**Sarcasm Detection and Classification using Generative AI and Natural Language Processing.**

# Abstract

The significance of this sarcasm classification research lies in its pioneering approach to detect sarcasm and classify deadpan sarcasm in news headlines. By specifically focusing on deadpan sarcasm, the study contributes to a understanding of sarcastic nuances, filling a gap in existing literature. This research employs advanced natural language processing techniques and large language models, notably leveraging OpenAI's GPT-3.5, to classify both overt and deadpan sarcasm in news headlines. The innovation lies in integrating language models for sarcasm annotation, deadpan sarcasm detection as well as classification, and dataset

curation focusing exclusively on deadpan instances. This advances the field by offering a sophisticated approach to sarcasm classification. The research yields compelling results, showcasing a noteworthy 83.66% accuracy in sarcasm detection, led by logistic regression. Beyond overt sarcasm, the study into deadpan sarcasm detection, achieving a noteworthy 70% accuracy, with logistic regression demonstrating the highest recall at 0.68. And for deadpan sarcasm classification exhibiting the recall score of 0.66. What sets this research apart is the use of large language models for annotating the news headline dataset, outshining existing methodologies and providing notable results in different forms of sarcasm in news headlines, highlighting the research's significance in advancing the field.

# Introduction

In our modern digital age, the nuances of language hold significant importance. Among these nuances lies sarcasm – a rhetorical tool that cleverly conveys a meaning opposite to its literal words. Sarcasm serves as a powerful mode of intricate expression, especially crucial in the context of news headlines where conciseness is paramount.

Background:

This study delves into detecting nuances of sarcasm in news headlines, spanning overt and deadpan forms. News headlines, condensed windows into current events and opinions, offer insight into societal discourse. Detecting sarcasm here goes beyond surface understanding, revealing underlying sentiments. The research progresses beyond mere categorization, exploring intricate linguistic patterns and associated emotions.

Significance:

The research’s importance lies in enhancing our grasp of communication within news delivery. Amidst rampant misinformation, accurate sarcasm detection fosters discerning readership. Differentiating between overt and deadpan sarcasm presents a formidable challenge, requiring advanced classification techniques.

Scope:

This study comprehensively investigates sarcasm detection in news headlines, using advanced natural language processing methods. Employing meticulously curated data from satirical and factual sources, the study undergoes essential preprocessing: tokenization, stemming, contraction replacement, ensuring text consistency. Feature engineering introduces diverse linguistic and sentiment-based attributes. Negative word counts, polarity scores via n-grams, ellipsis frequency, duplicated letters, and vowel repetition enhance the analysis, capturing intricate linguistic cues.

Limitations:

However, this research has defined limitations. Its primary focus is detecting overt and deadpan sarcasm in news headlines. While ensemble methods are employed, the study doesn’t encompass potential social implications of misinterpreted sarcasm or demographic

interpretation variations. These areas hold promise for future exploration within sarcasm detection and communication analysis.

In conclusion, this research advances sarcasm detection, encompassing overt and intricate deadpan forms in news headlines. Exploring Sentiment Score, and positive/negative word counts further enriches deadpan sarcasm detection. This research bridges a critical gap in understanding sarcasm detection’s application in news headlines, offering insights into language, sentiment, and context interplay.

# Related Work

## Literature review

Yogesh Kumar and Goel (2020) [9] worked on the concept of sarcasm and its impact on sentiment analysis within the context of social media platforms. The study aimed involved varied tones and moods exhibited by writers on such platforms, analyzing how these sentiments could potentially influence others. A key objective was the development of models and techniques for detecting sarcasm, with the aim of enhancing the overall performance of sentiment analysis models. The data pre-processing steps included lemmatization for word standardization and complexity reduction, alongside the removal of usernames, URLs, and hashtags to streamline the dataset. Notably, the paper detailed various features employed for sarcasm classification, including sentiment-related measures, punctuation-based features, lexical and syntactic attributes, and pattern-related elements such as N-grams. Furthermore, the authors used optimization techniques utilized for sarcasm detection, categorizing them into different algorithms that aimed to enhance efficiency by minimizing costs through comparison of solutions.

