Journal of Computing Research and Innovation (JCRINN) Vol. 8 No.2 (2023) (pp1-10)
https://jcrinn.com : eISSN: 2600-8793

Unveiling Sarcastic Intent: Web-Based Detection of Sarcasm in News

Mohd Nazzim Lahaji[1], Tajul Rosli Razak[2], Mohammad Hafiz Ismail[3]
[1,2] School of Computing Sciences, College of Computing, Informatics and Mathematics, Universiti Teknologi Mara, Selangor, Malaysia
[3]School of Computing Sciences, College of Computing, Informatics and Mathematics, Universiti Teknologi Mara, Perlis, Malaysia

*Corresponding author: * tajulrosli@uitm.edu.my*

Received Date: 6 June 2023
Accepted Date: 5 July 2023
Revised Date: 10 August 2023
Published Date: 1 September 2023

HIGHLIGHTS

- The proposed algorithm effectively distinguishes between sarcastic and non-sarcastic headlines by analysing the semantic features of words and the underlying attitude conveyed by the headline's structure.
- Data pre-processing played a critical role in preparing the dataset for analysis and modelling, ensuring the accuracy and reliability of the system.
- The significance of this study lies in its novel contribution to the field of sarcasm detection in news headlines.

ABSTRACT

Detecting sarcasm in news headlines poses a significant challenge due to the intricate nature of language and the subtle nuances of sarcastic expressions. This study uses machine learning techniques to introduce a novel web-based sarcasm detection system tailored explicitly for news headlines. This study's key novelty and contribution lie in addressing the domain-specific problem of sarcasm detection in news headlines, which has received limited attention in previous research. The proposed algorithm effectively distinguishes between sarcastic and non-sarcastic headlines by analysing the semantic features of words and the underlying attitude conveyed by the headline's structure. Data pre-processing played a critical role in preparing the dataset for analysis and modelling, ensuring the accuracy and reliability of the system. A comparative study was conducted to validate the system's performance, benchmarking it against existing approaches. The results demonstrate the superiority of the developed model in sarcasm detection for news headlines. The system's unique output classifies sarcastic words into low, moderate, or high probabilities of being sarcastic, providing valuable insights into the intensity of sarcasm. Notably, the system is user-friendly and versatile, capable of processing diverse inputs effectively. The significance of this study lies in its novel contribution to the field of sarcasm detection in news headlines. By addressing the specific challenges of this domain, the developed system enhances the overall understanding and interpretation of news content. It is a valuable tool for individuals and news organisations, enabling swift and reliable identification of sarcasm in headlines, thereby enriching news comprehension and analysis.

Keywords: *Sarcasm Detection, News Headline, News Headline, Web Application*

INTRODUCTION

Sarcasm is a sophisticated form of expressing emotions in which the speaker says the opposite of what they intend. Detecting sarcasm and feelings on social networking sites has been the subject of extensive research. With the expansion of e-services such as e-commerce, e-tourism, and e-business, companies are eager to incorporate sentiment and sarcasm analysis into their marketing strategies to gauge the public's brand perception. Consequently, an effective modelling system for emotion and sarcasm can address the abovementioned issue. Newspapers frequently use sarcasm in their headlines to attract the attention of readers. Most of the time, however, readers cannot recognise the irony in the headlines, forming a false impression of the news and passing it on to their peers and co-workers (Shrikhande et al., 2020).

Every individual on earth possesses a unique social constitution. They frequently interact in a variety of ways, both positively and negatively. However, sarcasm is the most significant factor that can influence interpersonal communication. Essentially, sarcasm is the use of cynicism and mockery to convey contempt. Detecting sarcasm will accurately classify this sentence as sardonic or non-sarcastic. The difficulty becomes evident when individuals comprehend a statement without knowing the speaker's facial expressions or tone (Thakur Sakshiand Singh, 2020). This paper will analyse the statement using train data extracted from news titles.

