Journal of Science and Technology
ISSN: 2456-5660 Volume 9, Issue 1 (Jan -2024)
www.jst.org.in *DOI:https://doi.org/10.46243/jst.2024.v9.i1.pp1 76-88*

SARCASMET: EXTENSION OF LEXICON ALGORITHM FOR EMOJI-BASED SARCASM DETECTION FROM TWITTER DATA

M. Sravan Kumar Babu[1], S. Sathvika[2], G. Sai Nikitha[2], P. Lahari[2]
[1]Assistant Professor,[2]UG Students, Department of Cyber Security Engineering.
[1,2]Malla Reddy Engineering College for Women, Maisammaguda, Dhulapally, Kompally, Secunderabad-500100, Telangana, India.

To Cite this Article

M. Sravan Kumar Babu, S. Sathvika, G. Sai Nikitha, P. Lahari, **"SARCASMET: EXTENSION OF LEXICON ALGORITHM FOR EMOJI-BASED SARCASM DETECTION FROM TWITTER DATA"** *Journal of Science and Technology, Vol. 09, Issue 1 - Jan 2024, pp76-88*

Article Info

Received: 25-11-2023 Revised: 05 -12-2023 Accepted: 15-1-2024 Published: 25-1-2024

Abstract

Since the industrial revolution, the original way of communicating; face-to-face communication has been used as a model to develop the various ways of communicating known to date. Transposing the principles and codes of natural face-to-face communication to today's online communication is a major challenge for developers. In today's digital era, social media platforms like Twitter have become a hub for expressing opinions, emotions, and humor. Sarcasm, a form of verbal irony, is a prevalent means of communication on such platforms. However, detecting sarcasm in online content poses a significant challenge due to the absence of vocal intonations and facial expressions. This necessitates the development of reliable methods to automatically identify and understand sarcasm in tweets. Sarcasm makes use of positive lingual contents to convey a negative message. Different types of approaches have been developed to implement sarcasm detection on online communication platforms. However, the levels of efficiency of these approaches have been the principal worries of developers. Lexicon algorithm is used to determine the sentiment expressed by a textual content. This sentiment might be negative, neutral, or positive. It is possible to be sarcastic using only positive or neutral sentiment textual contents. Hence, lexicon algorithm can be useful but insufficient for sarcasm detection. It is necessary to extend the lexicon algorithm to come up with systems that would be proven efficient for sarcasm detection on neutral and positive sentiment textual contents. In this work, two sarcasm analysis systems both obtained from the extension of the lexicon algorithm have been proposed for that sake. The first system consists of the combination of a lexicon algorithm and a pure sarcasm analysis algorithm. The second system consists of the combination of a lexicon algorithm and a sentiment prediction algorithm. Finally, the proposed model aims to detect the sarcasm from the text and emotion icon with improved efficiency.

Keywords: Emoji-Based Sarcasm, Lexicon Algorithm, Twitter Data, Sarcasmet.

1. INTRODUCTION

The extension of the Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data is a cutting-edge approach to tackle the nuanced and ever-evolving phenomenon of sarcasm in online social media, with a specific focus on Twitter. Sarcasm in text is notoriously challenging to detect due to its reliance on context [1], tone, and subtlety, making it a perfect testbed for natural language processing and sentiment analysis techniques. The Lexicon Algorithm, a well-established method for sentiment analysis, forms the basis for this extension. In this advanced approach, researchers have incorporated a comprehensive emoji lexicon to enhance the algorithm's performance [2]. Emojis play a vital role in conveying sentiment and emotions in online communication, often being key indicators of sarcasm. By integrating an extensive emoji lexicon into the Lexicon Algorithm, this extension aims to improve the accuracy of sarcasm detection in Twitter data. The algorithm begins by preprocessing the Twitter data, which typically involves tokenization, stemming, and removing stopwords. It then identifies sentences or tweets containing potential sarcasm cues, such as negations, contrastive conjunctions, and sentiment-laden words [3]. The innovation lies in the algorithm's ability to consider emojis alongside these textual cues. Emojis, when used ironically or sarcastically, can completely reverse the sentiment of a sentence, and the algorithm is designed to recognize such patterns. Furthermore, the extension considers the surrounding context of tweets, considering user-specific patterns and common sarcasm structures in the Twitterverse. Machine learning techniques, such as supervised learning or neural networks [4], may be employed to fine-tune the algorithm's performance and adapt it to the evolving nature of sarcasm on Twitter.

The implications of this extended Lexicon Algorithm are far-reaching, with potential applications in sentiment analysis, brand monitoring, and social media trend analysis. Accurately detecting sarcasm in Twitter data can help improve our understanding of public sentiment, social trends, and user engagement, benefiting businesses, researchers, and social media platforms themselves. However, it's important to note that this field is dynamic, and ongoing research and adaptation are necessary to keep pace with the ever-changing landscape of online communication The motivation for extending the Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data stems from the growing significance of social media as a platform for communication, information dissemination, and sentiment analysis [5]. Twitter has emerged as a prominent medium where individuals express their thoughts, opinions, and emotions. Within this vast pool of textual data, the detection of sarcasm has become increasingly crucial. Sarcasm, characterized by the expression of one's meaning by using language that typically signifies the opposite, often relies on tone, context, and subtlety. In the digital realm, where facial expressions and vocal cues are absent, sarcasm poses a unique challenge for sentiment analysis algorithms. Traditional lexicon-based approaches have shown promise in detecting sentiment, but they often struggle with the complexity of sarcasm. Emojis, on the other hand, have become an integral part of online communication, serving as non-verbal indicators of sentiment and emotion. Users frequently employ emojis to add layers of meaning, including sarcasm, to their text [6]. Thus, the motivation for extending the Lexicon Algorithm lies in the recognition that emojis are valuable cues for discerning sarcasm within Twitter data. Moreover, the ability to accurately detect sarcasm in social media data has wide-ranging implications. It can empower businesses to gauge customer sentiment more accurately, helping them make informed decisions about product development and marketing strategies. Researchers can gain deeper insights into public opinions and trends, facilitating studies on social dynamics and sentiment evolution. Social media platforms themselves can benefit from improved content moderation and user experience enhancement by better understanding user intentions.