Jamia and Ashraf [13] (2018) presents a unique methodology for detecting self-deprecating sarcasm within online social media, with a specific focus on Twitter. The study combines rule- based and machine learning techniques to discern tweets containing self-deprecating sarcasm. The authors also extract and analyze various features from the tweets. Their work involves

the application of three distinct classifiers to classify tweets as either self-deprecating or not. The methodology's effectiveness is assessed using a Twitter dataset and is benchmarked against existing sarcasm detection methods. Preprocessing involves the removal of URLs, mentions, retweets, hashtags, and other extraneous elements from the tweets. Tokenization is used to segment the tweets into individual words or tokens, while part-of-speech tagging

assigns grammatical tags to each token. The feature engineering process incorporates a set of 11 features, encompassing self-around features and hyperbolic features.

Sundararajan and Palanisamy (2019-20) [10] present a perspective on sarcasm detection, focusing on classifying different types of sarcasm to uncover the underlying emotional intent. The paper aims to enhance existing approaches by delving into the emotional behavior of individuals, offering insights through the relationship between emotional states and sarcasm types. Preprocessing involves cleaning raw Twitter data by removing hashtags, URLs, and links, followed by algorithmic POS tagging, stemming, and lemmatization to structure the data. Feature engineering encompasses a diverse set, including sentiment-related metrics, n-grams, emoji sentiment, and punctuation indicators. The study leverages techniques such as Rough Set theory for feature selection, Fuzzy Logic to handle ambiguity, and ensemble methods to improve classification performance. The ensemble feature selection approach combines linguistic, sentiment, and contradictory features for high accuracy in sarcasm detection. Results show accuracy of 92.7% for overall sarcasm classification, with precision in identifying distinct sarcasm types: Polite (95.98%), Rude (96.2%), Raging (99.79%), and Deadpan (86.61%).

The paper by Mandala Vishal Rao and Sindhu C. [16] aims to address the challenge of identifying sarcasm in product reviews on Amazon. The authors highlight the importance of understanding the sentiment behind user feedback, which can range from straightforward to sarcastic expressions. They emphasize the need for sentiment analysis in determining the true intent of the reviewer. The paper discusses the initial tasks involved in the process, including dataset selection from Amazon, data preprocessing such as tokenization, polarity identification, stemming, and lemmatization, as well as feature extraction techniques like term frequency, inverse document frequency, and n-grams. Classification algorithms such as Support Vector Machine (SVM), K Nearest Neighbors, and Random Forest are implemented for sarcasm detection, with evaluation based on accuracy metrics. Their approach involves data retrieval, preprocessing, feature extraction, classification, and evaluation. The results indicate an accuracy of 67.58% for SVM, 62.34% for Random Forest, and 61.08% for K-Nearest Neighbors in detecting sarcasm in product reviews.

The study in [11] focuses on sarcasm detection in Twitter, a vital task for sentiment analysis. It employs TextBlob for preprocessing and RapidMiner for polarity and subjectivity analysis. Weka is used to calculate tweet accuracy with Naïve Bayes and SVM classifiers. The findings offer tweet polarity, subjectivity confidence, and accuracy insights, enabling precise user opinion analysis.

Amer and Siddiqui (2022) [6] present an algorithm for enhancing sarcasm detection within social networks, with a specific focus on Twitter. Their research captures contextual and lexical-based features that influence semantic understanding. The proposed algorithm introduces three distinct feature sets: sarcastic-based, lexical-based, and context-based features. Utilizing a range of supervised machine learning classifiers including k-nearest neighbor (KNN), naïve Bayes (NB), support vector machine (SVM), and Random Forest (RF), the study aims to achieve heightened precision, accuracy, recall, and F1 score for sarcasm detection. Data preprocessing involves various steps such as tokenization, stop word removal, noise removal, stemming, punctuation removal, URL elimination, word elongation and truncation handling, contraction replacement, and Part-of-Speech tagging. KNN demonstrates a particularly high F1-score of 91.65%, along with precision and recall values of 90.76 and 90.89 respectively.

