iJRASET
International Journal For Research in Applied Science and Engineering Technology

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY

Volume: 9 **Issue:** VIII **Month of publication:** August 2021

DOI: https://doi.org/10.22214/ijraset.2021.37663

www.ijraset.com

Call: 08813907089 | E-mail ID: ijraset@gmail.com

International Journal for Research in Applied Science & Engineering Technology (IJRASET)
ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

Sarcasm Detection in Text Data Using Glove Embedding

Samrudhi Naik[1], Amit Patil[2]
[1]*Research Scholar,* [2]*Assistant Professor, Department of Computer Engineering ,Goa College of Engineering Farmagudi, Goa University*

Abstract: ***Sarcasm is a way of expressing feelings in which people say or write something which is completely different or opposite to what they actually mean to say. Hence it is very difficult to identify sarcasm . It is usually an ironic or satirical remark tempered by humor. Mainly, people use it to say the opposite of what's true to make someone look or feel foolish. Understanding the sarcasm can improve the accuracy of sentiment analysis. Sentiment analysis (or opinion mining) is a natural language processing technique used to determine whether data is positive, negative or neutral. This helps in identifying what the opinions of users or individual or society are. In this project an attempt is made to develop a model to detect if a sentence is sarcastic or if it is not sarcastic.***

Keywords: Sarcasm detection, GloVe Embedding, LSTM, Natural Language Processing, Sentiment

I. INTRODUCTION

Nowadays, internet has become very popular and is easily available, as a result of which lot of opinions and discussions happen all over the internet. There are many social media platforms, blogposts, e-commerce sites and many other websites where such opinions or discussion are done. Analyzing the text data to find sentiments of users from such texts is a challenging task. Sentiment analysis is one of the methods to analyze the text data. The text data can be in the form of discussions, customer reviews ,comments on social media, feedback , blog posts, movie reviews etc. This generates a lot of data. This data can be used to do social media monitoring, customer support/feedback, product analysis , market research etc.[1].

However, due to the limitations of the unofficial language and characters used , it is very difficult to understand the opinions of users and conduct such an analysis. In addition, the presence of sarcasm is even more difficult: sarcasm is when a person says something that they don't really mean [2].

According to the Cambridge Dictionary, sarcasm means the use of remarks that clearly mean the opposite of what they say, made in order to hurt someone's feelings or to criticize something in a humorous way.[3]

Sarcasm detection is a very narrow research field in NLP. It is very difficult to detect or identify sarcasm from a text as there are various factors on which it depends like the tone in which the sentence is said, the context of discussion etc. In this paper an attempt is done to determine if a text data is sarcastic or not using deep learning.

II. LITERATURE SURVEY

Ellen Riloff et.al [4] identified sarcasm by considering situation and sentiment phrases. Negative situation with positive phrases may yield to sarcasm. They considered phrases that were limited to specific syntactic structures.

S. Muresan et.al[5] detected and classified sarcastic tweets using machine learning ,also impact of vocabulary and practical factors on the performance, and to classify positive and negative tweets were investigated

Bharti et.al[6] proposed two approaches. The first approach was to detect sarcasm by using the occurrence of interjection word and the other approach was parsing based lexicon generation algorithm. They then combined both of these approaches to detect sarcasm.They used f Score for evaluation.

S. Muresan et.al [7] consider detecting sarcasm as a task of removing ambiguity meaning that the word can have a literal meaning or irony, and therefore the irony of the word.

Ashwin Rajadesingan et.al[8]proposed an approach to detect sarcasm by leveraging behavioral traits intrinsic to users expressing sarcasm. They used the user's past tweets to identify these behavioral traits. Observational, Behavioral sciences and Physiological aspects were considered.

Basak et al. [9] used SVM classifier to classify public shaming tweets. They automated this task of classification .The different classes used were abusive, shaming tweets etc.