2. LITERATURE SURVEY

Despite sarcasm having long been studied in the social sciences, automated detection of sarcasm in text is a new area of study. Recently, the research community in the domain of natural language processing (NLP) and machine learning (ML) has been interested in automated sentiment classification [7]. An NLP-based technique uses linguistic corpora and language features to understand qualitative data. ML systems utilize unsupervised and supervised classification approaches based on tagged or unlabeled information to interpret sarcastic remarks.

An extensive dataset for sarcasm text detection was created by Khodak et al. [8]. Before comparing their findings with methods such as sentence vectorization, bag-of-words, and bag-of-bigrams, the authors performed hand annotation. They discovered that hand sarcasm detection outperformed other methods. Eke et al. [9] evaluated a range of earlier studies on sarcasm recognition. According to this review research, N-gram and part-of-speech tag (POS) methods were the most frequently used feature extraction algorithms. Binary interpretation and word frequencies were used for feature representation, nevertheless. The review also noted that the chi-squared test and information gain (IG) method were frequently employed for feature selection. In addition, the maximum entropy, naive Bayes, random forests (RF), and support vector machine (SVM) classification techniques were used. A review of several “Customized Machine Learning Algorithms” (CMLA) and “Adapted Machine Learning Algorithms” (AMLA) utilized in sarcasm detection research was also published by Sarsam et al. in [10]. Their findings concurred with those of Eke et al. They found that CNN-SVM can perform better when both lexical and personal characteristics are used. The SVM performs better using lexical, frequency, pragmatic, and part-of-speech labeling.

Prior to this, models based on machine learning were created; these models primarily acquire language features and train these qualities over classifiers learned by machine learning. Machine learning has been used for content-based features by Keerthi Kumar and Harish [11]. Before submitting the data to the clustering method for various filters, the authors used feature selection approaches such as “Information Gain” (IG), “Mutual Information” (MI), and chi-square. An SVM was used to categorize data at the very end. Pawar and Bhingarkar [12] employed an ML classification model for sarcasm detection on a related subject. They collected data on recurring themes, punctuation, interjections, and emotions. Using the random forest and SVM techniques, these feature sets for classification were learned.

A different strategy was to go beyond individual words and understand the meaning of the sentences. “Long short-term memory” (LSTM), bidirectional LSTM, gated attention modules, or directed attention might be used primarily to perform this task. Sarcasm in Twitter posts was identified using neural network architecture by Ghosh and Veale [13]. They created an architecture using CNN and bidirectional LSTM. There were input data embeddings offered; the first were data from Twitter, and the second were author-related data. Word vectors were sent to bidirectional LSTM after being routed through CNN layers for feature learning. Dense layers received the bidirectional LSTM output vectors and used the SoftMax function to perform classification. Similarly, Ghosh, Fabbri, and Muresan [14] identified sarcastic comments using multiple LSTM and contextual data. Before judging whether or not anything was sarcastic, they studied the last tweets to grasp the perspective of the present statement. Liu et al. [15] used tweets to detect sarcasm and solely utilized content elements such as POS, punctuation, numerical data points, and emoticons.

3. Proposed System

3.1 Overview

Lexicon algorithm is used to determine the sentiment expressed by a textual content. This sentiment might be negative, neutral, or positive. It is possible to be sarcastic using only positive or neutral sentiment textual contents. Hence, lexicon algorithm can be useful but yet insufficient for sarcasm detection. It is necessary to extend the lexicon algorithm to come out with systems that would be proven efficient for sarcasm detection on neutral and positive sentiment textual contents. In this paper, two sarcasm analysis systems both obtained from the extension of the lexicon algorithm have been proposed for that sake. The first system consists of the combination of a lexicon algorithm and a pure sarcasm analysis algorithm. The second system consists of the combination of a lexicon algorithm and a sentiment prediction algorithm. Figure 4.1 shows the proposed system model. The detailed operation illustrated as follows:

Step 1. Twitter Dataset: The research work begins with the collection of a dataset containing tweets from the social media platform Twitter. This dataset serves as the primary source of textual content for analysis. It comprises a mix of tweets with varying sentiments, including positive, neutral, and potentially sarcastic ones.

Step 2. Data Preprocessing: Before any analysis can take place, the Twitter dataset undergoes a series of preprocessing steps to clean and prepare the text for further processing. Data preprocessing involves tasks such as:

- Tokenization: Breaking down the tweets into individual words or tokens.
- Stemming: Reducing words to their base or root form.
- Stopword Removal: Eliminating common, non-informative words (e.g., "and," "the," "in").
- Special Character Removal: Stripping away punctuation and special characters.

Step 3. Extension of Lexicon Algorithm: In this step, the researchers extend the traditional Lexicon Algorithm, which is primarily used for sentiment analysis. The extension aims to make the Lexicon Algorithm more suitable for detecting sarcasm, even in tweets with positive or neutral sentiments.