The research paper by Daniel and Marina [1] focuses on the challenging task of detecting sarcasm in textual data using machine and deep learning methods. Sarcasm's linguistic complexity, context dependence, and potential to express negative sentiment using positive words make it difficult for computational analysis. The study employs a dataset of 1.3 million social media comments containing both sarcastic and non-sarcastic instances. Various machine learning models, such as logistic regression, ridge regression, support vector machines, and deep learning models like Bidirectional Long Short-Term Memory and BERT-based models, are applied and compared for sarcasm detection. The paper highlights the superiority of deep learning models, particularly the BERT-based approach, over traditional machine learning methods for this task.

The study in [8] by Ray, A delves into the unexplored relationship between sarcasm and emotions. It introduces a novel task of detecting emotions within sarcastic expressions using a multimodal dataset, MUStARD++. The dataset includes emotion, valence, arousal, and sarcasm-type annotations. Emotion recognition within sarcasm is benchmarked with fusion models, outperforming existing sarcasm detection techniques. The paper contributes an

enriched dataset, corrections to emotion labels, and insights into emotion detection in sarcasm, aiding conversational systems and human interaction analysis.

The research paper by Bagate and Suguna (2022) [5] aims to implement sarcasm detection in text, particularly tweets, without relying on the presence of the #sarcasm keyword. Their approach employs machine learning and deep learning techniques to enhance text understanding and classification accuracy. The paper introduces a multi-faceted methodology involving data preprocessing, feature engineering, and ensemble methods. A variety of features are employed, encompassing word similarity scores, sentiment analysis, deep learning models like LSTM and BERT, attention mechanisms, word embeddings, and contextual attributes. These features are combined with machine learning algorithms such as SVM, logistic regression, random forest, and neural networks. The presence or absence of hashtags like #sarcasm is also explored as a feature. Combining multiple classification models, including Emotion, Personality, Sentiment, and Text, to create an ensemble model. This model merges processed vectors from different embedding techniques like Glove, enhancing prediction accuracy and overall performance of the classifiers.

# 1. Sarcasm Detection

## Dataset:

The selection of the "**News Headlines Dataset for Sarcasm Detection**" from **Kaggle** is driven by clear labelling of sarcastic and non-sarcastic headlines, sources including content from **TheOnion** and regular news from **HuffPost**, and its use as a benchmark in the Kaggle community. This dataset's incorporation of various forms of sarcasm and its presence on Kaggle's collaborative platform make it a relevant and valuable resource for developing and advancing effective sarcasm detection models. **26,000+** instances classified into sarcastic and non-sarcastic texts. Each record consists of three attributes:

The **‘is\_sarcastic’** column shows 1 if the record is sarcastic otherwise 0, the **‘headline’** column consists the headline of the news article and the **‘article\_link’** column links to the original news article. Useful in collecting supplementary data. Rishabh Misra and Prahal Arora [2] that used the Headline dataset [7] displayed a word cloud in their study that analysis the linguistic characteristics of the text gathered from the two sources These word clouds highlight

the prevalent words in each category. They observe that there isn't an immediate clear differentiation of words within each category. This lack of distinction could stem from the fact

that sarcasm's definition relies on context and might not exclusively rely on particular words. They also displayed the advantages of the news headline dataset over the existing sarcasm datasets.

## Preprocessing Steps:

Pre-processing is necessary to transform raw text into a clean, structured format that can be effectively utilized by machine learning models. While in the work of Abdullah Yahya and Tamanna [6] the instances of punctuations were removed but in case of sarcasm detection, the presence of ellipses, exclamations, capitalized letters and hashtags might help in detecting the instances of sarcasm. This may be subjective for various datasets, hence it necessary to analyze the data and choose the corresponding pre-processing steps accordingly. Therefore, in our case we will work on using pre-processing steps such as Tokenization which involves breaking down the headlines into individual words or tokens used in multiple studies. This step ensures that each word is treated as a separate unit during analysis, enabling further processing. Stemming is the process of reducing words to their root form. This helps in reducing variations of the same word and capturing the core meaning of words. And also replacing contractions like "don't" or "can't" are expanded to their full forms ("do not" or "cannot"). This step ensures that the model treats contractions and their expanded forms consistently. The preprocessing steps ensures that the textual data is appropriately formatted and normalized for analysis. For the sarcasm detection and sarcasm classification of deadpan sarcasm. The above mentioned Pre-processing steps were used throughout.

