic language without sarcasm may still trigger activation patterns similar to ironic tone.

Rotary Position Embeddings (RoPE) (Context Retention)

RoPE encodes relative token distances, preserving contextual relationships between a comment and its parent. This aids continuity and literal coherence, explaining the 0.60 accuracy in non-sarcastic cases. Yet, because RoPE models spatial rather than pragmatic relations, it misses the social incongruity behind context-based irony, producing the weak 0.17 contextual-incongruity accuracy.

RMS Normalization (Recall Amplification)

RMSNorm stabilizes gradient magnitude, creating uniform activation scaling across layers. This architectural smoothing improves recall (0.72) by detecting nearly all tonal anomalies but simultaneously reduces precision (0.6406) through oversensitivity. It also supports literal-case robustness by damping activation variance when context and tone align.

Dense Distillation from MoE (Loss of Pragmatic Depth)

Distilling DeepSeek-R1's original Mixture-of-Experts into a dense 14 B model removed specialized expert pathways once responsible for pragmatic and discourse

reasoning. Consequently, rhetorical understanding, especially contradiction (0.13) and feigned ignorance (0.33), collapsed into shallow semantic heuristics. The model tends to equate any form of semantic opposition with sarcasm, demonstrating semantic compression typical of dense distillation.

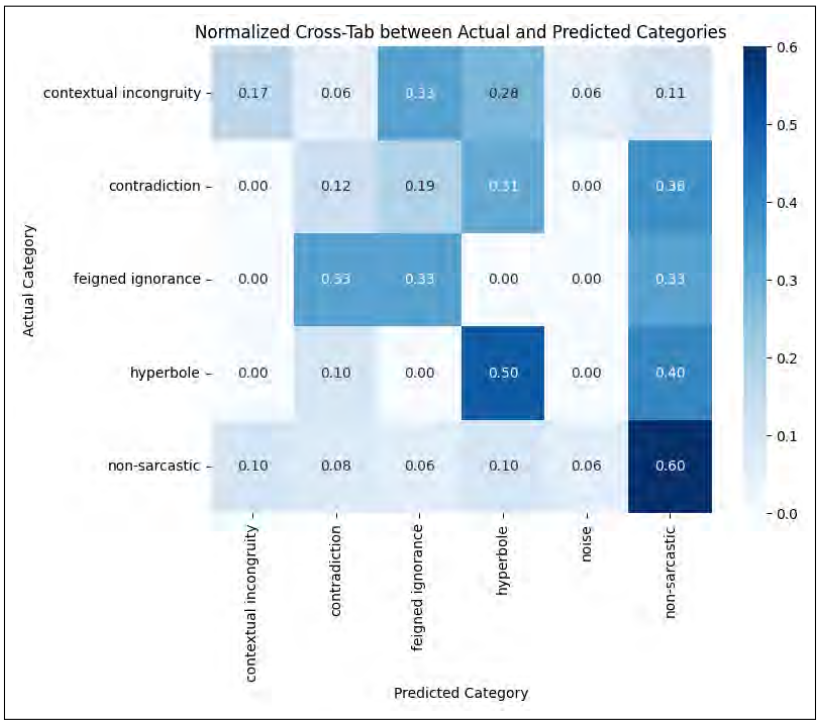


Figure 5.25: Normalized Cross-Tab between Actual and Predicted Categories

5.3 Final Design Adjustments

After the first zero-shot prompting scheme, there was a major amendment to improve the reasoning clarity and signal particularity of the sarcasm detecting procedure. The previous structure was more concerned with output formatting and mechanical readability; it was not as deep in interpretation to do qualitative analysis. The updated prompt thus added a definite Chain-of-Thought (CoT) scheme; it was a four-step rationalization model that simulates systematic human reasoning.

```
system_message_prompt = """
You are a highly accurate sarcasm detection AI.
Your task is to analyze a comment by thinking through it step-by-step to
    determine if it is sarcastic.

Your reasoning MUST follow a clear, sequential Chain-of-Thought process
    to arrive at the classification. Do not state the conclusion first.
    Structure your explanation using the following four steps:
1. **Context Analysis:** What is the context from the subreddit and the
    parent comment?
2. **Comment Analysis:** What is the literal meaning of the user's
    comment?
3. **Sarcasm Signal Analysis:** Is there a contradiction, hyperbole,
    feigned ignorance, contextual incongruity, or another sarcastic
    signal? ONLY IDENTIFY ONE SIGNAL.
4. **Conclusion:** Based on the signal identified in Step 3 (or lack
    thereof), is the comment sarcastic?


Your final output MUST be a single, valid JSON object and nothing else.

---
CRITICAL JSON FORMATTING RULES:
1. **Escape Double Quotes:** Any double quotes (") inside the
    "explanation" string MUST be escaped with a backslash, like this: \".
2. **Do NOT add new lines (\n) in the explanation**: Keep the entire
    explanation on a single line.
3. **Do NOT add the "```json" string before the JSON object
---

The JSON object must contain two keys:
1. "explanation": A string containing your four-step Chain-of-Thought
    reasoning.
2. "sarcasm_classification": An integer: 1 if the comment is sarcastic, 0
    if it is not.

Example of a valid response that follows all rules:
{
  "explanation": "Step 1: Context Analysis - The discussion is in the
      r/Metal subreddit, where users are expected to be knowledgeable
      about metal bands. The parent comment is about As I Lay Dying
      (AILD). Step 2: Comment Analysis - The user's comment literally
      states they were unaware that AILD is a metal band. Step 3:
```

```
    Incongruity Detection - There is a strong incongruity here. AILD is
    a very prominent and well-known band within the metalcore genre.
    Claiming ignorance of this fact within an expert community is
    highly improbable. This points to feigned ignorance. Step 4:
    Conclusion - The act of feigning ignorance about a core topic
    within a niche community is a classic form of sarcasm. Therefore,
    the comment is sarcastic.",
  "sarcasm_classification": 1
}
"""
```

Listing 5.1: Adjusted System message prompt used for model instruction

```
user_prompt_template = """
Analyze the following content:
- Subreddit: r/{subreddit}
- Parent Comment: "{parent_comment}"
- Comment: "{comment}"
"""
```

Listing 5.2: User prompt template

The system message prompt has also been redesigned to make the model do the sarcasm classification by an analytical reasoning process step by step instead of giving a direct classification. These four required steps were as follows:

Context Analysis: The model initially processes the subreddit and parent comment, and uses them to do situational and cultural inferences.