Sentiment analysis usually consists of three elements depending on the context:
(i) Opinions or emotions: An opinion is also referred to as polarity, whereas feelings can be qualitative, such as sad, joy, anger, surprise, disgust, or happiness or quantitative, such as rating a movie on a scale of one to ten.
(ii) Subject: It refers to the subject of the discussion where one opinion can discuss more than one aspect of the same issue. For instance, the phone's camera is excellent. However, the battery life is disappointing, and
(iii) Opinion hole refers to the author/person who expresses the opinion (Lamba & Madhusudhan, 2022).

Emotion detection serves multiple purposes, including identifying sarcasm in various contexts, such as client opinions, employee feedback, and online discussions. Recognising irony presents a challenge in natural language processing, particularly on platforms like Twitter, where it can be difficult for humans to distinguish. Irony, often akin to lying, involves expressing the opposite of one's intended meaning, making it problematic to detect. However, advancements in natural language processing and machine learning have led to the development of algorithms that analyse linguistic features and contextual information to identify sarcasm effectively (Ashwitha et al., 2021).

Media companies have increasingly used sarcasm in news headlines to capture readers' attention and convey more profound messages. However, readers often struggle to recognise sarcasm in written text, leading to misunderstandings. To address this, online applications for sarcasm detection in news headlines have been developed to accurately identify and comprehend sarcasm in written language. Detecting sarcasm in news headlines is challenging due to the absence of nonverbal cues like facial expressions and tone of voice. Nevertheless, computer algorithms utilising natural language processing and machine learning techniques have been designed to learn from past instances and leverage linguistic features and contextual information to detect sarcasm in a written text (Misra & Arora, 2023).

As a result, sentiment analysis is critical in both marketing and business. Knowing what customers want is the first step toward making a business popular. Business and marketing organisations can plan and make better decisions if they have enough information and expertise gained from collected datasets.

BACKGROUND

The sentiments are the feedback, feelings, thoughts and opinions given by a person or a company. Finding the nature of those sentiments is possible through opinion mining or sentiment analysis. These reviews are essential in any perspective, whether for users or the company (Zain Naqvi, n.d.). Much work has been attempted in sentiment analysis (SA)/opinion mining of natural language texts (NLT) and social media. One of the primary objectives of such tasks is to allocate polarity, either positive (+ve) or damaging (−ve), to a part of the text. Nevertheless, at a similar time, the problem of assigning the degree of positivity and negativity of a particular text occurs. The situation becomes more difficult in the case of text gathered from social sites, as these sites contain several emoticons and sarcastic words with hidden meanings and expressions (Gupta et al., 2020).

This study will examine previous research on sarcasm detection to determine the best practical methods for use in this study. The relevance of an investigation will be determined by its relevance to the present research aims, the technique used, and the precision of the data collected. This study will also analyse the most recent developments in the field and the application of the chosen methodologies to web applications. By studying and comparing the results of previous research, this study will be able to select the most effective techniques for sarcasm detection in news headlines and increase the accuracy of this study's web application. This will guarantee that the current study makes a significant addition to the field of sarcasm detection and gives essential insights into public attitudes and views.

Bayesian Classifier

Bayesian Classifier is a statistical method that uses Bayes' theorem to make predictions based on probability (Eddy, 2004). Bayes' theorem states that the likelihood of an event occurring given some observations is proportional to the possibility of the statements given the event multiplied by the prior probability of the event. The Bayesian Classifier uses this theorem to calculate the posterior probability of each class given the words and then chooses the course with the highest probability as the prediction.

Random Forest

Random Forest is a meta-algorithm which uses several Decision trees. It combines the prediction value of each decision tree. It uses the majority vote method, which returns the class with majority votes. Sometimes, a Decision tree grows deeply and faces the problem of overfitting and learning irregular patterns. Nevertheless, Random Forest solves this problem of overfitting (Vaishnavi et al., 2022).

Support Vector Machine

Support Vector Machines (SVMs) are a popular machine learning algorithm used in various applications, including sentiment analysis. The basic idea behind SVMs is to find a hyperplane in a high-dimensional space that separates the data into two classes with maximum margin. In sentiment analysis, the input data collects textual reviews or tweets. The output is the sentiment associated with each text, such as positive, negative, or neutral (Zhang et al., 2020).