## Feature Engineering:

Feature engineering is a critical step in sarcasm detection because it enables the translation of linguistic and contextual nuances of sarcasm into a format that machine learning models can understand and utilize effectively.

In the context of sarcasm detection, feature engineering serves several crucial purposes.

Expanding upon the groundwork laid by Rajeswari and Shanthi Bala (2018) [14] in sarcasm detection, this study strategically enhances the efficacy of the widely adopted TF-IDF (Term Frequency-Inverse Document Frequency) vectorization method. While retaining the purpose of **TF-IDF vectorizer** which is employed to transform the textual content of the news headlines in the data into a numerical representation. This transformation generates a matrix

of TF-IDF features, where each row corresponds to a headline and each column represents a unique word in the entire dataset. Moreover, the study extends this approach by counting **positive and negative words** in text. These counts help assess the emotional tone and intensity of sentiment expressed. Feature engineering for sentiment-related attributes is highlighted in the paper by Kumar and Goel (2020) [9]. According to the research of Abdullah Amer and Tamanna Siddique, counting **Duplicated Letters and Vowel Repeats** in text indicates specific forms of expression, humor, or emphasis. Additionally, these patterns become a probable indication of sarcastic texts, and duplication of vowels to indicate the same. We also added features for **counting nouns, verbs, adjectives, and adverbs** which can enhance sarcasm. Sarcasm often hinges on the contrast between literal and intended meanings, with nouns and verbs highlighting such disparities. Adjectives aid in identifying exaggerated or ironic descriptions, while adverbs convey the speaker's tone and intensity, all contributing to a more comprehensive sarcasm analysis.

In our work, we used the SentimentIntensityAnalyzer (SID) to expand into the emotional undercurrents within our headline dataset. By breaking down each headline into its constituent words and extracting n-grams (word sequences of varying lengths) also present in the work of Yogesh Kumar and Nikita Goel, we effectively captured the essence of **sentiment expressed** across diverse linguistic structures. We then employed the SID to assign polarity scores to these n-grams, quantifying their emotional tone. The compound score provided by the analyzer encapsulated both positive and negative sentiment aspects. Aggregating these polarity scores allowed us to compute an overall sentiment polarity score for each headline.

## Ensemble Methods:

Ensemble methods are used in our research expecting to enhance the accuracy and reliability of our sarcasm detection model. By combining the predictions of multiple classifiers, compensate for individual weaknesses, and expect improved overall performance.

We employed two ensemble techniques:

1. Random Forest and Gradient Boosting Classifier: We combined Random Forest, which merges multiple decision trees expecting better performance, and Gradient Boosting, which corrects errors sequentially. This synergy maximizes the benefits of each approach and enhances our model's predictive power.
2. Adaboost Model: We also utilized Adaboost, which iteratively emphasizes misclassified instances, enhancing accuracy by focusing on challenging cases. This technique reinforces our sarcasm detection system's ability to accurately classify complex instances.

These ensemble methods empower our sarcasm detection model to achieve higher accuracy and reliability, ultimately trying to attain improved performance in identifying sarcasm in text. Whilst it was found that performing the classification separately gave better accuracy and recall score compared to done by ensemble methods.