Step 4. Sentiment Identification: The extended Lexicon Algorithm is then employed to identify the sentiment expressed in each tweet. This involves determining whether a tweet conveys a positive, negative, or neutral sentiment based on the polarity of the individual terms within the tweet. This sentiment identification provides a baseline understanding of the sentiment in the tweets, which includes both non-sarcastic and potentially sarcastic content.

Step 5. Sentiment Analysis: Building upon the sentiment identification, the researchers perform a more in-depth sentiment analysis. This analysis goes beyond mere polarity determination and seeks to recognize nuanced expressions of sentiment, including sarcasm. The Lexicon Algorithm's extension allows it to consider the possibility of sarcasm within tweets, especially those with positive or neutral sentiments.

Step 6. Prediction of Tweet (Normal and Sarcastic): In this final step, the research work proposes two systems for sarcasm detection within tweets that primarily express positive or neutral sentiments:

a. Lexicon Algorithm + Pure Sarcasm Analysis Algorithm: The first system combines the extended Lexicon Algorithm with a dedicated pure sarcasm analysis algorithm. This combination aims to enhance sarcasm detection in tweets that may not contain obvious negative sentiment cues.

b. Lexicon Algorithm + Sentiment Prediction Algorithm: The second system combines the extended Lexicon Algorithm with a sentiment prediction algorithm. This combination seeks to predict whether a tweet is normal (non-sarcastic) or sarcastic, even when the sentiment expressed is positive or neutral.

These two systems leverage the extended Lexicon Algorithm's capabilities to identify subtle sarcasm cues within tweets that might otherwise be missed by traditional sentiment analysis techniques.

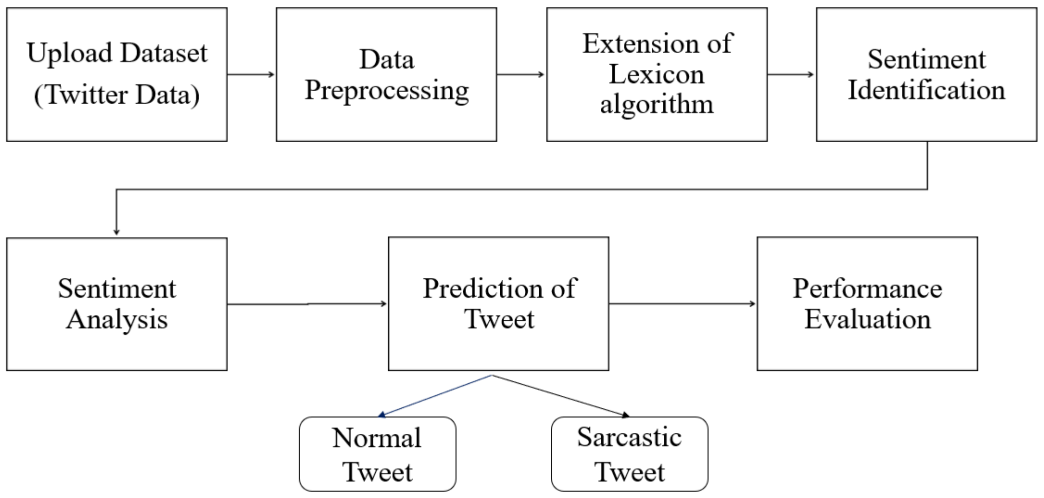


Figure 1. Proposed System model.

3.2 Data preprocessing

Data preprocessing is a process of preparing the raw data and making it suitable for a machine learning model. It is the first and crucial step while creating a machine learning model. When creating a machine learning project, it is not always a case that we come across the clean and formatted data. And while doing any operation with data, it is mandatory to clean it and put in a formatted way. So, for this, we use data pre-processing task.

Why do we need Data Pre-processing?

A real-world data generally contains noises, missing values, and maybe in an unusable format which cannot be directly used for machine learning models. Data pre-processing is required tasks for cleaning the data and making it suitable for a machine learning model which also increases the accuracy and efficiency of a machine learning model.

Step 1: Load the Hotel Review Dataset: In this step, load the dataset into your data analysis environment. This dataset typically includes text reviews (the input) and corresponding labels (the output), which can be sentiments like "positive" or "negative." The goal is to train a model to predict these labels based on the text.

Step 2: Data Cleaning: Data cleaning involves several sub-steps: Removing Special Characters and Punctuation: Special characters (e.g., @, $, %) and punctuation (e.g., !, ?, .) are often irrelevant to sentiment analysis and can be removed to focus on the actual text content. Handling Irrelevant Information: Sometimes, there might be metadata or other information in the text data that's not relevant to the analysis. You should remove such information to concentrate on the review text itself.

Step 3: Tokenization:

- Tokenization is the process of splitting the text into smaller units, such as words or phrases (tokens). This step is crucial because it breaks down the text into manageable pieces for further analysis.
- For example, the sentence "I love this hotel" would be tokenized into ["I", "love", "this", "hotel"].

Step 4: Convert Text to Lowercase:

- Converting all text to lowercase ensures consistency in your text data. It prevents the model from treating "good" and "Good" as two different words, which could lead to incorrect feature extraction and modeling.
- For example, "Good" and "good" should be treated as the same word.