PROPOSED METHOD: WEB-BASED DETECTION OF SARCASM IN NEWS HEADLINES

The proposed method for web-based sarcasm detection on news headlines is designed to identify sarcastic statements within news headlines automatically published online. The technique uses natural language processing (NLP) algorithms to analyse the linguistic patterns and contextual cues present in the headline text to determine whether the headline is meant to be sarcastic. It can provide valuable insights into the use of sarcasm in online news headlines. It can help journalists and media analysts better understand the impact of irony on readers' perceptions of news stories.

Data Collection

This section describes the data collection process for the New Headlines Sarcasm dataset (Misra & Grover, 2021). The dataset will be provided in the JavaScript Object Notation (JSON) format, a standard text-based format for representing structured data based on JavaScript object syntax (Working with JSON - Learn Web Development | MDN, 2022). The initial dataset consists of three attributes, but any unnecessary details will be removed during pre-processing. The data used in the system is sourced from two news websites, The Onion for sarcastic data and HuffPost for non-sarcastic data. The dataset is collected directly from these websites, and only the headlines are extracted for analysis. It should be noted that all data used in this study were collected in compliance with ethical standards and relevant regulations regarding data privacy and protection. Informed consent was not required for this study as the data was publicly available and contained no personally identifiable information.

Data Pre-processing

In sentiment analysis, data pre-processing is crucial since it enables the cleaning and organisation of data in preparation for examination. The pre-processing task involves several duties, such as removing unnecessary or irrelevant information, correcting errors, and formatting the data in a manner suitable for the intended analysis. Furthermore, pre-processing the data may help improve the sentiment analysis's accuracy by eliminating noise or bias from the data, leading to higher-quality results. It is essential to note that even minor inaccuracies in the data can significantly impact the study's conclusions. Therefore, data pre-processing plays a critical role in ensuring the validity and reliability of the research findings.

Development of the Model

The following process will identify the requirements in terms of hardware and software. The hardware must be able to support the software. Any additional software used in the implemented system must have a CPU of x86 64-bit, either Intel or AMD architecture, with a minimum of 4 GB of RAM and 5 GB of free disk space. Regarding software requirements, a modern operating system, Windows 7 above, Mac OS 10.11 or higher run by 64-bit or Linux: RHEL 6/7, 64-bit. The other condition will be the code editors such as BlueJ, PyCharm or Visual Studio Code. For documenting this project will use Microsoft Office.

Python's massive array of libraries, such as sci-kit-learn, TensorFlow, and Keras, which can be used to design and apply machine learning models quickly, makes it a popular option for machine learning and data analysis. This is one reason why Python is such a popular programming language. In this study, Python will be used to write and build the model's code. Additionally, libraries will be employed to assess the correctness of the algorithm and its f1 score and make the system user-friendly for the person who will ultimately be using it.

Implementation of the model

There are 3 phases needed in this implementation: develop the system, train the dataset based on a chosen algorithm to compare their accuracy as necessary and create the Graphical User Interface (GUI). For the first phase, this project will install all the necessary dependencies that will be needed to develop the system model using Python. Next up will be importing any essential modules or libraries that will be used to create and train the sentiment analysis model. The model that will be used is *pandas* and *sklearn*. Then, the dataset will be loaded for the subsequent phases and go through data pre-processing and transformation. Finally, this project will apply the best model to train the models. As shown in Figure 1, below are the steps in sentiment analysis.