Comment Analysis: This then breaks down the literal meaning of the comment made by the user.

Sarcasm Signal Analysis: The model discovers a single linguistic or pragmatic source of sarcasm - contradiction, hyperbole, feigned ignorance or contextual incongruity. This limitation is what introduces signal-level specificity, which reduces the overgeneralized thought.

Lastly, the model is the summation of the arguments that come before it to conclude that the comment is sarcastic or not. The JSON specification was further extended to a two-key schema to make the output machine readable:
"explanation" containing the entire, flattened Chain-of-Thought logic in a single line (neither line breaks nor unescaped quotes), and
"sarcasm_classification" the last binary choice (1= sarcastic, 0= non-sarcastic).
The main interest in the presentation of the CoT component was to get the internal reasoning of the model on the field of explanation. This enabled the researchers to qualitatively evaluate the way the model was identifying sarcasm - by ensuring that its labelling matched to a reasonably sensible explanation or it was just a case of a pattern following superficially. Through these generated explanations it was possible to ascertain whether a model deduced sarcasm on the right grounds i.e. right identification of contradiction or feigned ignorance as opposed to accidental lexical correlations.

This structural improvement added a trace of reasoning that was interpretable and still had the strict formatting that was required by batch evaluation and mechanical parsing. JSON sanitization rules (escaped quotes, no newline characters) were added to ensure the compatibility of the downstream evaluation scripts and prevent parsing errors of various open-source LLMs.
In theory, this change made the experiment a guided reasoning task rather than a pure zero-shot classification task, which interpolated the quantitative accuracy measure with the qualitative interpretability measure. With a standardized line of reasoning, it was now possible to evaluate not only whether models were doing the right thing detecting sarcasm, but how and why they made the decision, giving much more insight on how models behave and how architectures reason.

Model	**Prompt Strategy**	**Accuracy**	**Precision**	**Recall**	**F1-Score**
LLaMA 3.1 8B	JSON + Context + CoT	64.0%	60.00%	84.00%	70.00%
Mistral 7B	JSON + Context + CoT	59.57%	56.82%	**100.00%**	72.46%
Qwen3 8B	JSON + Context + CoT	68.00%	62.86%	88.00%	73.33%
Gemma3 12B	JSON + Context + CoT	59.18%	55.56%	**100.00%**	71.43%
Deepseek R1 14B	JSON + Context + CoT	70.00%	65.62%	84.00%	73.68%

Table 5.17: Model Performance Comparison on Sarcasm Detection (After Adjusted Design)

Coding Sheet Preparation: Model outputs were grouped into a new CSV file to enable a detailed assessment in which the following columns were included: subreddit, parent comment, comment, original label, predicted label, explanation, model raw response. The data were then coded and scores for each of the four-item groups on a separate coding sheet with longer columns to be analyzed by hand: subreddit, parent comment, comment, original label, predicted label, explanation, model raw response, prediction category, actual category, key evidence of explanation, response leaked, explanation leaked. The actual category column was manually labelled and the labels used were the sarcasm devices (e.g., hyperbole, contradiction, feigned ignorance, contextual incongruity) through contextual interpretation of parent comment and comment. Model explanations were automatically extracted into predicted categories which could be compared to evaluate model reasoning, interpretability and reliability in classification. The coding sheets maintained a balanced distribution of labels as well.

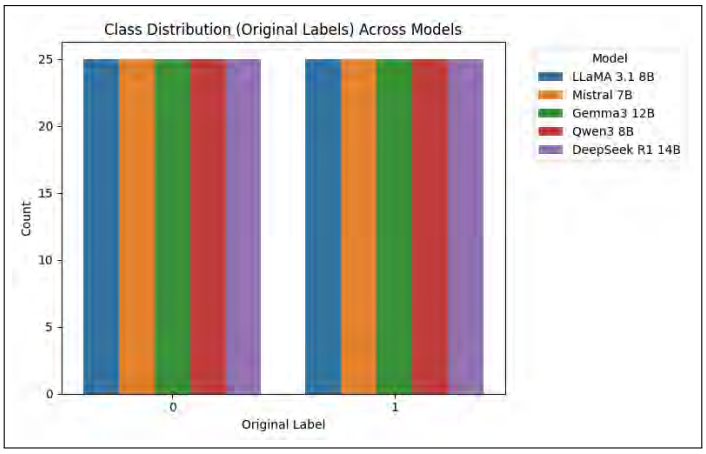
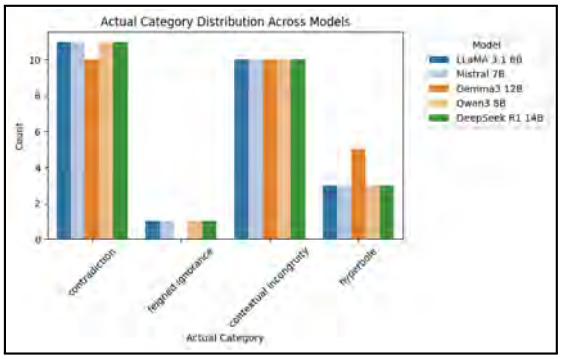
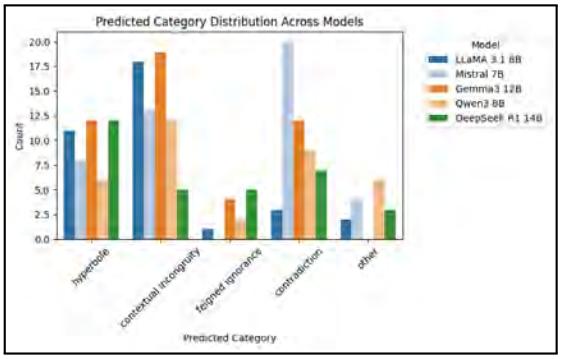


Figure 5.26: Balanced Distribution of Original Labels



(a) Distribution of Actual Sarcasm Devices Across all Models



(b) Distribution of Predicted Sarcasm Devices Across all Models

Figure 5.27: Comparison between Distribution of Actual & Predicted Sarcasm Devices

5.4 Statistical Analysis

5.4.1 Llama 3.1 8B

Architectural Stability and Data Sensitivity

Metric	Dataset 28 (N = 50)	Dataset 29 (N = 50)	Δ (Difference)
Accuracy	0.64	0.72	+0.08
Precision	0.60	0.69	+0.09
Recall	0.84	0.80	-0.04
F1-Score	0.70	0.74	+0.04

Table 5.18: Performance Metrics for Datasets 28 and 29

The small variance of the architecture (8 percentage points in accuracy) shows reasonable architectural stability.Unlike Qwen3-8B, which demonstrated significant decay in performance when subsets are used, Llama 3.1-8B exhibits relatively stable performance across the subsets. This stability indicates that it can generalize sarcasm patterns successfully from its training on various Reddit-like and conversation corpora, although not perfectly- it still fluctuates with the input's rhetorical complexity.