Step 5: Remove Stop Words:

- Stop words are common words in a language that often don't carry significant meaning and can be safely removed to reduce noise in the data. Examples include "the," "is," "and," "in," etc.
- Removing stop words can help improve the efficiency of the model and reduce the dimensionality of the data without losing much valuable information.

Step 6: Apply Stemming or Lemmatization:

- Stemming and lemmatization are techniques used to reduce words to their root form, which helps in standardizing words and improving feature extraction.
- Stemming: It involves removing suffixes from words to obtain the word stem. For example, "jumping" becomes "jump," "flies" becomes "fli," etc. Stemming is more aggressive but may result in non-words.
- Lemmatization: It is a more advanced technique that reduces words to their base or dictionary form (lemma). For example, "better" becomes "good," "running" becomes "run," etc. Lemmatization is more accurate but computationally expensive.
- The choice between stemming and lemmatization depends on your specific NLP task and dataset.

3.3 Dataset Splitting

In machine learning data pre-processing, we divide our dataset into a training set and test set. This is one of the crucial steps of data pre-processing as by doing this, we can enhance the performance of our machine learning model. Suppose if we have given training to our machine learning model by a dataset and we test it by a completely different dataset. Then, it will create difficulties for our model to understand the correlations between the models. If we train our model very well and its training accuracy is also very high, but we provide a new dataset to it, then it will decrease the performance. So, we always try to make a machine learning model which performs well with the training set and with the test dataset.

Training Set: A subset of dataset to train the machine learning model, and we already know the output.

Test set: A subset of dataset to test the machine learning model, and by using the test set, model predicts the output.

3.4 Extended Lexicon Algorithm

In this work, the lexicon algorithm has been extended in two ways to generate two systems that could be more efficient for sarcasm analysis, especially on neutral and positive sentiment textual contents.

First system: The first system as shown in Figure 2 is the combination of a lexicon algorithm and a pure sarcasm analysis algorithm. This system takes textual contents as input. These contents could be from various social media platforms like Twitter or Facebook. The textual contents are parsed into the lexicon algorithm for polarity computation. Then the positive sentiment contents are parsed into the pure sarcasm analysis algorithm for sarcasm detection. The final output of this system is a list of sarcastic and non-sarcastic lingual contents.

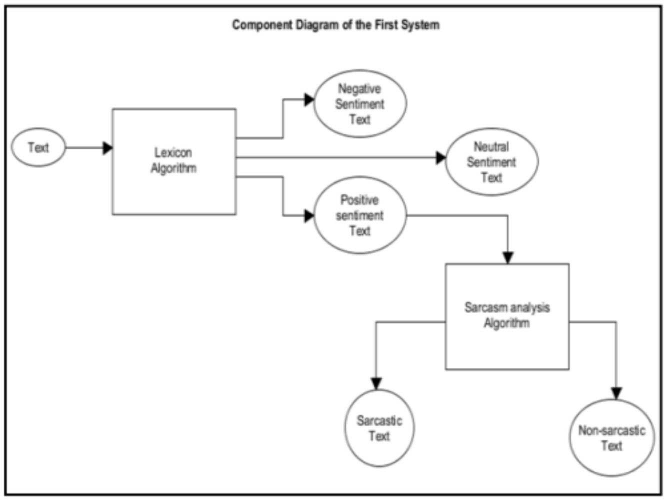


Figure 2. Component diagram of the first system. An overview of the arrangement of components of the first system.

Second system: The second system is the combination of a lexicon algorithm and a sentiment prediction algorithm. The lexicon algorithm is used here the same way as in the first system. The sentiment prediction algorithm consists of a mechanism that can predict the sentiment of a textual content that would be made under a specific environment. The sentiment prediction algorithm takes as input the details of the environment under which a lingual content would be made notably the state of the context, the author's knowledge of the domain he/she would talk about, the author's level of education, the author's personality, the author's relationship with his/her interlocutor. The sentiment prediction algorithm processes these details and predicts the sentiment of the textual content that would be formed under that environment. The results from both the algorithms are compared. In Case the results are different for a textual content, this later is classified as sarcastic else it is classified as non-sarcastic as shown in Figure 3.

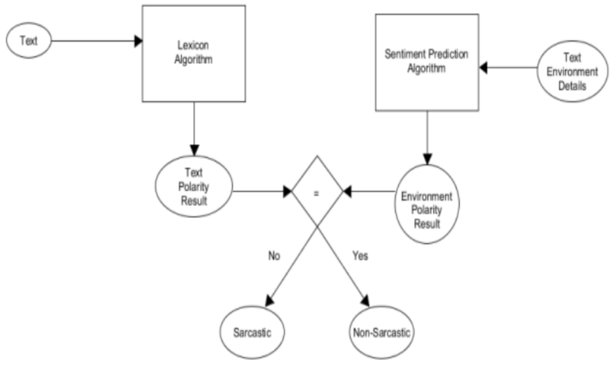


Figure 2. Component diagram of the second system. An overview of the arrangement of the components of the second system.

These environment details are processed based on a training data set that was formed as follow:

- Each environment detail had a polarity. The polarity could be negative, neutral or positive as shown in Table 4.I.
- An AND operation was performed in between the details polarities of each textual content environment to predict the sentiment of the textual content that would be made under each environment. The training data set of the sentiment prediction algorithm was formed of the environment details polarities and their predicted sentiment (TABLE II)
- P(A|B) is a conditional probability: the likelihood of event A occurring given that B is true.
- P(B|A) is also a conditional probability: the likelihood of event B occurring given that A is true.
- P(A) and P(B) are the probabilities of observing A and B independently for each other; this is known as the marginal probability.
- Naïve Bayes is mainly used for classification purposes. It is an algorithm that discriminates different objects based on certain features. This algorithm is built after the Bayes theorem which assumes that all features within a class are independent from one another and that is why it is known as ‘naive’.