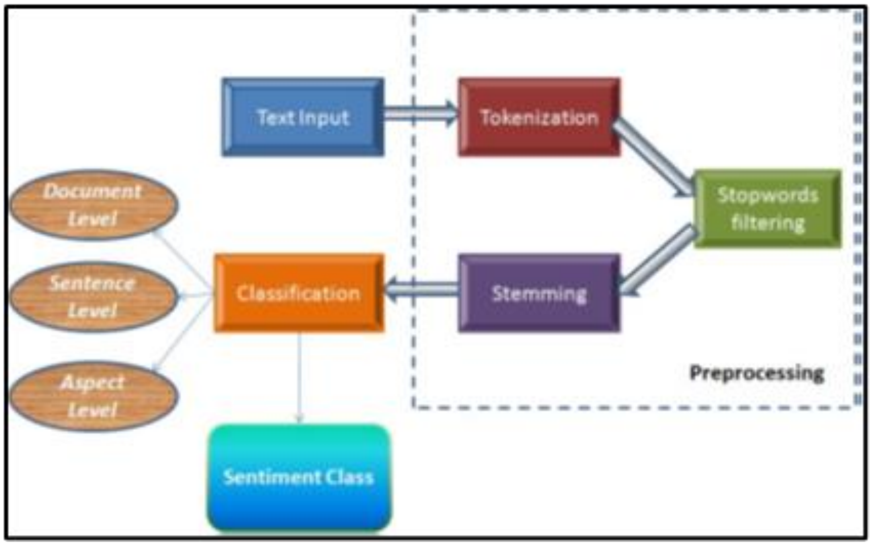


Figure 1. Steps of Sentiment Analysis (Sagnika et al., 2020).

Figure 1 illustrates the initial phase, which involves gathering the text input. Subsequently, the accumulated text undergoes pre-processing to ensure its suitability for further analysis. Once the pre-processing step is completed, the text is subjected to classification. Various metrics such as accuracy, precision, recall, and f1-score assess the model's performance. Suppose the model meets the study's requirements. In that case, it can be prepared for deployment in a production setting to classify incoming data in real time (Vaishanvi et al., 2022).

EXPERIMENT

The experiment conducted in this study aimed to develop and evaluate a web-based sarcasm detection system for news headlines. In this study, the model that has been selected are Multinominal Naïve Bayes, Support Vector Machine and Random Forest Model. This model will undergo data pre-processing, and they will be compared with each other regarding their accuracy and f1-score. This study will initially read the data using the *Pandas* library, as shown in Figure 2 below.

```
#read the dataset
import pandas as pd
data = pd.read_json(r'C:\Users\Nazzim\Desktop\FYP\code\Sarcasm_Headlines_Dataset_v2.json', lines=True)
```

Figure 2. Example of code on reading the dataset of JSON file

Then, it will undergo a process of data cleaning. Although cleaning the data can be time-consuming, ensuring the analysis results are accurate and insightful is essential. The data pre-processing stage may involve several tasks, such as removing noise, correcting errors, and consistently formatting the data. Data pre-processing is often necessary because real-world data is messy and unstructured and may contain errors, inconsistencies, and irrelevant information. By cleaning and preparing the data before performing sentiment analysis, the accuracy and reliability of the research can be improved, and the study can be conducted with high-quality data. This can help ensure that the study's findings are valid, reliable, and informative and can provide valuable insights into using sarcasm in news headlines.

```
is_sarcastic                                          headline
           1  Thirty something scientists unveil Doomsday cl...
           0  dem rep. totally nails why congress is falling...
           0  eat your veggies: 9 deliciously different recipes
           1  Inclement Weather prevents liar from getting t...
           1  mother comes pretty close to using word 'strea...
           0                               my white inheritance
           0          5 ways to file your taxes with less stress
           1  richard branson's global-warming donation near...
           1  shadow government getting too large to meet in...
           0                 lots of parents know this scenario

                                      article_link
https://www.theonion.com/thirtysomething-scien...
https://www.huffingtonpost.com/entry/donna-edw...
https://www.huffingtonpost.com/entry/eat-your-...
https://local.theonion.com/inclement-weather-p...
https://www.theonion.com/mother-comes-pretty-c...
https://www.huffingtonpost.com/entry/my-white-...
https://www.huffingtonpost.com/entry/5-ways-to...
https://www.theonion.com/richard-bransons-glob...
https://politics.theonion.com/shadow-governmen...
https://www.huffingtonpost.comhttp://pubx.co/6...
```

Figure 3. Example attributes and data in the dataset

Figure 3 shows three attributes in the dataset, but only '*is_sarcastic'* and 'headline' are needed in this model. Therefore, the next step is to remove the attributes of '*article_link'*. This step is crucial because deleting unneeded characteristics can assist in minimising noise in the data, enhancing the model's performance. A noise in the dataset is unrelated information that hurts the model and can take various forms, such as unrelated words or phrases, misspellings, and out-of-context text. Removing such noise will improve performance, allowing the model to focus on the most relevant and significant data.