## Classifiers and Results:

The accuracy of various classifiers in text classification was explored through a series of experiments. An initial setup involving tokenization, stemming, elongated word handling, and contraction expansion was employed. The chosen features encompassed positive and negative word counts, polarity scores, and detection of ellipses. Logistic regression achieved an accuracy of 83.36%, while random forest and Multinomial Naive Bayes attained accuracies of 81.64% and 80% respectively. In another reading, the same preprocessing steps were used, but additional features were introduced, including counts of nouns, verbs, adjectives, and adverbs. When using a random forest classifier, this feature augmentation led to an accuracy of 81.86%. On the other hand, logistic regression showed improved performance with an accuracy of 83.56%. In contrast to the approach taken by Abdullah Amer and Siddiqui in their research [5], where elongated and truncated words were simply shortened to their original forms, in our work counting of duplicated letters and vowel repeats as features, and eliminating of handling of elongated words as a pre-processing step. We observed, there wasn't a substantial increase in the results. With the accuracy of 83.47% from logistic regression to 83.66%, 83.66%, 83.66% by logistic regression, from 81.80% to 82.05%, 81.92%, 82.03%, 82.29%,82.07%. by Random Forest and Multinomial Naïve Bayes with 80.94%, 80.96%, 80.42%, 81.24% accuracy. When performed with same feature set with ensemble technique by using Random Forest and Gradient Boosting classifier exhibited the score of 80.29% and recall of 0.75. And with the Adaboost model attaining the accuracy of 73.61% and 0.82 recall Achieving a recall of about 81% by random forest and logistic regression whilst, Multinomial Naïve Bayes emits a lower recall of 65.73%.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Metric | Logistic Regression | Multinomial Naïve Bayes | Random Forest | Random Forest and Gradient  Boosting | Adaboost model |
| Accuracy | 83.66% | 81.24% | 82.29% | 80.29% | 73.61% |

**Pre-Processing for Deadpan Sarcasm:**

### Sarcasm Classification by OpenAI

Drawing inspiration from the groundbreaking study conducted by Gilardi, Alizadeh, and Kubli (July, 2023) [3], which explored the potential of large language models (LLMs) ChatGPT in specific, for text annotation tasks, our research adopted an innovative approach to tackle the challenging task of sarcasm classification. Also, a study by Mohammad Belal, James She and Simon Wong (June,2023) [4] hones in on the application of ChatGPT or sentiment analysis tasks showcasing ChatGPT's superior performance over conventional lexicon-based algorithms, across diverse datasets signal its potential to enhance the accuracy and efficiency of data labeling. The study employed an innovative approach to classify instances of sarcasm within a dataset. Leveraging the capabilities of the OpenAI API, specifically the GPT-3.5 "text-davinci-003" model, known for its advanced text analysis and generation, the research aimed to automatically categorize text samples into distinct sarcasm types, including "Deadpan sarcasm," "Self-deprecating sarcasm," and others. The methodology integrated the OpenAI API by incorporating necessary API key authentication. A custom-designed function facilitated interaction with the API, allowing the analysis of input text and generation of predictions regarding the specific type of sarcasm present.

In practice, the research dataset consisted of annotated text examples, each indicating the presence of sarcasm. Systematically evaluating the dataset, instances identified as sarcastic were isolated. Subsequently, the GPT-3.5 model was utilized to predict the precise category of sarcasm exhibited. Notably, the model's effectiveness lay in its capacity to generate contextually relevant responses when prompted with structured instructions.

Through analysis, the research successfully categorized sarcasm instances into various types. Among these types, "Deadpan sarcasm" emerged as the most frequent, occurring 1440 times. The study further conducted a classification task specifically focusing on "Deadpan sarcasm."

### Data Resampling:

To address the imbalance, a resampling technique involving undersampling was employed. This step aimed to create a balanced representation of the classes and enhance the model's classification performance. After undersampling the training data of majority class of the dataset had 2256 instances and testing data had 622 instances. Informed by the success of ChatGPT in outperforming crowd-workers for text annotation tasks [8], this research demonstrated the power of sophisticated techniques in automating intricate tasks like sarcasm classification, thereby contributing to a more nuanced understanding of the diverse ways sarcasm is expressed within textual data.

# Deadpan Sarcasm Detection

## Dataset Selection:

For our deadpan sarcasm detection task, we crafted our dataset by exclusively focusing on headlines with deadpan sarcasm. In the labeling process, instances of deadpan sarcasm were distinctly marked as '1,' while both non-sarcastic headlines and headlines with other types of sarcasm were uniformly labeled as '0.' By assigning '1' to deadpan sarcasm instances and '0' to all other instances, we aim to develop a robust model capable of discerning deadpan sarcasm amidst a variety of headline types, including other forms of sarcasm and non-sarcastic statements.