5.4.2 Mistral 7B

Architectural Stability and Data Sensitivity

Metric	Dataset 31 (N = 50)	Dataset 32 (N = 50)	Δ (Difference)
Accuracy	0.5957	0.5102	-0.0855
Precision	0.5682	0.5102	-0.0580
Recall	1.0000	1.0000	0.0
F1-Score	0.7246	0.6757	-0.0489

Table 5.19: Performance Metrics for Datasets 31 and 32

There is a small but noticeable decline in performance between Dataset 31 and Dataset 32, indicating that the model is not perfectly stable across different datasets. The slight drop in Accuracy, Precision, and no drop in Recall reflects a similar behavior with their combined performance. Though the accuracy (both combined and separate) may be quite low compared to most models, the small difference between dataset suggests that the model is stable between varying datasets.

5.4.3 Gemma3 12B

Architectural Stability and Data Sensitivity

The model's performance across two distinct datasets (Dataset 35 and Dataset 36) shows the extent of its stability and generalizability when exposed to different sarcasm contexts.

Metric	Dataset 35 (N = 50)	Dataset 36 (N = 50)	Δ (Difference)
Accuracy	0.5600	0.5918	+0.0318
Precision	0.5319	0.5556	+0.0237
Recall	1.000	1.000	0.0
F1-Score	0.6944	0.7143	+0.0199

Table 5.20: Performance Metrics for Datasets 35 and 36

There is a small but noticeable **increase** in performance between Dataset 35 and Dataset 36, indicating that the model is not perfectly stable across different datasets. The slight drop in Accuracy, Precision, and no drop in Recall reflects a generalization weakness, where the model faces challenges in maintaining consistency when exposed to more complex, subtle, or varied sarcasm. Though the accuracy (both combined and separate) may be quite low compared to most models, the small difference between dataset suggests that the model is stable between varying datasets.

5.4.4 Qwen3 8B

Architectural Stability and Data Sensitivity

The results of the comparison between the two distinct, balanced data subsets show the extent of architectural stability of the Qwen3-8B model, which is essential in making a generalization.

Metric	Dataset 38 (N = 50)	Dataset 39 (N = 50)	Δ (Difference)
Accuracy	0.6800	0.5400	-0.1400
Precision	0.6286	0.5278	-0.1008
Recall	0.8800	0.7600	-0.1200
F1-Score	0.7333	0.6230	-0.1103

Table 5.21: Performance Metrics for Datasets 38 and 39

The 14 point-drop in Accuracy between Dataset 38 and Dataset 39 indicates that it contains a critical architectural weakness to sample variability. The success of the model is not evenly strong but strongly affected by the nature and the complexity of the input data in context. This shows that although the Transformer architecture is good in capturing the overall concept of sarcasm, its consistency in performance is indicative of a generalization weakness faced when more difficult, subtle, or more varied contextual incongruities are faced in Dataset 39. Thus, the Qwen3-8B model has architectural constraints with regards to its out-of-distribution stability of this complex language task.

5.4.5 DeepSeek-R1-14B

Architectural Stability and Data Sensitivity

Metric	Dataset 40 (N = 50)	Dataset 41 (N = 50)	Δ (Difference)
Accuracy	0.7000	0.6200	-0.0800
Precision	0.6562	0.6250	-0.0312
Recall	0.8400	0.6000	-0.2400
F1-Score	0.7368	0.6122	-0.1246

Table 5.22: Performance Metrics for Datasets 40 and 41

The table above clearly illustrates an 8-point drop in Accuracy and a significany 24-point drop in Recall across datasets demonstrate mild architectural instability. This sensitivity is likely to be arised from the QKV Bias mechanism in the attention layer, when rhetorical style or subtlety changes. While RoPE stabilizes long-range dependencies between comment and parent comment, the dense, non-expert architecture of DeepSeek-R1-14B shows diminished adaptability to varied sarcastic expressions.

5.5 Comparisons and Relationships

5.5.1 Llama 3.1 8B

The relationship between the Actual Sarcasm Category (manually-labeled) and the Predicted Category (based on a CoT explanation from the model) from the coding sheets shows how well the model is reasoning about why a comment is sarcastic.

Overall Reasoning Congruence

Out of all the True Positive predictions, the CoT explanation was consistent with the correct manually labeled sarcasm device in 43.90% of cases.This moderate congruence suggests that the CoT explanations of Llama 3.1-8B partially capture a true understanding of the reason, as opposed to an erratic post-hoc rationalization.

Once in a while, the model places its reasoning successfully on the right rhetorical sign, especially if the sign is a hyperbole or a direct contradiction but it inevitably fails with subtle pragmatic signs. detected.

Metric	Count	Rate
Total True Positives	41	-
Matching Device Labels	18	-
Consistency Rate	-	43.90%

Table 5.23: CoT Congruence Collapse Analysis

Dominating Predicted Structures: The reasoning of the model comes down very much to lexical and semantic exaggeration (Hyperbole) and surface-level Contradiction as its major explanatory strategies. This implies that the architecture takes advantage of powerful token-level embeddings and sentiment polarity-based attention, which is effective at identifying overt cases of irony but reduces the complexity of more abstract rhetoric cases.