The next step that will be done in this phase is applying the function of stop words, removing punctuations, memorable characters, and numerical tokens, and casing the characters. The example of the output after using this function is shown in Figure 4 below.

```
thirty something scientists unveil doomsday cl...
dem rep totally nails congress falling short g...
          eat veggies deliciously different recipes
       inclement weather prevents liar getting work
mother comes pretty close using word streaming...
                                  white inheritance
                       ways file taxes less stress
richard branson global warming donation nearly...
shadow government getting large meet marriott ...
                          lots parents know scenario
```

Figure 4. Example result after applying the related function.

Based on Figure 4 above, the function of stop words is being used because stop words are words that are commonly used in natural language and are generally not helpful in determining the content or meaning of a document. Besides that, removing punctuation marks from a document before performing sentiment analysis can simplify the text and reduce noise in the data. This can be especially helpful if the punctuation marks do not convey sentiment and are irrelevant to the analysis. Lastly, casing refers to the capitalisation of letters in a word. For example, "thirty" and "Thirty" are the same word but have different cases. Before doing sentiment analysis, it is required to convert the text to a consistent point to guarantee that words are not recognised as separate entities only because of their capitalisation. As a result, converting all the words to lowercase or uppercase form can make the analysis more accurate and consistent. Figure 5 below shows the dataset's example of vectorise word using a count vectoriser.

	abandon	abandoned	abandoning	...	zookeeper	zoologists	zuckerberg
0	0	0	0	...	0	0	0
1	0	0	0	...	0	0	0
2	0	0	0	...	0	0	0
3	0	0	0	...	0	0	0
4	0	0	0	...	0	0	0
...	...	...	...	...	...	...	...
28614	0	0	0	...	0	0	0
28615	0	0	0	...	0	0	0
28616	0	0	0	...	0	0	0
28617	0	0	0	...	0	0	0
28618	0	0	0	...	0	0	0

Figure 5. Vectorize dataset using Count Vectorizer

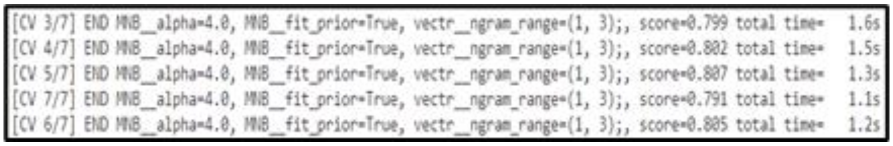
Based on Figure 5 above, the word has been vectorised using a count vectoriser. Count vectorisation is a technique for turning a group of text documents into a numerical representation known as a "document-term matrix". Vectorisation is required since most machine learning models can only deal with numerical data; thus, text data must be translated into a numerical form before being utilised as input to the model. Vectorisation enables the extraction of text data characteristics that may be used to train a machine learning model. Essentially, vectorisation is a crucial stage in preparing a machine learning model for sentiment analysis since it converts text input into a form the model can comprehend and utilise to create predictions.