III.PROBLEM STATEMENT

Sarcasm occurs most often in user-generated content such as Facebook comments, tweets, etc. It arises when people want to express their negative sentiments using positive words. It can be difficult for machines as well as for humans to understand sarcasm. Common topics, interests, and historical information must be shared between two people to make sarcasm available. Some believe that sarcasm is a sign of high intelligence ,but it's not. Instead wit is a sign of high intelligence. Sarcasm can affect an individual emotionally.[10] There are some attempts done previously to detect sarcasm. If detection of sarcasm is not done, then it can mislead people as they might think that the sarcastic sentence is truly what the person means to say. This can lead to unwanted arguments. The aim of this paper is to detect sarcasm from text data. It will identify if a given sentence is sarcastic or not.

IV.PROPOSED SOLUTION

Since there are no specific rules to detect sarcasm in a sentence it becomes a very difficult task to detect features of sarcasm. Sarcasm is also context based and therefore it might make sense to people who know the context while confusing the people who are not aware of the concept. It also depends on situation and time and its meaning can change with situation and time. This problem can be solved by using neural networks which can be used to implicitly extract features.

In this approach a Bi-directional LSTM model is proposed to detect sarcasm. The dataset used is obtained from Kaggle. The dataset contains news headlines , since news headlines have a great chances of containing sarcasm in them and hence can be used in detection of sarcasm. After obtaining the dataset , it was pre-processed. During the preprocessing of the dataset the removal of punctuations, special symbols, lowering, stop word removal and tokenization was performed. After the preprocessing is done , we obtain a clean dataset. Then the dataset is split into train and test data. Next, to obtain the word vectors, we use word embedding technique GloVe word embedding. This technique is applied to the preprocessed dataset. The vectors obtained by this method are then given to the Bi-LSTM model. The model is trained and the performance metrics are calculated to check the performance of the model. Lately, the classification is done. An input sentence is taken and detected if it is sarcastic or not sarcastic.

Therefore the architecture of the proposed model consists of 1)Obtaining the dataset 2)Preprocessing the dataset 3)Word embedding 4)Bi-directional LSTM model.

'Fig. 1.' depicts the general flow diagram of the proposed sarcasm detection model.

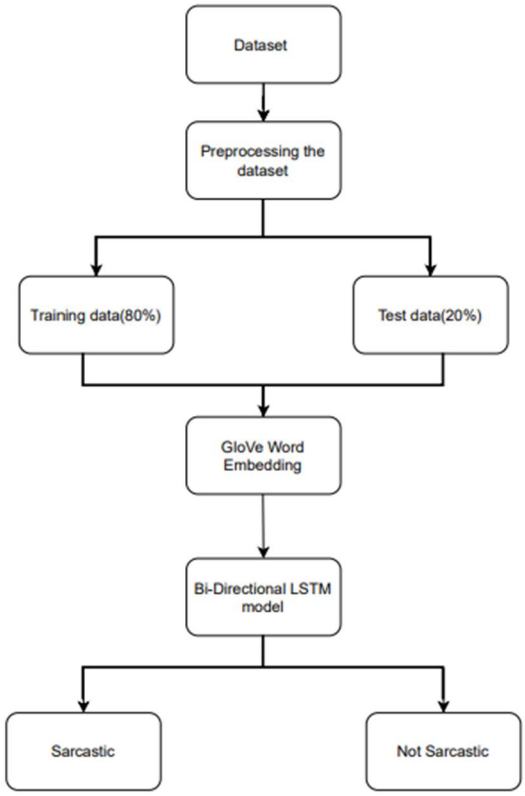


Fig.1 General Flow diagram of the proposed sarcasm detection model.

ISSN: 2321-9653; IC Value: 45.98; SJ Impact Factor: 7.429
Volume 9 Issue VIII Aug 2021- Available at www.ijraset.com

A. Dataset

The dataset used in this project is News Headlines dataset for sarcasm detection[12] obtained from Kaggle. It is a high quality dataset.

In most of the experiments related to sarcasm detection use of twitter datasets is done, but they contain limitations related to noise therefore in order to overcome this limitation we have used this News Headlines dataset for Sarcasm Detection . It is collected from two news website.

TheOnion which aims at producing sarcastic versions of current events and all the headlines from News in Brief and News in Photos categories (which are sarcastic). It also contains real (and non-sarcastic) news headlines from HuffPost.

This new dataset has following advantages over the existing Twitter datasets: There are no spelling mistakes and informal usage as the news headlines are written by professionals in formal manner. Also, the sole purpose of TheOnion is to publish sarcastic news, we obtain high-quality labels with much less noise as compared to Twitter datasets.