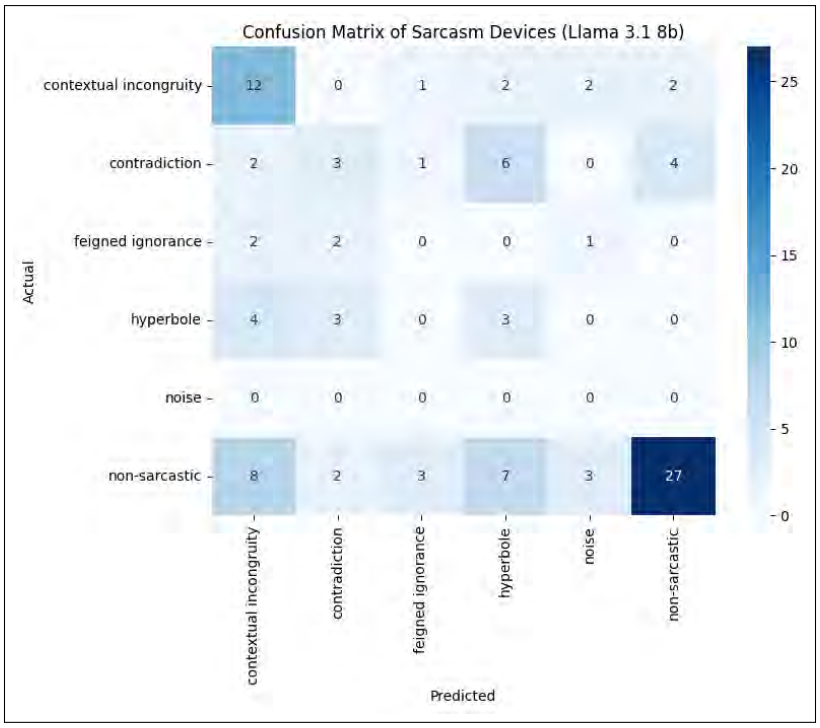


Figure 5.28: Confusion matrix between actual and predicted sarcasm devices.

Llama 3.1-8B shows partial conflation of nuanced devices into the simpler logical structure of contradiction.When the sarcasm is spoken with feigned ignorance or with a subtle form of hyperbole, then the model simplifies it to a contradiction pattern.This is due to a semantic compression bias-the architecture is able to see an inconsistency but not the rhetorical motivation for it.

5.5.2 Mistral 7B

Reasoning Congruence: Chain-of-Thought (CoT) analysis explores the fidelity of the model's internal reasoning by examining its 50 True Positives and comparing the predicted sarcasm device against the actual one.

Metric	Count	Rate
Total True Positives	50	-
Matching Device Labels	18	-
Consistency Rate	-	36.00%

Table 5.24: CoT Congruence Collapse Analysis

The analysis revealed a Consistency Rate of 36.00%, with the model's predicted device label matching the actual label in only 18 of the 50 successful predictions. This score indicates that in significantly less than half of its correct classifications, the reasoning provided by Mistral 7B aligned with the manually identified reason. This low consistency suggests a notable disconnect between the model's final output and its internal logic, pointing to a potential weakness in its reasoning process rather than a consistently clear and acceptable decision path

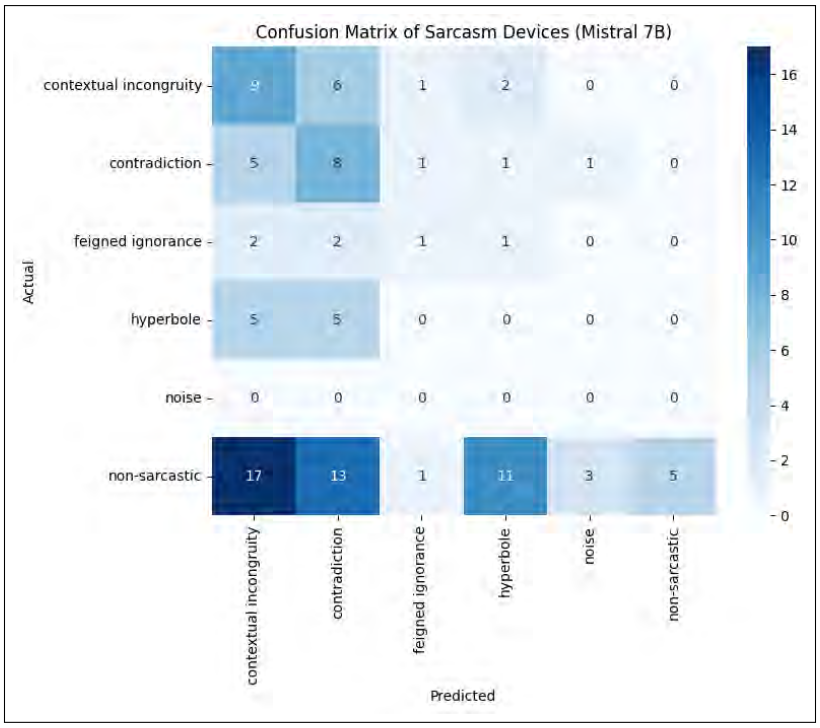


Figure 5.29: Confusion matrix between actual and predicted sarcasm devices.

5.5.3 Gemma3 12B

CoT Congruence Collapse:
The Chain-of-Thought (CoT) reasoning in Gemma12B is not consistent enough to be aligned with the actual sarcasm categories detected by the model. The updated analysis of the True Positive Category Consistency reveals that out of the 50 True Positive sarcastic comments, 32 had matching sarcasm device labels. This results in a consistency ratio of 64%.

Metric	Count	Rate
Total True Positives	50	-
Matching Device Labels	32	-
Consistency Rate	-	64.00%

Table 5.25: CoT Congruence Collapse Analysis

The CoT outputs frequently serve as post-hoc justifications rather than accurate reflections of the underlying reasons for the sarcasm detection. This relatively low consistency ratio suggests that while the model can correctly classify sarcasm, it struggles to consistently rationalize why certain sarcasm devices were identified, especially for more complex or subtle sarcasm types. Although the consistency rate is quite higher than other models.