The following step is to apply pipeline functionality. A pipeline is a set of actions or procedures to accomplish a specific objective. These processes often entail some mix of data pre-processing, model training, and model performance evaluation. Because pipelines enable one to automate and simplify the process of training and testing machine learning models, pipelines are advantageous when dealing with massive datasets or when one needs to train many models quickly. Figure 6 below shows the example code used in the implementations.

```
pipe1 = Pipeline([('vectr', CountVectorizer(analyzer='word', preprocessor=None, min_df=1,)),
                  ('MNB', MultinomialNB()),])
```

Figure 6. Example code of pipeline

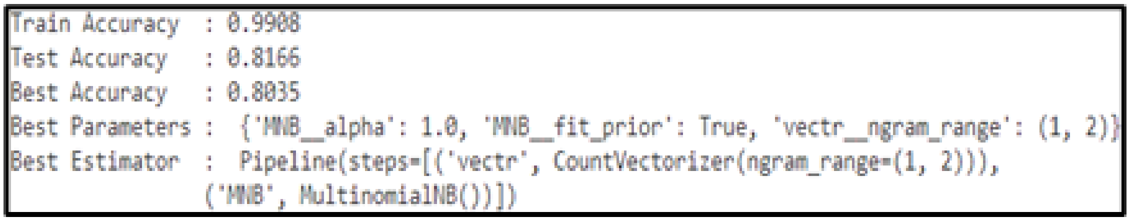
In the third step of training the model, GridSearchCV from the scikit-learn package is used. This strategy enables the search for the optimal model hyperparameters. Hyperparameters are practitioner-specified parameters that are not learnt from the training data. GridSearchCV specifies a range of potential values for these hyperparameters and automatically trains and assesses a model for each conceivable combination of values, delivering the optimal combination. Figure 7 below is an example of the GridSearchCV process.



```
[CV 3/7] END MNB__alpha=4.0, MNB__fit_prior=True, vectr__ngram_range=(1, 3);, score=0.799 total time=   1.6s
[CV 4/7] END MNB__alpha=4.0, MNB__fit_prior=True, vectr__ngram_range=(1, 3);, score=0.802 total time=   1.5s
[CV 5/7] END MNB__alpha=4.0, MNB__fit_prior=True, vectr__ngram_range=(1, 3);, score=0.807 total time=   1.3s
[CV 7/7] END MNB__alpha=4.0, MNB__fit_prior=True, vectr__ngram_range=(1, 3);, score=0.791 total time=   1.1s
[CV 6/7] END MNB__alpha=4.0, MNB__fit_prior=True, vectr__ngram_range=(1, 3);, score=0.805 total time=   1.2s
```

Figure 7. Example of 'GridSearchCV' process

Based on Figure 7 above, GridSearchCV will train and evaluate a model for every combination of hyperparameter values, returning the variety that produces the best results based on the scoring metric. Figure 8 below displays an example of the outcomes of the best scoring measure.



```
Train Accuracy  : 0.9908
Test Accuracy   : 0.8166
Best Accuracy   : 0.8035
Best Parameters :  {'MNB__alpha': 1.0, 'MNB__fit_prior': True, 'vectr__ngram_range': (1, 2)}
Best Estimator  :  Pipeline(steps=[('vectr', CountVectorizer(ngram_range=(1, 2))),
                  ('MNB', MultinomialNB())])
```

Figure 8. Examples of best-scoring measures

As shown in Figure 8 above, the best parameters and estimator will be utilised as references for constructing the machine learning model, followed by comparing the chosen methods to choose the one with the highest accuracy.

RESULT AND DISCUSSION

This section will discuss web-based sarcasm detection on news headlines and present – the study's findings, analysis, and interpretation.

Model Comparison

First, the comparison between selected algorithms was presented. As can be seen in Table 1 below, the selected models that will be compared are Multinomial Naive Bayes (MNB), Support Vector Machine (SVM) and Random Forest (RF) using the method in data pre-processing.

Table 1. Accuracy comparison.

Model	MNB	SVM	RF
Accuracy	0.8087	0.8055	0.7739
F1-Score	0.7972	0.7863	0.7423

The comparison in Table 1 above shows that the three algorithms, naïve Bayes, support vector machine and random forest, are based on the accuracy and f1-score. The highest in both will be selected as the model for use in the Sarcasm Detection on the News Headline. Since accuracy helps to assess the model's effectiveness, it is a critical measure in sentiment analysis. In conclusion, Multinominal Naive Bayes has been selected as the algorithm that will be trained to become the model employed in this study because of its high accuracy and f1-score. A higher F1 score indicates superior performance since it is determined by taking the harmonic mean of the accuracy and recall scores. A high F1 score is essential because the model must generate accurate predictions while catching as many positive examples as possible.