Each record in the dataset consists of three attributes:

is_sarcastic: 1 if the record is sarcastic otherwise 0

headline: the headline of the news article

article_link: link to the original news article.

The dataset consists of 25358 sarcastic headlines while 29970 non sarcastic headlines.

B. Preprocessing the Dataset

Next step is the preprocessing the dataset. Before giving the data to the model preprocessing of the data is to be done. The preprocessing includes removal of punctuations, special symbols, lowering, stop word removal and tokenization.. We have also done padding in order to obtain equally sized sequences. After the preprocessing of the data is done , the dataset is splitted into training data(80%) and testing data(20%).

C. Word Embedding

In natural language processing, word embedding is a term used for the representation of words for text analysis, typically in the form of a real-valued vector that encodes the meaning of the word such that the words that are closer in the vector space are expected to be similar in meaning[11].

The words are put to a vector space by word vectors. The similar words cluster together while the words which are different repel. The word embedding that we have used is Glove[Global Vectors].

GloVe is a very powerful word vector learning technique[13]. The advantage of using GloVe is because , it does not rely only on local statistics(local context information of words), but also incorporates global statistics (word co-occurrence) to obtain word vectors. It basically learns vectors or words from their co-occurrence information, i.e. how frequently they appear together in large text corpora.GloVe allows for parallel implementation, which means that it’s easier to train over more data.

D. The Model

The model used is Bidirectional LSTM.

A bi-directional LSTM , is a process of making any neural network have the sequence information in both the directions backwards and forwards.

Bi-LSTM is different from regular LSTM as the input flows in two directions[14]. This helps to preserve the future and the past information.

The model is implemented using tensorflow. The model contains embedding layer, LSTM layer, dropout is also added. The activation used is sigmoid activation . The loss is obtained by using binary cross entropy function. We have used the adam optimizer in this model.

V. RESULTS

The dataset consists of 25358 sarcastic headlines while 29970 non sarcastic headlines. ‘Fig.2’ shows the number of sarcastic and non sarcastic headlines in the dataset.

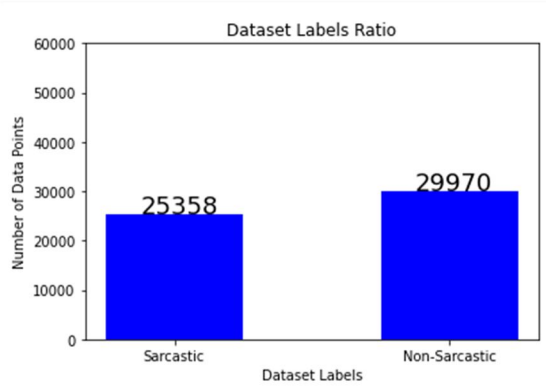


Fig. 2. Dataset Labels Ratio

'Fig. 3.'shows a word cloud of the words which have occurred frequently in the sarcastic headlines.

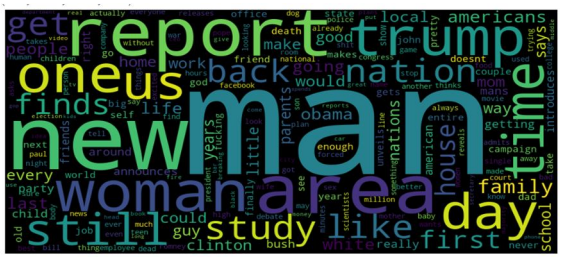


Fig.3 Word cloud depicting words most occurred in sarcastic sentences

The model was trained for 25358 sarcastic headlines and 29970 non sarcastic headlines.The model has received an accuracy of 96% after training the model for 25 epochs.

Fig.3 Training loss and validation loss

From the above 'Fig. 3.' we can see that both the training loss and the validation loss is decreasing with the increasing number of epochs and we can also train it for some more epochs as there is no overfitting.