4. RESULTS AND DISCUSSION

Figure 3 illustrates the graphical user interface (GUI) of the sarcasm detection application. It shows the buttons, text fields, and other elements that users interact with. Users can perform tasks such as uploading datasets, initiating preprocessing, running algorithms, and visualizing results through this interface. In Figure 4, the user interface is shown after the user has uploaded a dataset. The uploaded dataset contains tweets and associated data. The interface displays a confirmation message to inform the user that the dataset has been successfully uploaded along with the count of tweets in given dataset. Figure 5 presents the user interface displaying preprocessed data after applying the Natural Language Toolkit (NLTK) library for text processing. The interface shows cleaned and tokenized text, and potentially processed emojis. This step aims to prepare the data for further analysis. Figure 6 showcases the results obtained after applying lexicon-based sentiment analysis with polarity computation on the preprocessed data. The user interface displayed each tweet's polarity scores, indicating their positivity, negativity, and neutrality. The interface also indicates whether the tweet is categorized as positive, negative, or neutral.

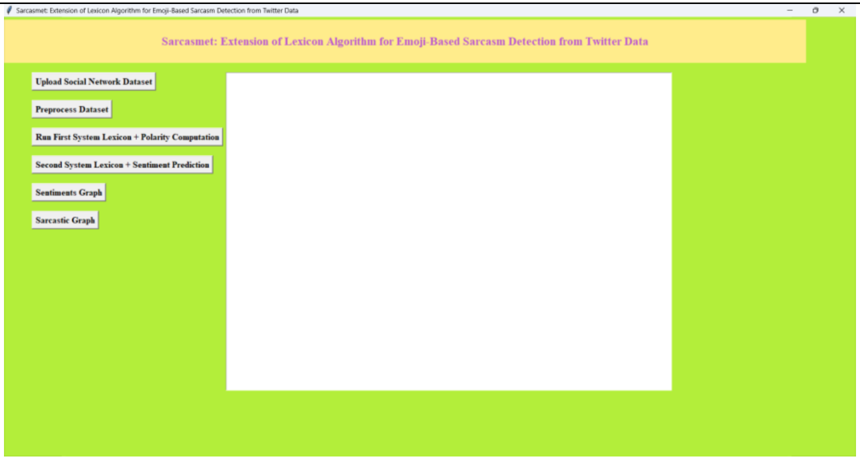
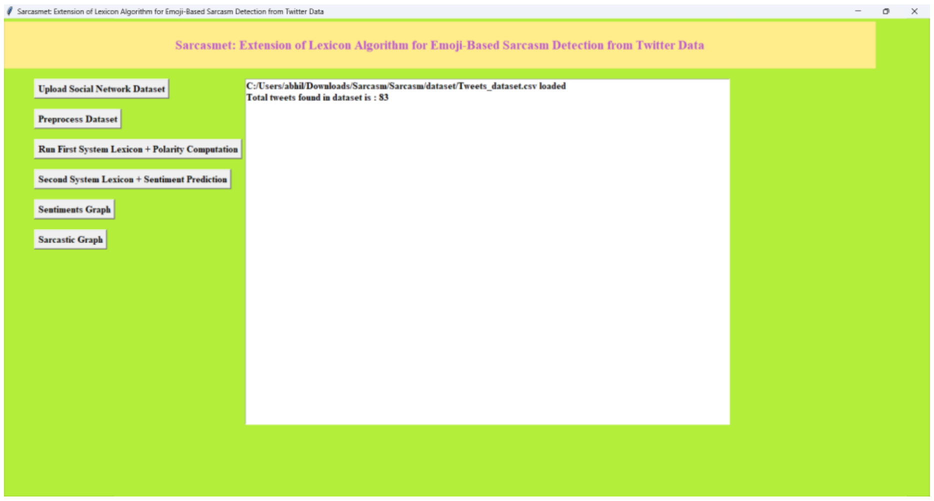
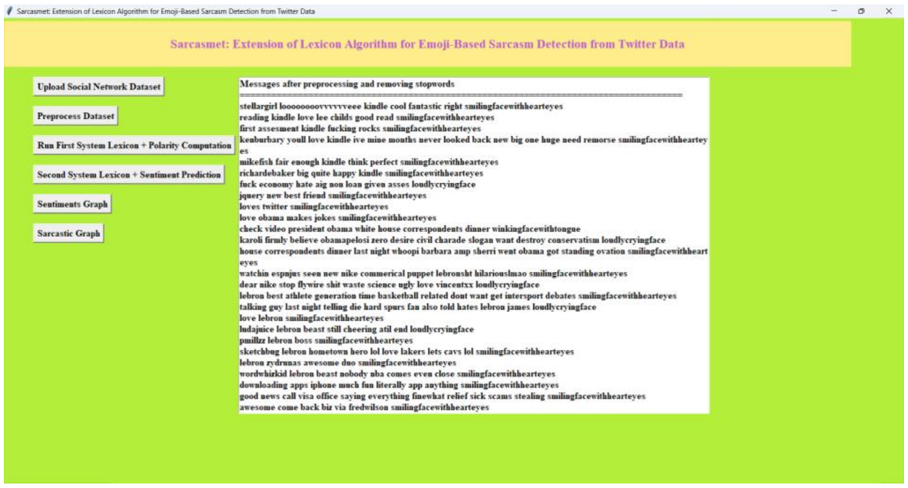

Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Upload Social Network Dataset
Preprocess Dataset
Run First System Lexicon + Polarity Computation
Second System Lexicon + Sentiment Prediction
Sentiments Graph
Sarcastic Graph

Figure 3: User interface application of proposed NLP-based sarcasm detection model.


Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Upload Social Network Dataset
Preprocess Dataset
Run First System Lexicon + Polarity Computation
Second System Lexicon + Sentiment Prediction
Sentiments Graph
Sarcastic Graph

```
C:/Users/abhil/Downloads/Sarcasm/Sarcasm/dataset/Tweets_dataset.csv loaded
Total tweets found in dataset is : 83
```

Figure 4: Illustration of user interface application after uploading the dataset.


Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Upload Social Network Dataset
Preprocess Dataset
Run First System Lexicon + Polarity Computation
Second System Lexicon + Sentiment Prediction
Sentiments Graph
Sarcastic Graph

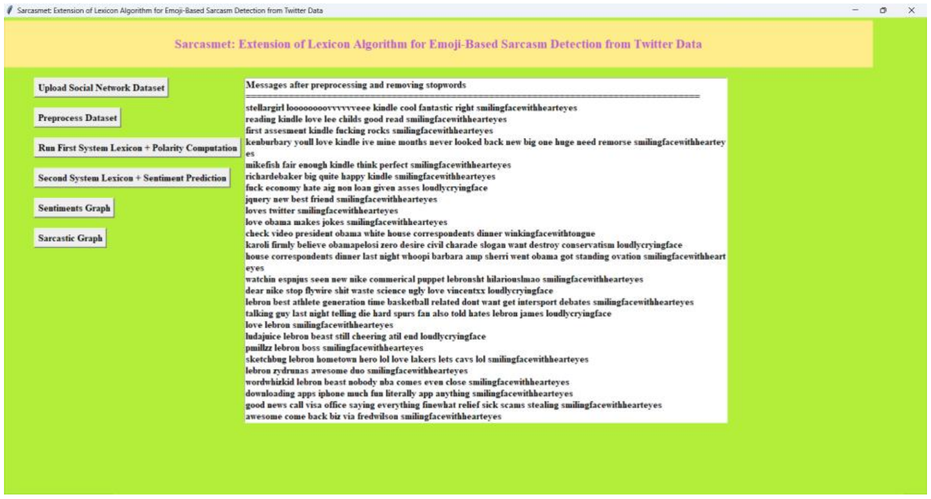
```
Messages after preprocessing and removing stopwords
=================================================================================================
stellargirl loooooooovvvvvveee kindle cool fantastic right smilingfacewithhearteyes
reading kindle love lee childs good read smilingfacewithhearteyes
first assesment kindle fucking rocks smilingfacewithhearteyes
kenburbary youll love kindle ive mine months never looked back new big one huge need remorse smilingfacewithheartey
es
mikefish fair enough kindle think perfect smilingfacewithhearteyes
richardebaker big quite happy kindle smilingfacewithhearteyes
fuck economy hate aig non loan given asses loudlycryingface
jquery new best friend smilingfacewithhearteyes
loves twitter smilingfacewithhearteyes
love obama makes jokes smilingfacewithhearteyes
check video president obama white house correspondents dinner winkingfacewithtongue
karoli firmly believe obamapelosi zero desire civil charade slogan want destroy conservatism loudlycryingface
house correspondents dinner last night whoopi barbara amp sherri went obama got standing ovation smilingfacewithheart
eyes
watchin espnjus seen new nike commerical puppet lebronsht hilariouslmao smilingfacewithhearteyes
dear nike stop flywire shit waste science ugly love vincentxx loudlycryingface
lebron best athlete generation time basketball related dont want get intersport debates smilingfacewithhearteyes
talking guy last night telling die hard spurs fan also told hates lebron james loudlycryingface
love lebron smilingfacewithhearteyes
ludajuice lebron beast still cheering atil end loudlycryingface
pmillzz lebron boss smilingfacewithhearteyes
sketchbug lebron hometown hero lol love lakers lets cavs lol smilingfacewithhearteyes
lebron zydrunas awesome duo smilingfacewithhearteyes
wordwhizkid lebron beast nobody nba comes even close smilingfacewithhearteyes
downloading apps iphone much fun literally app anything smilingfacewithhearteyes
good news call visa office saying everything finewhat relief sick scams stealing smilingfacewithhearteyes
awesome come back biz via fredwilson smilingfacewithhearteyes
```

Figure 5: Proposed UI application with preprocessed data using NLTK.

In Figure 7, the user interface presents the results of sentiment prediction using lexicon-based sentiment analysis. Alongside polarity scores, the UI also indicates whether a tweet is predicted as sarcastic or not based on sentiment and certain words.