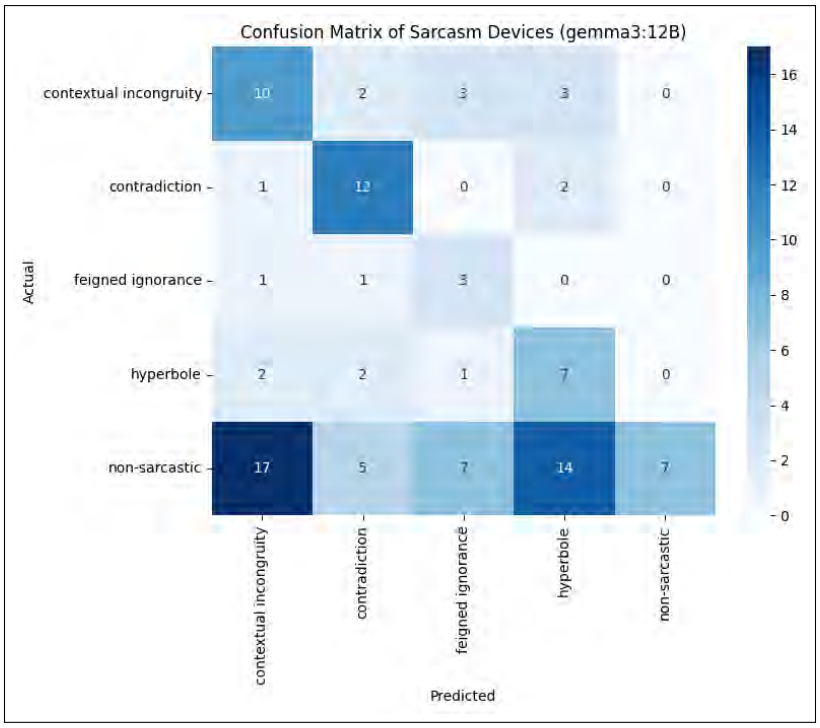


Figure 5.30: Confusion matrix between actual and predicted sarcasm devices.

5.5.4 Qwen3 8B

The architecture of the Qwen3-8B is best understood according to the connection between the Actual Sarcasm Category and the device mentioned by the model in its CoT-based explanation. This provides an answer to the question of whether the model is predicting the right reason.

Overall Reasoning Congruence: In the case of the 41 instances of properly characterized sarcastic (True Positives, TP) correctly identified, the model-reported reason in the CoT description was only congruent with the manually-labeled Actual Category 39.02% of the time. This is because this low congruence rate confirms the CoT mechanism of the model is more of a post-hoc rationalization layer, that is, the model produces a viable explanation that has already been determined by the lower-order detecting mechanism that the sarcastic intent has been detected.

Metric	Count	Rate
Total True Positives	41	-
Matching Device Labels	16	-
Consistency Rate	-	39.02%

Dominating Predicted Structures: Two very similar concepts also provided the majority of the model predictions: Contradiction (22 times) and Contextual Incongruity (21 times). These two machines, based on the most immediate and visible kinds of semantic decomposition, are obviously the architectural choice in the reasoning of the Qwen3-8B.

Architectural Conflation: The most profound structural limitation of the model can be seen in the way it manages subtle devices of sarcasm: Hyperbole and Feigned Ignorance. This tendency shows a steady architectural conflation. When the sarcasm itself is accomplished by Hyperbole (exaggerated or faked sincerity) or Feigned Ignorance (false naivete), the Qwen3-8B model does not have the finer semantic detail to tell the difference between the particular rhetorical trope. Rather, the logical reasoning aspect of the model simplifies the whole phenomenon to the simplest logical inconsistency that it is capable of handling-the general principle of Contradiction.

Therefore, though the LLM has sufficient training to identify the signal of sarcastic intent due to contextual cues (the role of its high Recall) its symbolic reasoning abilities are weak. The model realizes that the comment is incorrect or inconsistent with the context, but it has no ability to distinguish on how it is incorrect (i.e. it is incorrect due to exaggeration, naivete, or direct opposition). In the case of the Qwen3-8B, the arsenal of the rhetorical tools of sarcasm is reduced to a single, overriding logical action: Is there a contradiction?

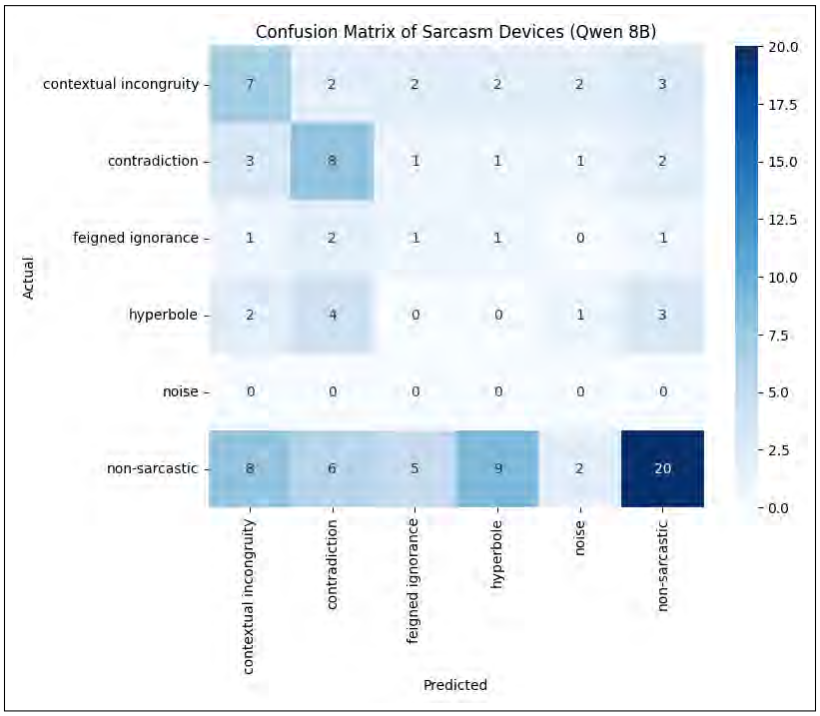


Figure 5.31: Confusion matrix between actual and predicted sarcasm devices.

5.5.5 DeepSeek-R1-14B

CoT Congruence Collapse

DeepSeek-R1-14B operates upon the data through a Chain-of-Thought (CoT) framework that functions largely as a post-facto justification mechanism for its predictions. While the CoT explanations exhibit logical and structured reasoning, analysis of the coding sheets reveals that this reasoning is not always perfectly aligned with the rhetorical device being classified.

Metric	Count	Rate
Total True Positives	36	-
Matching Device Labels	12	-
Consistency Rate	-	33.33%

Table 5.26: CoT Congruence Collapse Analysis

This obvious incongruence arises from reasoning homogenization, which is likely to be introduced during distillation; the dense model frequently reuses similar justification templates (e.g., "The literal meaning contradicts the context") across multiple rhetorical forms. Out of all sarcastic samples, only 33.33% demonstrated clear reasoning label congruence where the explanation aligned with the actual device, while 66.67% showed CoT incongruence cases where the model correctly recognized semantic irony but misidentified the underlying rhetorical device (such as labeling feigned ignorance as contradiction).
Consequently, DeepSeek-R1-14B's reasoning remains semantically coherent yet rhetorically inconsistent: it detects sarcasm reliably but often explains it through the wrong rhetorical lens.