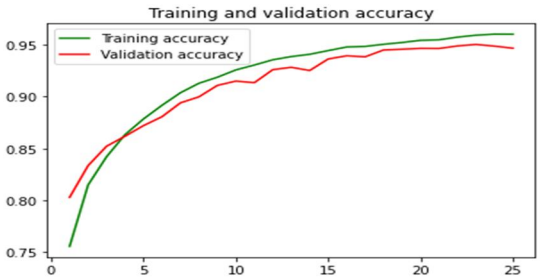


Fig. 4. Training accuracy and validation accuracy.

The above 'Fig. 4.' shows the training accuracy and the validation accuracy of the model for 25 epochs.

The performance is measured using the accuracy , precision and recall.

1) *Accuracy:* It shows the overall accuracy of the instances which are correctly classified to the total number of the instances. It is calculated by the following formula:

$$Accuracy = \frac{TP+TN}{TP+TN+FP+FN} \qquad (1)$$

Where, TP = true positive, TN = true negative, FP = false positive, FN = false negative.

2) *Precision:* It represents the percentage of relevant sarcastic headlines. That is, it measures the amount of headlines categorized as sarcastic against the total number of headlines classified as sarcastic. It is calculated by the following formula:

$$Precision = \frac{TP}{TP+FP} \qquad (2)$$

3) *Recall:* It represents the percentage of relevant sarcastic headlines that have been searched. That is, against the total number of sarcastic headlines, measured the number of headlines that are normally classified as sarcastic. It is calculated by the following formula:

$$Recall = \frac{TP}{TP+FN} \qquad (3)$$

Model	Accuracy	Precision	Recall
Bi-directional LSTM	96%	95%	95.3%

Results

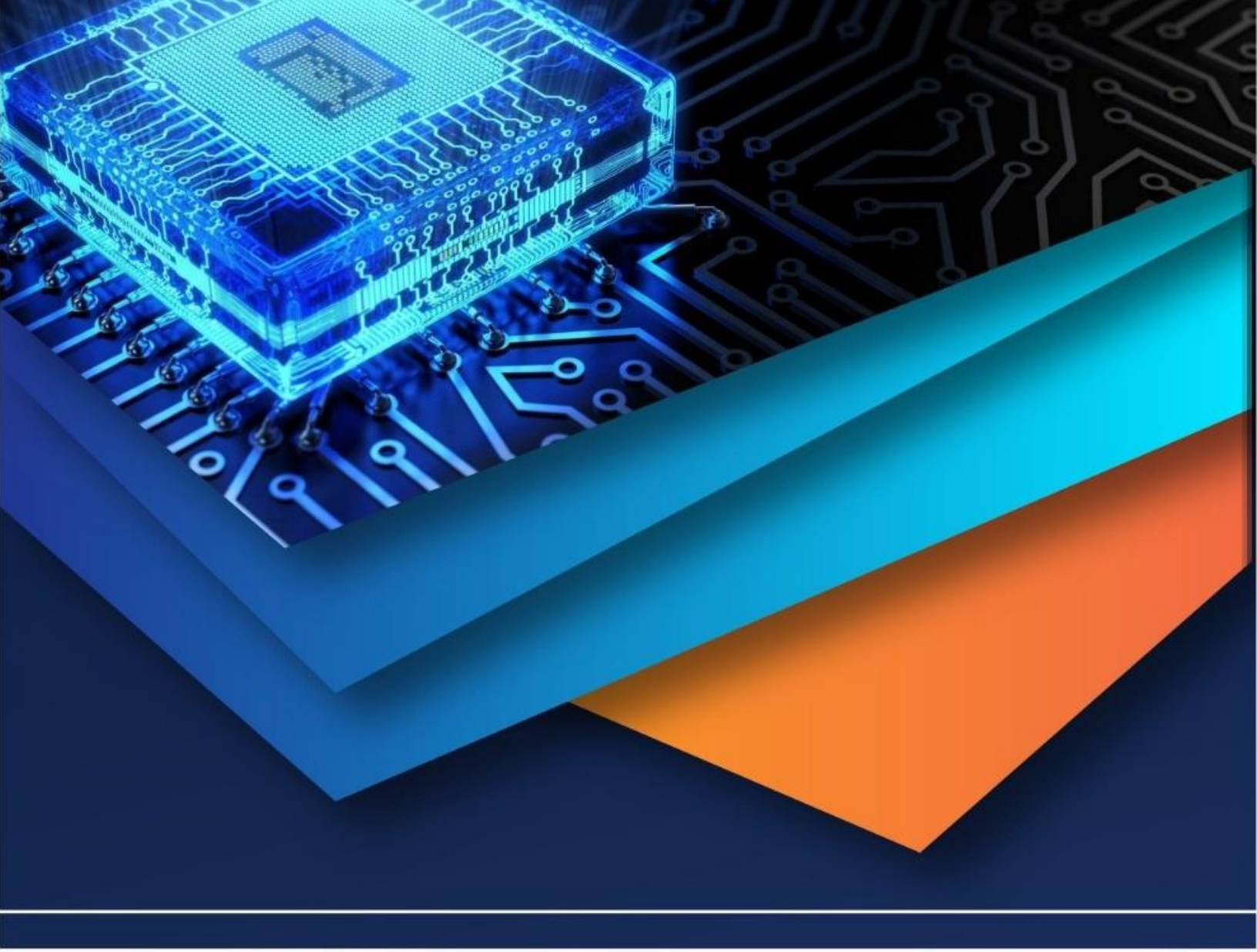
'Table I' shows the results obtained by the model. According to the results obtained the model gives an accuracy of 96% which is clearly better than the discrete model algorithms used for sarcasm detection.