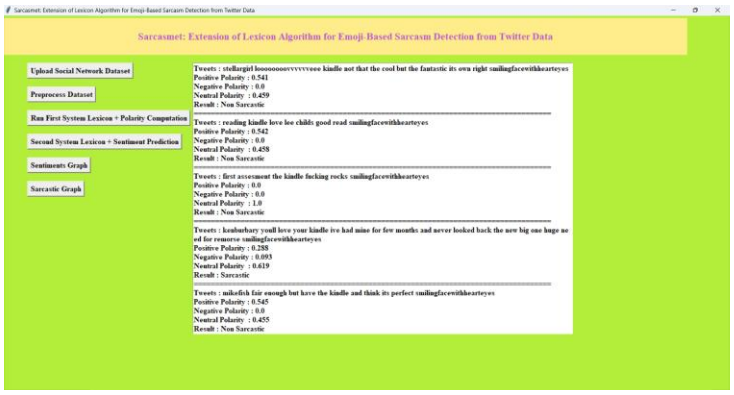
Figure 8 displays a pie chart that visualizes the distribution of sentiments in the dataset. The chart is divided into segments representing positive, negative, and neutral sentiments. The size of each segment indicates the proportion of tweets falling into each sentiment category.

Figure 9 shows a graph that visualizes the sarcasm predictions made by the proposed NLP-based extended lexicon model. The graph likely has two bars: one representing the number of sarcastic tweets and the other representing the number of non-sarcastic tweets. This visualization provides insights into the model's ability to detect sarcasm.


Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Upload Social Network Dataset
Preprocess Dataset
Run First System Lexicon + Polarity Computation
Second System Lexicon + Sentiment Prediction
Sentiments Graph
Sarcastic Graph

```
Messages after preprocessing and removing stopwords
==========================================================================================
stellargirl loooooooovvvvvveee kindle cool fantastic right smilingfacewithhearteyes
reading kindle love lee childs good read smilingfacewithhearteyes
first assesment kindle fucking rocks smilingfacewithhearteyes
kenburbary youll love kindle ive mine months never looked back new big one huge need remorse smilingfacewithheartey
es
mikefish fair enough kindle think perfect smilingfacewithhearteyes
richardebaker big quite happy kindle smilingfacewithhearteyes
fuck economy hate aig non loan given asses loudlycryingface
jquery new best friend smilingfacewithhearteyes
loves twitter smilingfacewithhearteyes
love obama makes jokes smilingfacewithhearteyes
check video president obama white house correspondents dinner winkingfacewithtongue
karoli firmly believe obamapelosi zero desire civil charade slogan want destroy conservatism loudlycryingface
house correspondents dinner last night whoopi barbara amp sherri went obama got standing ovation smilingfacewithheart
eyes
watchin espnjus seen new nike commerical puppet lebronsht hilariouslmao smilingfacewithhearteyes
dear nike stop flywire shit waste science ugly love vincentxx loudlycryingface
lebron best athlete generation time basketball related dont want get intersport debates smilingfacewithhearteyes
talking guy last night telling die hard spurs fan also told hates lebron james loudlycryingface
love lebron smilingfacewithhearteyes
ludajuice lebron beast still cheering atil end loudlycryingface
pmillzz lebron boss smilingfacewithhearteyes
sketchbug lebron hometown hero lol love lakers lets cavs lol smilingfacewithhearteyes
lebron zydrunas awesome duo smilingfacewithhearteyes
wordwhizkid lebron beast nobody nba comes even close smilingfacewithhearteyes
downloading apps iphone much fun literally app anything smilingfacewithhearteyes
good news call visa office saying everything finewhat relief sick scams stealing smilingfacewithhearteyes
awesome come back biz via fredwilson smilingfacewithhearteyes
```

Figure 6: Obtained results of proposed UI application with lexicon + polarity computation.


Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Sarcasmet: Extension of Lexicon Algorithm for Emoji-Based Sarcasm Detection from Twitter Data
Upload Social Network Dataset
Preprocess Dataset
Run First System Lexicon + Polarity Computation
Second System Lexicon + Sentiment Prediction
Sentiments Graph
Sarcastic Graph

```
Tweets : stellargirl loooooooovvvvvveee kindle not that the cool but the fantastic its own right smilingfacewithhearteyes
Positive Polarity : 0.541
Negative Polarity : 0.0
Neutral Polarity  : 0.459
Result : Non Sarcastic
==========================================================================================
Tweets : reading kindle love lee childs good read smilingfacewithhearteyes
Positive Polarity : 0.542
Negative Polarity : 0.0
Neutral Polarity  : 0.458
Result : Non Sarcastic
==========================================================================================
Tweets : first assesment the kindle fucking rocks smilingfacewithhearteyes
Positive Polarity : 0.0
Negative Polarity : 0.0
Neutral Polarity  : 1.0
Result : Non Sarcastic
==========================================================================================
Tweets : kenburbary youll love your kindle ive had mine for few months and never looked back the new big one huge ne
ed for remorse smilingfacewithhearteyes
Positive Polarity : 0.288
Negative Polarity : 0.093
Neutral Polarity  : 0.619
Result : Sarcastic
==========================================================================================
Tweets : mikefish fair enough but have the kindle and think its perfect smilingfacewithhearteyes
Positive Polarity : 0.545
Negative Polarity : 0.0
Neutral Polarity  : 0.455
Result : Non Sarcastic
```

Figure 7: Results of lexicon + sentiment prediction for proposed sarcasm detection model.

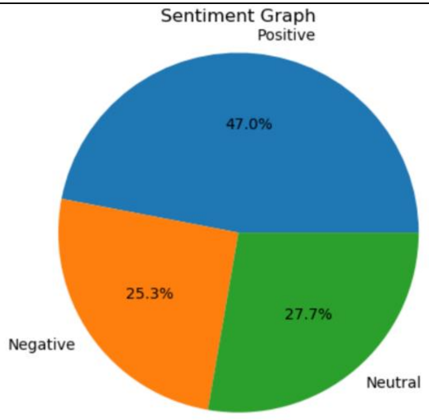


Figure 8: Pie chart of sentiment prediction on tweets dataset with different classes (positive, negative, and neutral).