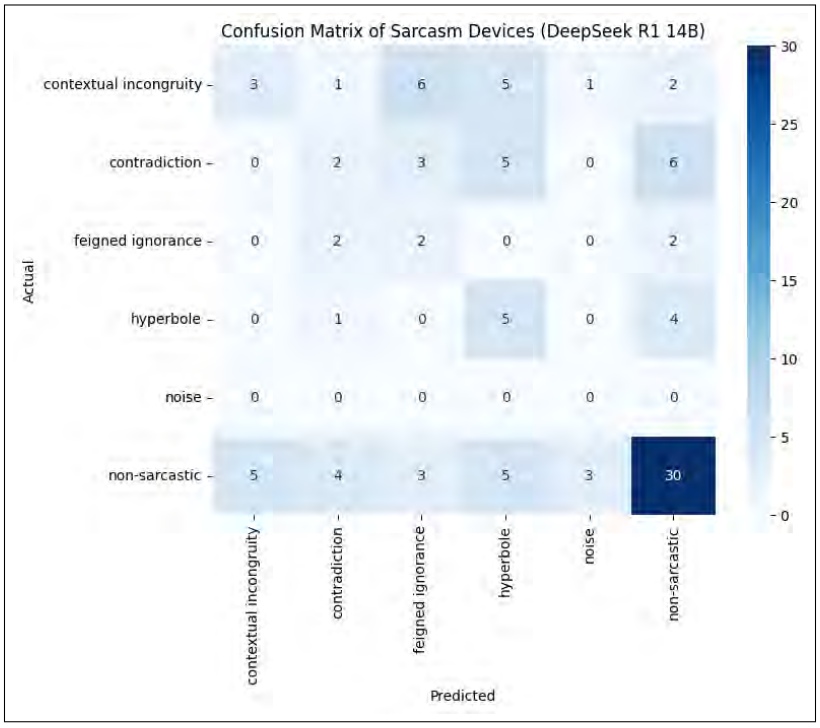


Figure 5.32: Confusion matrix between actual and predicted sarcasm devices.

5.5.6 Per-device Performance comparison

Below is the grouped bar chart for all 5 models "per-device" performance.

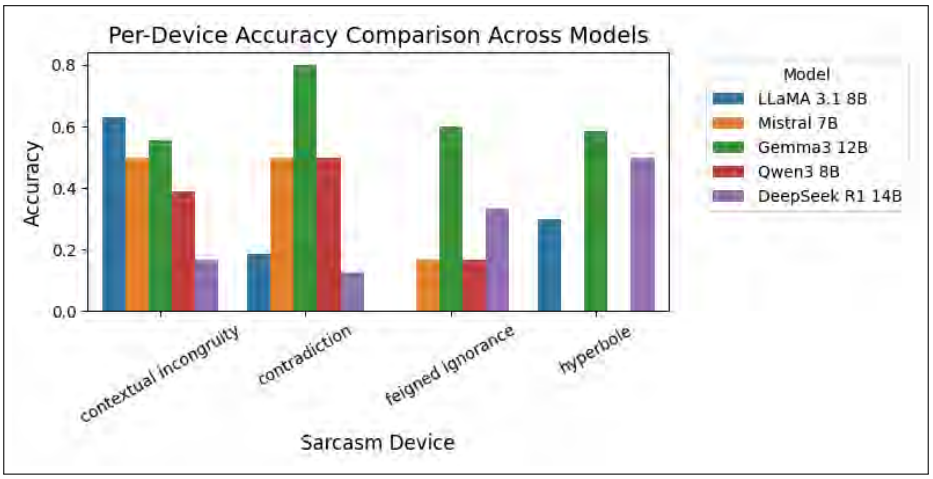


Figure 5.33: Per-device Performance

5.6 Discussion

5.6.1 Llama 3.1 8B

The Llama 3.1-8B model functions as a well-balanced, contextually aware sarcasm classifier. The Recall and Precision trade-off is tighter than Qwen3-8B, indicating improved stability and pragmatic calibration. In addition, Llama 3.1-8B tends to reason through overt lexical and logical cues and often resolves subtle rhetorical devices through contradiction. In terms of architecture, Llama 3.1-8B can be described as a RoPE-enhanced contextual incongruity recognizer, with this mid-scale language model:

- Stability across datasets is acceptable
- Reliable at detecting hyperbole and explicit contradiction
- Partial (though still improved) Chain-of-Thought congruence ($\approx 41\%$)
- Residual bias toward logical rather than cognitive sarcasm forms.

Considering the language models analyzed here, Llama 3.1-8B has a remaining bias toward logic, as opposed to cognitive sarcasm forms. Overall, Llama 3.1-8B can position itself as a linguistically competent (although pragmatically limited) sarcasm analyzer, having an output that will evaluate correct results in most instances, but not always evaluate correct results for the correct reasons.

5.6.2 Mistral 7B

Mistral 7B is a high performance, incongruity detector that is architecturally optimized. It is notable for its high efficiency and exceptional detection sensitivity (Recall). Application of GQA and SWA in it is effective in overcoming the parameter constraint leading to a balanced and very efficient predictive model. However, this specialized performance profile reveals a critical trade-off. The model achieves its perfect recall at the direct expense of precision, resulting in a highly imbalanced predictive model that generates a large volume of false positives. Furthermore, its rhetorical analysis is structurally simplified. This leads to a "rhetorical collapse" where the model fails to classify nuanced sarcasm types accurately, exhibiting significant biases like Hyperbole over-projection and weaknesses like the Feigned Ignorance blind spot. Ultimately, Mistral 7B is a powerful but blunt instrument for this task. Its architecture prioritizes efficiency and ensures no sarcastic instance is missed, but this comes at the cost of low precision and a failure to capture the subtleties of human rhetoric.

5.6.3 Gemma3 12B

Gemma12B operates as a highly sensitive sarcasm detection model. Its perfect Recall score of 1.0000 ensures that no sarcastic instances are missed, making it a powerful tool for exhaustive detection. However, this high sensitivity comes at the cost of low Precision (0.5435), which stems from a significant weakness in identifying non-sarcastic comments (0.14 accuracy). This results in a high number of false positives, where neutral text is misclassified as sarcastic.