VI. CONCLUSION

Sarcasm detection is a challenging task in the field of NLP. It is very difficult to detect sarcasm as it depends on various factors. In this paper an attempt is made to detect sarcasm from text data. We have proposed an approach using Bi-directional LSTM and GloVe word embedding. For future work we can work on a larger dataset for better output and also an API can be built to detect if there is sarcasm in text or not.

REFERENCES

[1] https://monkeylearn.com/blog/sentiment-analysis-applications/
[2] Yi Tay†, Luu Anh Tuan, Siu Cheung Huiφ, JianSuδ, "Reasoning with Sarcasm by Reading Inbetween"arXiv:1805.02856v1[cs.CL] 8 May 2018
[3] https://dictionary.cambridge.org/dictionary/english/sarcasm
[4] E. Riloff, A. Qadir, P. Surve, and Silva."Sarcasm as contrast between a positive sentiment and negative situation." In EMNLP, pages 704–714. ACL, 2013.
[5] S. Muresan, R. Gonzalez-Ibanez, D. Ghosh, and N. Wacholder, "Identification of nonliteral language in social media: A case study on sarcasm," J. Assoc. Inf. Sci. Technol., Jan. 2016.
[6] S. K. Bharti, K. S. Babu, and S. K. Jena, "Parsing-based sarcasm sentiment recognition in twitter data," in Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015. ACM, 2015, pp. 1373–1380
[7] D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or not: Word embeddings to predict the literal or sarcastic meaning of words," in Proc. EMNLP,Sep. 2015
[8] Ashwin Rajadesingan, Reza Zafarani, and Huan Liu. 2015. Sarcasm Detection on Twitter: A Behavioral Modeling Approach." In Proceedings of the Eighth ACM International Conference on Web Search and Data Mining Association for Computing Machinery, New York, NY, USA, 97–106.
[9] R. Basak, S. Sural, N. Ganguly and S. K. Ghosh, "Online Public Shaming on Twitter: Detection, Analysis, and Mitigation," in IEEE Transactions on Computational Social Systems, vol. 6, no. 2, pp. 208-220, April 2019, doi: 10.1109/TCSS.2019.2895734.
[10] https://www.goodtherapy.org/blog/the-problem-with-sarcasm-0815185
[11] https://en.wikipedia.org/wiki/Word_embedding
[12] https://www.kaggle.com/rmisra/news-headlines-dataset-for-sarcasm-detection
[13] https://medium.com/sciforce/word-vectors-in-natural-language-processing-global-vectors-glove-51339db89639
[14] https://analyticsindiamag.com/complete-guide-to-bidirectional-lstm-with-python-codes/



10.22214/IJRASET

45.98

IMPACT FACTOR: 7.129

IMPACT FACTOR: 7.429

INTERNATIONAL JOURNAL FOR RESEARCH

IN APPLIED SCIENCE & ENGINEERING TECHNOLOGY