## Feature Engineering

The feature engineering steps for both deadpan sarcasm detection and classification are identical, leveraging a combination of lexicon-based sentiment analysis and sentiment score calculation. Positive and negative word counts capture sentiment polarity, while sentiment

analysis computes an overall sentiment score. The VADER sentiment analysis tool, specifically, provides a compound sentiment score, aiding in the distinction of emotionally neutral headlines.

## Results:

The results of deadpan sarcasm detection using three different machine learning models exhibit consistent performance across multiple runs. For Logistic Regression, recall values range from

0.68 to 0.65, with corresponding accuracies consistently at 70%. Support Vector Machine (SVM) yields recall scores of 0.65 to 0.64, accompanied by accuracies fluctuating between 69% and 70%. Random Forest, on the other hand, demonstrates recall values spanning from

0.62 to 0.57, with accuracy varying between 66% and 70%. These metrics collectively suggest stable model performance in identifying deadpan sarcasm, providing a evaluation of the models' ability to capture relevant instances across different runs.

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Logistic Regression | Support  Vector Machine | Random Forest |
| Accuracy | 70% | 70% | 70% |
| Recall | 0.68 | 0.65 | 0.62 |

# Deadpan Sarcasm Classification

## Dataset Selection:

For our various sarcasm classification task, we exclusively focused on sarcastic headlines. We designed our dataset by assigning labels where headlines embodying any particularly focused type of sarcasm (deadpan) were marked as '1,' and other types of sarcasm were marked as '0.' We deliberately excluded non-sarcastic headlines to ensure that our models were trained and

tested exclusively on instances of sarcasm, facilitating a more focused analysis of deadpan sarcasm detection.

## Feature Engineering:

The features for identifying deadpan sarcasm encompass both lexicon-based sentiment analysis and sentiment score calculation. The count of positive and negative words quantifies the sentiment polarity within the text, capturing the subtle emotional contrasts that define deadpan sarcasm. Additionally, sentiment analysis computes a sentiment score, offering a nuanced evaluation of the text's emotional tone. With this, one feature utilized the VADER (Valence Aware Dictionary and Sentiment Reasoner) sentiment analysis tool to assess the emotional tone of the headlines. The compound sentiment score, provided by VADER, indicated the overall sentiment. Headlines with compound scores below a specified threshold (0.2) were identified as potentially lacking emotional content, enabling the distinction of emotionally neutral headlines Collectively, these features delve into the intricate interplay between sentiment and intended meaning, effectively distinguishing deadpan sarcasm. Collectively, this approach delves into the interplay between sentiment and intended meaning, discerning instances of deadpan sarcasm.

## Results:

After conducting multiple runs of the classification task using various classifiers with the specified features of sentiment score calculation, counting Positive and Negative words in the text and calculating compound score it was observed that the classifiers produced the following recall scores: Logistic Regression achieved the highest recall score of 0.66, 0.66,0.66, followed by Random Forest with 0.56, 0.56, 0.56, and Support Vector Machine (SVM) with 0.60, 0.60,

0.60.

|  |  |  |  |
| --- | --- | --- | --- |
| Metric | Logistic Regression | Support  Vector Machine | Random Forest |
| Recall | 0.66 | 0.60 | 0.56 |

# Conclusion:

In this study, a comprehensive approach was employed to detect and classify sarcasm, particularly focusing on the form of deadpan sarcasm in news headlines. Leveraging advanced natural language processing techniques, the research utilized data preprocessing, innovative feature engineering, and Machine Learning Techniques to enhance the performance of the model.

Innovatively, the research harnessed the capabilities of ChatGPT for sarcasm classification, notably focusing on deadpan instances. By integrating the OpenAI API, particularly the GPT-

3.5 "text-davinci-003" model, the study automated the categorization of headline samples into distinct sarcasm types. This unique application of ChatGPT demonstrates its effectiveness in automating intricate tasks, contributing to a detailed understanding of various sarcasm expressions within textual data.