A notable strength is the model's explanatory power. When it correctly identifies a sarcastic comment (a True Positive), it also classified the specific sarcastic device with 64% accuracy (32 out of 50 instances). It shows particular strength in identifying complex forms like contradiction (80% accuracy). Furthermore, the model demonstrates strong architectural stability, maintaining consistent performance metrics across different datasets, which indicates reliable generalization.

5.6.4 Qwen3 8B

The Qwen3-8B is an architectural high-Recall, generalized incongruity detector. It has performance which can be described as an inherent trade-off between detection sensitivity and predictive accuracy and stability. The fact that it relies on Contradiction as the primary explanatory mechanism in the CoT and in particular in analyzing Hyperbole and Feigned Ignorance is a crucial discovery that demonstrates that the model has an architectural structure that constrains its ability to provide fine-grained, human-level rhetorical analysis.

5.6.5 DeepSeek-R1-14B

The DeepSeek-R1-14B architecture demonstrates a clear trade-off between semantic breadth and pragmatic precision. SwiGLU and RMSNorm make it an efficient semantic detector, high-recall, emotionally sensitive, and contextually stable. Yet, the loss of Mixture-of-Experts specialization reduces its ability to distinguish nuanced devices: only 33.33% of sarcastic cases showed reasoning label congruence, while 66.67% revealed Chain-of-Thought incongruence.
In essence, DeepSeek-R1-14B reliably recognizes that sarcasm exists but often misinterprets why, showing how distillation-driven architectural simplification preserves semantic stability at the cost of pragmatic precision in humor understanding.

5.6.6 Overall Performance Comparison (All Model)

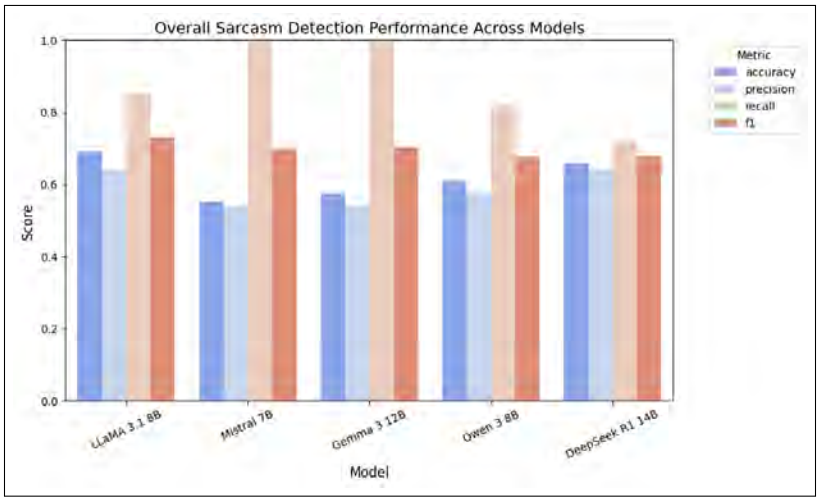


Figure 5.34: Overall Performance Grouped

Chapter 6

Conclusion

6.1 Summary of Findings

Llama 3.1-8B

Llama 3.1-8B provides stable and tight Recall/Precision coverage, balancing between Recall and Precision. It detected sarcasm mostly on lexical and logical signals, especially on explicit forms like hyperbole and contradiction. Though the model is still slightly biased towards logical over cognitive sarcasm and, despite moderate Chain-of-Thought congruence (41%), it fails to consistently interpret sarcasm, especially of subtler cues.

Mistral 7B

Mistral 7B has excellent Recall which comes at the cost of low Precision, therefore including a large number of false positives. It is determined to be an efficient and a sensitive approach for the sarcasm detection but lacks in a subtle rhetorical analysis. This architecture struggles from falling into rhetorical collapse, since it failed to correctly resolve sarcasm devices like "Feigned Ignorance" and tends to over-generate hyperbole. It has a heavy-foot but effective tool for identifying sarcasm.

Gemma3 12B

The Gemma3 12B model has a high Recall (1.0000), which makes it quite "sarcastic happy" (i.e. doesn't tend to "not" predict sarcasm), but at the cost of low Precision (0.5435). The model is stumped by other, non-sarcastic comments, but when there is sarcasm to be found, it can label with 64% accuracy the nature of that sarcasm type the model especially effective in detecting contradiction (80%). It maintains architectural stability which is showing strong generalization across datasets.

Qwen3-8B

We have a model Qwen3-8B dedicated for high Recall in general sarcasm detection but its main explanatory attentional mechanism (i.e., Contradiction) restricts detailed rhetorical analysis. It fails to distinguish between forms of sarcasm, such as Hyperbole and Feigned Ignorance, precluding delicate applications for complex

sarcasms. The performance of the model indicates a hidden trade-off between sensitivity and specificity detection.

DeepSeek-R1-14B

We are presenting the deep learning model DeepSeek-R1-14B for high-recall detection of sarcasm, with attention to semantic consistency and emotion awareness. It applies SwiGLU and RMSNorm, which makes it both contextually stable and efficient. But it has problems handling subtle sarcasm, reflected in its low (33.33%) chain-of-thought congruence and high (66.67%) incongruence. It can recognize sarcasm but uses in a mistaken way the motivation behind it, pointing to a trade-off between semantic scope and pragmatic accuracy.

6.2 Contribution to the Field

In the present thesis, we have made a significant contribution to the domain of sarcasm detection and analysis of large language models (LLMs). First of all, we employed prompt engineering techniques, we tried various prompts multiple times to evaluate the performance of the model, and it allowed us to optimize it and learn the peculiarities of the sarcasm recognition in various LLMs. Upon this, we have developed a complete structure of sarcasm detection analysis in the real world comments through excel. In this model, sarcasm in the real-life text is identified, the identification performed by the sarcasm tool, as interpreted by the model output and the comparison of the output with the structural findings of the internal logic process of the model. More specifically, we discovered its strong suits, in its ability to pick up overt sarcasm (e.g. hyperbole, contradiction, context incongruity and feigned ignorance) and the weaknesses, i.e. that it is still biased towards logical sarcasm. As part of this comprehensive analysis, we have given a robust method to evaluate and improve the interpretability and accuracy of sarcasm detection models that is pointing to the balance of Recall and Precision and the Chain-of-Thought agreement in LLCs. These contributions offer a better direction of what is still to be done to improve sarcasm detection systems in natural language processing, to make them more efficient and able to comprehend more human complex rhetoric.