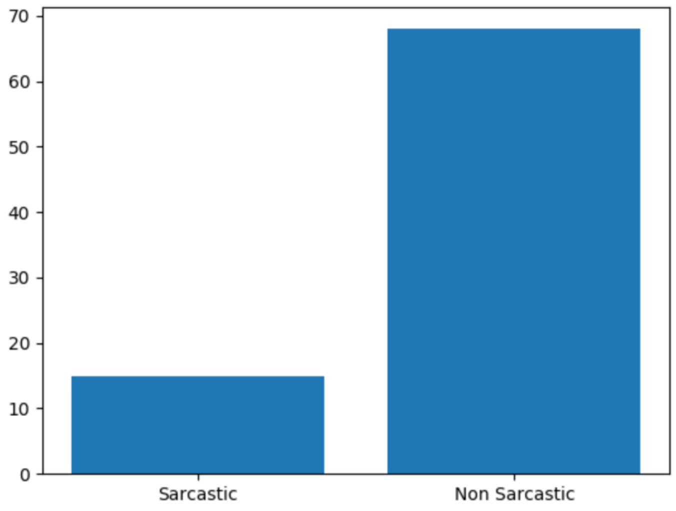


Figure 9: Sarcasm prediction graph using proposed NLP-based extended lexicon model.

5. Conclusion

The aim of this work was to propose ways to extend the lexicon algorithm in order to build systems that would be more efficient for sarcasm detection. This aim has been successfully met as two systems have been developed to address this situation. However, in the first system, it had been noticed that the training set of the sarcasm analysis algorithm must be relevant to the actual data that need to be analyzed in order to obtain meaningful results and to improve the accuracy of the system. The second system constitutes a vast area of study. Some work needs to be done in order to develop a system that would allow the collection of environmental details under which the textual contents would be made on social media platforms. A consolidated way of computing the sentiment polarity of the environments based on their details should also be developed.

References

1. Edwards, V.V. Sarcasm: What It Is and Why It Hurts Us. 2014. Available online:

https://www.scienceofpeople.com/sarcasm-why-it-hurts-us/ (accessed on 5 October 2021).

2. Rothermich, K.; Ogunlana, A.; Jaworska, N. Change in humor and sarcasm use based on anxiety and depression symptom severity during the COVID-19 pandemic. J. Psychiatr. Res. 2021, 140, 95–100.
3. Ezaiza, H.; Humayoun, S.R.; Al Tarawneh, R.; Ebert, A. Person-vis: Visualizing personal social networks (ego networks). In Proceedings of the 2016 CHI Conference Extended Abstracts on Human Factors in Computing Systems, San Jose, CA, USA, 7–12 May 2016; pp. 1222–1228.
4. Akula, R.; Garibay, I. Viztract: Visualization of complex social networks for easy user perception. Big Data Cogn. Comput. 2019, 3, 17.
5. Singh, B.; Sharma, D.K. Predicting image credibility in fake news over social media using multi-modal approach. Neural Comput. Appl. 2021, 34, 21503–21517.
6. Singh, B.; Sharma, D.K. SiteForge: Detecting and localizing forged images on microblogging platforms using deep convolutional neural network. Comput. Ind. Eng. 2021, 162, 107733.
7. Wallace, B.C. Computational irony: A survey and new perspectives. Artif. Intell. Rev. 2015, 43, 467–483.
8. Khodak, M.; Saunshi, N.; Vodrahalli, K. A large self-annotated corpus for sarcasm. In Proceedings of the Eleventh International Conference on Language Resources and Evaluation (LREC), Miyazaki, Japan, 7–12 May 2018.
9. Eke, C.I.; Norman, A.A.; Shuib, L.; Nweke, H.F. Sarcasm identification in textual data: Systematic review, research challenges and open directions. Artif. Intell. Rev. 2020, 53, 4215–4258.
10. Sarsam, S.M.; Al-Samarraie, H.; Alzahrani, A.I.; Wright, B. Sarcasm detection using machine learning algorithms in Twitter: A systematic review. Int. J. Mark. Res. 2020, 62, 578–598.
11. Keerthi Kumar, H.M.; Harish, B.S. Sarcasm classification: A novel approach by using Content Based Feature Selection Method. Procedia Comput. Sci. 2018, 143, 378–386.
12. Pawar, N.; Bhingarkar, S. Machine Learning based Sarcasm Detection on Twitter Data. In Proceedings of the 5th International Conference on Communication and Electronics Systems (ICCES), Coimbatore, India, 10–12 June 2020.
13. Ghosh, A.; Veale, T. Magnets for sarcasm: Making sarcasm detection timely, contextual and very personal. In Proceedings of the 2017 Conference on Empirical Methods in Natural Language Processing, Copenhagen, Denmark, 7–11 September 2017; pp. 482–491.
14. Ghosh, D.; Fabbri, A.S.; Muresan, S. Sarcasm analysis using conversation context. Comput. Linguist. 2018, 44, 755–792.
15. Liu, L.; Priestley, J.L.; Zhou, Y.; Ray, H.E.; Han, M. A2text-net: A novel deep neural network for sarcasm detection. In Proceedings of the IEEE First International Conference on Cognitive Machine Intelligence (CogMI), Los Angeles, CA, USA, 12–14 December 2019; IEEE: Piscataway, NJ, USA, 2019; pp. 118–126.