Graphical User Interface

Next, the graphical user interface (GUI) is essential to the web-based sarcasm detection system for news headlines was discussed. It provides a user-friendly interface for users to interact with the design and efficiently detect sarcasm in news headlines.

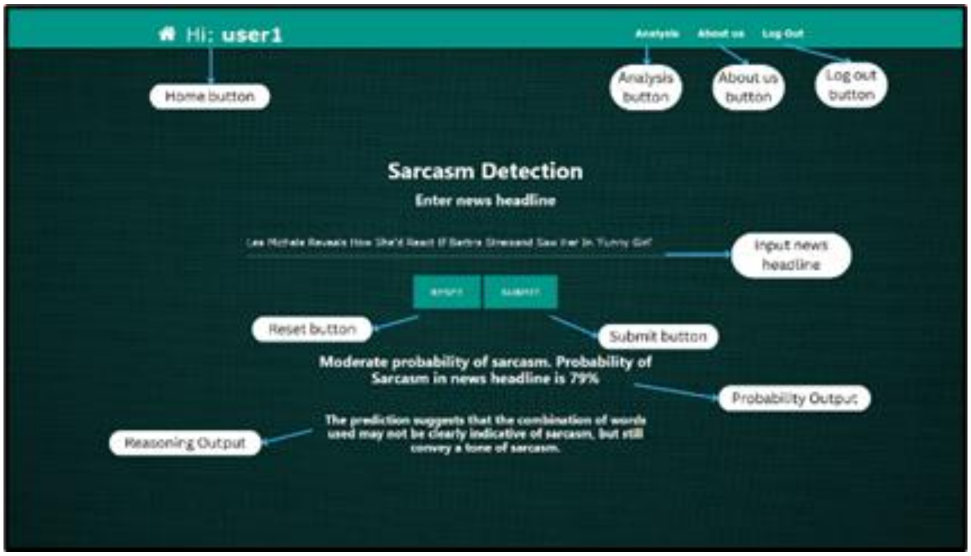


Figure 8. Example of output on GUI

The user submits the news headline by clicking the "submit" button; the system will generate an output that includes. Additionally, the system will indicate the level of sarcasm in the news title. To illustrate, let us consider the example input, "Lea Michele Reveals How She Would React If Barbra Streisand Saw Her In 'Funny Girl'," sourced from the HuffPost website (*Lea Michele Reveals How She'd React If Barbra Streisand Saw Her In "Funny Girl,"* n.d.).

Based on the probability of the prediction, it will show the reasoning as shown below based on the baseline researched by Richards; research indicates that human analysts tend to agree around 80-85% of the time. This is the baseline we (usually) try to meet or beat when training a sentiment scoring system (*Sentiment Accuracy: Explaining the Baseline and How to Test It - Lexalytics*, n.d.):

(i). *High Probability (>=80%):*
Many of the words used in the prediction are commonly used in sarcasm, which likely contributed to the high probability of irony being detected.

(ii). *Moderate Probability (>=50% and <80%):*
The prediction suggests that the combination of words used may not clearly indicate sarcasm but still convey a tone of sarcasm.

(iii). *Low probability (<50%):*

The prediction suggests that the statement contains no language commonly associated with sarcasm, indicating a low probability of irony.

CONCLUSION AND RECOMMENDATIONS

In conclusion, we have presented a strategy for addressing the domain-specific problem of sarcasm detection in news headlines using machine learning techniques. By analysing the semantic features of words and the underlying attitude conveyed by the headline structure, the approach demonstrated promise for differentiating between sardonic and non-sarcastic headlines. While developing a web application for sarcasm detection in news headlines has many advantages, it is essential to address the associated limitations, such as the difficulty of accurately detecting sarcasm, the absence of labelled data, the possibility of bias, and the possibility of misuse. Future research could concentrate on developing more robust and accurate algorithms, expanding the availability of labelled data, and examining the ethical and societal implications of sarcasm detection technology. Also, future research may include multi-lingual or cross-cultural sarcasm detection models, audio and video sarcasm detection, transfer learning and active learning strategies.