6.3 Recommendation for Future Work

Although this thesis offers valuable understanding of the sarcasm detection in large language models, it can be said that there are a number of future research paths that can be used to improve the area:
Improvement of Model Precision: A significant weakness that we found during our analysis is the trade-off between Recall and Precision during sarcasm detection models. Future investigations can be aimed at more complicated architectures or refined older models to enhance Precision, which results into fewer false positives. It may be beneficial to consider hybrid models or Mixture-of-Experts mechanisms in order to tackle this issue and allow them to better specialise without impairing the performance of recall.

Optimization of the Rhetorical Device Detection: The models used in this thesis, like Llama 3.1-8B, are not as effective at identifying the less obvious types of sarcasm like Feigned Ignorance and complicated types of Hyperbole. For future studies, refinement of the Chain-of-Thought (CoT) reasoning mechanism could come in handy, possibly by further uniting the intent modeling and pragmatic understanding domains, to better represent these subtle rhetorical moves.

Cross-Domain Generalization: The existing models have good results in particular datasets, including Reddit-like or conversation corpora. Nevertheless, the process of cross-domain generalization should be tested and refined so that the process of sarcasm detection is sound across various environments (e.g., news articles, formal writing, or cultural differences in sarcasm). The future work might also imply the development of more diversified multilingual and cross-context data to assess the model and its robustness.

Explainability and Transparency in Models: In some models, such as Gemma12B, the explanation has much value in terms of detecting sarcasm, yet interpretability is a problem. Increasing the explainability of models by making reasoning process more transparent may enable researchers and developers to gain better insight into the process of why a model will detect sarcasm in a specific manner. It may be helpful to investigate the mechanisms of attention and map saliency to visualize the process of decisions made by the models and describe it in real-time.

Practical Use and Experimentation: Future work really excites me as the models of sarcasm detection can be implemented in the context of social media applications like facebook instagram etc, chatbots used by customers, or automated content moderators. The real-time, high-volume testing models would help to achieve newer insights into the performance and result in/spect improvements in the scalability and operational efficiency. Secondly, a combination of sarcasm detectors and emotion recognition models may provide a more detailed insight into the dynamics of conversations.

Through these areas, subsequent studies can make sarcasm detectors much further, more precise, and applicable in different areas.

Bibliography

[1] C. Burfoot and T. Baldwin, "Automatic satire detection: Are you having a laugh?" In *Proceedings of the ACL-IJCNLP 2009 conference short papers*, 2009, pp. 161–164.

[2] R. González-Ibánez, S. Muresan, and N. Wacholder, "Identifying sarcasm in twitter: A closer look," in *Proceedings of the 49th annual meeting of the association for computational linguistics: human language technologies*, 2011, pp. 581–586.

[3] A. Joshi, V. Sharma, and P. Bhattacharyya, "Harnessing context incongruity for sarcasm detection," in *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 2: Short Papers)*, 2015, pp. 757–762.

[4] D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words," in *proceedings of the 2015 conference on empirical methods in natural language processing*, 2015, pp. 1003–1012.

[5] S. Muresan, R. Gonzalez-Ibanez, D. Ghosh, and N. Wacholder, "Identification of nonliteral language in social media: A case study on sarcasm," *Journal of the Association for Information Science and Technology*, vol. 67, no. 11, pp. 2725–2737, 2016.

[6] D. Ghosh, A. R. Fabbri, and S. Muresan, "The role of conversation context for sarcasm detection in online interactions," *arXiv preprint arXiv:1707.06226*, 2017.

[7] D. Ghosh and S. Muresan, "" with 1 follower i must be awesome: P." exploring the role of irony markers in irony recognition," in *Proceedings of the International AAAI Conference on Web and Social Media*, vol. 12, 2018.

[8] A. R. F. D. Ghosh and S. Muresan, "Sarcasm analysis using conversation context," *Computational Linguistics*, vol. 44, no. 4, pp. 755–792, Dec. 2018. DOI: 10.1162/coli_a_00336.

[9] D. Ghosh, "An empirical study of verbal irony," Ph.D. dissertation, Rutgers University-School of Graduate Studies, 2018.

[10] D. Ghosh, E. Musi, K. Upasani, and S. Muresan, "Interpreting verbal irony: Linguistic strategies and the connection to the type of semantic incongruity," *Society for Computation in Linguistics*, vol. 3, no. 1, pp. 76–87, 2020. DOI: 10.7275/91ey-3n44.

[11] T. Chakrabarty, D. Ghosh, S. Muresan, and N. Peng, "*R*3: Reverse, retrieve, and rank for sarcasm generation with commonsense knowledge," *arXiv preprint arXiv:2004.13248*, 2020.

[12] A. Baruah, K. Das, F. Barbhuiya, and K. Dey, "Context-aware sarcasm detection using bert," in *Proceedings of the Second Workshop on Figurative Language Processing*, 2020, pp. 83–87.

[13] D. Ghosh, A. Vajpayee, and S. Muresan, "A report on the 2020 sarcasm detection shared task," *arXiv preprint arXiv:2005.05814*, 2020.

[14] D. Ghosh, R. Shrivastava, and S. Muresan, "Laughing at you or with you: The role of sarcasm in shaping the disagreement space," *arXiv preprint arXiv:2101.10952*, 2021.

[15] M. Khodak, N. Saunshi, and K. Vodrahalli, *A large self-annotated corpus for sarcasm*, 2018. arXiv: 1704.05579 [cs.CL]. [Online]. Available: https://arxiv.org/abs/1704.05579.

[16] Llama Team, AI Meta, "The llama 3 herd of models," *arXiv preprint arXiv:2407.21783*, 2024. DOI: 10.48550/arXiv.2407.21783. [Online]. Available: https://arxiv.org/abs/2407.21783.

[17] M. A. team, *Announcing mistral 7b*, https://mistral.ai/news/announcing-mistral-7b, Mistral AI — News, 2023.

[18] Gemma Team, Google DeepMind, "Gemma 3 technical report," *arXiv preprint arXiv:2503.19786*, 2025. DOI: 10.48550/arXiv.2503.19786. [Online]. Available: https://arxiv.org/abs/2503.19786.

[19] Q. Team, *Qwen3 technical report*, 2025. arXiv: 2505.09388 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2505.09388.

[20] DeepSeek-AI, *Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning*, 2025. arXiv: 2501.12948 [cs.CL]. [Online]. Available: https://arxiv.org/abs/2501.12948.