The research yielded compelling results, demonstrating a notable accuracy of 83.66% in overt sarcasm detection. Logistic regression emerged as the leading classifier in this task. Additionally, the study delved into the intricate realm of deadpan sarcasm detection, consistently achieving a commendable accuracy of 70%, with logistic regression exhibiting the highest recall at a balanced 0.68. When specifically classifying deadpan sarcasm instances, logistic regression led with a recall of 0.66.

The future enhancements of this research involve extending the scope to explore potential social implications of misinterpreted sarcasm and demographic interpretation variations. Furthermore, there is room for investigating the application of sarcasm detection in broader communication analysis beyond news headlines. Exploring these avenues would provide valuable insights into the multifaceted nature of sarcasm and its impact on diverse audiences. Additionally, the study could benefit from further exploration of sentiment and emotional nuances in sarcasm, utilizing advanced techniques such as multimodal analysis. This forward- looking approach would contribute to the ongoing evolution of sarcasm detection

methodologies and their broader applications in understanding language, sentiment, and context.

# References:

1. Šandor, D., & Bagić Babac, M. (2023). Sarcasm detection in online comments using machine learning. Information Discovery and Delivery.
2. Misra, R., & Arora, P. (2023). Sarcasm detection using news headlines dataset. AI Open, 4, 13-18.
3. Gilardi, F., Alizadeh, M., & Kubli, M. (2023). Chatgpt outperforms crowd-workers for text- annotation tasks. *arXiv preprint arXiv:2303.15056*.
4. Belal, M., She, J., & Wong, S. (2023). Leveraging ChatGPT As Text Annotation Tool For Sentiment Analysis. arXiv preprint arXiv:2306.17177.
5. Bagate, R. A., & Suguna, R. (2022). Sarcasm Detection with and without# Sarcasm: Data Science Approach. *International Journal of Information Science and Management (IJISM)*, *20*(4), 1-15.
6. Abdullah Amer, A. Y., & Siddiqu, T. (2022). A novel algorithm for sarcasm detection using supervised machine learning approach. *AIMS Electronics & Electrical Engineering*, *6*(4)
7. Misra, R. (2022). News headlines dataset for sarcasm detection. *arXiv preprint arXiv:2212.06035*.
8. Ray, A., Mishra, S., Nunna, A., & Bhattacharyya, P. (2022). A multimodal corpus for emotion recognition in sarcasm. arXiv preprint arXiv:2206.02119.
9. Kumar, Y., & Goel, N. (2020). AI-Based learning techniques for sarcasm detection of social media tweets: State-of-the-art survey. *SN Computer Science*, *1*(6), 318.
10. Sundararajan, K., & Palanisamy, A. (2020). Multi-rule based ensemble feature selection model for sarcasm type detection in twitter. *Computational intelligence and neuroscience*, *2020*.
11. Kamal, A., & Abulaish, M. (2020). Self-deprecating humor detection: A machine

learning approach. In Computational Linguistics: 16th International Conference of the Pacific Association for Computational Linguistics, PACLING 2019, Hanoi, Vietnam, October 11–13, 2019, Revised Selected Papers 16 (pp. 483-494). Springer Singapore.

1. Suhaimin, M. S. M., Hijazi, M. H. A., Alfred, R., & Coenen, F. (2019). Modified

framework for sarcasm detection and classification in sentiment analysis. Indonesian Journal of Electrical Engineering and Computer Science, 13(3), 1175-1183.

1. Abulaish, M., & Kamal, A. (2018, December). Self-deprecating sarcasm detection: an amalgamation of rule-based and machine learning approach. In *2018 IEEE/WIC/ACM International Conference on Web Intelligence (WI)* (pp. 574-579). IEEE.
2. Rajeswari, K., & ShanthiBala, P. (2018). Sarcasm detection using machine learning techniques. Int J Recent Sci Res, 9, 26368-26372.
3. Saha, S., Yadav, J., & Ranjan, P. (2017). Proposed approach for sarcasm detection in twitter. Indian Journal of Science and Technology, 10(25), 1-8.
4. Rao, M. V., & Sindhu, C. (2021, March). Detection of sarcasm on amazon product reviews using machine learning algorithms under sentiment analysis. In 2021 sixth international conference on wireless communications, signal processing and networking (WiSPNET) (pp. 196-199). IEEE.