FUNDING

There was no specific grant for this research from any governmental, commercial, or not-for-profit funding agencies.

ACKNOWLEDGMENTS

The authors are grateful for the comments from reviewers in improving this manuscript. This research received no specific grant from any funding agency in the public, commercial, or not-for-profit sectors.

CONFLICT OF INTEREST DISCLOSURE

The authors declared that they have no conflicts of interest to disclose

REFERENCES

Ashwitha, A., Shruthi, G., Shruthi, H. R., Upadhyaya, M., Ray, A. P., & Manjunath, T. C. (2021). Sarcasm detection in natural language processing. *Materials Today: Proceedings*, *37*(Part 2), 3324–3331. https://doi.org/10.1016/J.MATPR.2020.09.124

Eddy, S. R. (2004). What is Bayesian statistics? *Nature Biotechnology*, *22*(9), 1177–1178. https://doi.org/10.1038/NBT0904-1177

Gupta, S., Singh, R., & Singla, V. (2020). Emoticon and text sarcasm detection in sentiment analysis. *Advances in Intelligent Systems and Computing*, *1045*, 1–10. https://doi.org/10.1007/978-981-15-0029-9_1/COVER

Lamba, M., & Madhusudhan, M. (2022). Sentiment Analysis. *Text Mining for Information Professionals*, 191–211. https://doi.org/10.1007/978-3-030-85085-2_7

Lea Michele Reveals How She'd React If Barbra Streisand Saw Her In "Funny Girl." (n.d.). Retrieved July 7, 2023, from https://www.huffpost.com/entry/lea-michele-streisand-funny-girl_n_63c26450e4b0d6724fce2640

Misra, R., & Arora, P. (2023). Sarcasm detection using news headlines dataset. *AI Open*, *4*, 13–18. https://doi.org/10.1016/J.AIOPEN.2023.01.001

Misra, R., & Grover, J. (2021). *Sculpting Data for ML: The first act of Machine Learning*.

Sagnika, S., Pattanaik, A., Mishra, B. S. P., & Meher, S. K. (2020). A review of multi-lingual sentiment analysis by machine learning methods. *Journal of Engineering Science and Technology Review*, *13*(2), 154. https://doi.org/10.25103/jestr.132.19

Sentiment Accuracy: Explaining the Baseline and How to Test It - Lexalytics. (n.d.). Retrieved July 7, 2023, from https://www.lexalytics.com/blog/sentiment-accuracy-baseline-testing/

Shrikhande, P., Setty, V., & Sahani, A. (2020). Sarcasm detection in newspaper headlines. *2020 IEEE 15th International Conference on Industrial and Information Systems (ICIIS)*, 483–487.

Thakur Sakshi and Singh, S. and S. M. (2020). Detecting Sarcasm in Text. In A. K. and M. P. and G. N. Abraham Ajith and Cherukuri (Ed.), *Intelligent Systems Design and Applications* (pp. 996–1005). Springer International Publishing.

Vaishanvi, S., Rajkaran, Y. P., Rahul, V., & Nirmal, L. M. (2022). Product Recommendation Using Sentiment Analysis. *8th International Conference on Engineering and Emerging Technologies, ICEET 2022*. https://doi.org/10.1109/ICEET56468.2022.10007234

Zain Naqvi, S. (n.d.). *SENTIMENT ANALYSIS OF PRODUCT REVIEWS AT DOCUMENT LEVEL*.

Zhang, B. J., Quick, R., Helmi, A., & Fitter, N. T. (2020). Socially assistive robots at work: Making break-taking interventions more pleasant, enjoyable, and engaging. *IEEE International Conference on Intelligent Robots and Systems*, 11292–11299. https://doi.org/10.1109/IROS45743.2020.9341291