Sarcasm Detection on Twitter Data

Shaila S G, Suman Patra, Joycelita Dias

Department of Computer Science, Dayananda Sagar University,

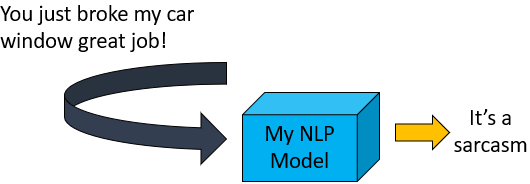
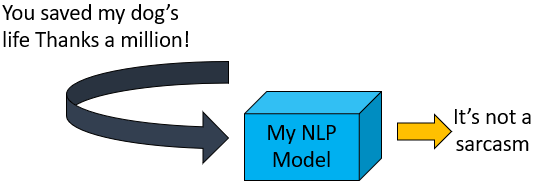
Bengaluru, Karnataka, India  
shaila-cse@dsu.edu.in. sumanpatra151215@gmail.com

**Abstract:**

Nowadays, due to rapid growth in the web technologies and internet usage, there will be lots of data generated and that are available in the web pages. People are using social media wed sites to connect to many peoples to exchange their knowledge, opinion and thoughts through Facebook, Instagram, Twitter and Google this are common platform to all. In the current generation, an individual opinion is expressed on social media. These views represent a variety of sentiments, which range from extremely positive to extremely negative. One of the many aims of a large number of organizations is to leverage those views in an attempt to market their products better. In this paper, we proposed the approach to classify the tweets into sarcastic and non-sarcastic sentences. Initially we do the preprocessing and feature extraction for tweeter dataset. Further we use Deep Learning algorithm for identifying sarcasm of the tweets. The classification algorithm like RNN and LSTM for - which uses depth wise separable convolutions to build lightweight deep neural networks which can be used in sarcasm detection. And we used Glove model obtaining vector representations for words. Here we are going with simple sentence to determine whether it’s sarcasm or not. The model is predicting accuracy of 92%.

**Keywords:** Sarcasm, RNN, LSTM, GloVe model

1. Introduction

Sarcasm is the caustic use of words, often in a humorous way. It can be reflected using rating of stars by providing a smaller number of stars and by giving some emojis. Twitter has become the biggest platform for people to express feelings, view and continuous occasions. Sarcasm is more common in the spots where there are capital letters, emojis, and interjection marks and so forth Sarcasm location is one of the unmistakable undertakings in opinion investigation. Traditional classification method involved methods such as LSTM which involves tuning parameters to set a design. However now deep learning method can be used to work on different conditions without trained. Detection of sarcastic content is vital to various NLP based systems such as text summarization and sentiment analysis. Sarcasm is used not only to make fun but also for criticizing other people, views, ideas etc. due to which sarcasm is very much used on twitter. Sarcasm can be conveyed in various ways like a direct conversation, speech, text etc. For example, live tweets and so on Regarding earlier years, the information of twitter has expanded a lot and consequently framing large information. Twitter has 315 million months to month dynamic clients, 82% of dynamic clients on portable and a great many tweets are being coursed through twitter consistently. Different associations as well as organizations are keen on twitter information for tracking down the perspectives on different individuals towards their items or occasions. Hence sarcasm is used in twitter. Here we also using NLP where in this language various techniques can be carried out. In given below example we see that the model is predicting whether the given sentence is sarcasm or not sarcasm. If we give input that is “You just broke my car window great job!” to NLP model its is predicting as it’s a sarcasm because the sentence we take here we expressing our negative opinion in a positive way. Similarly, if we give input that is “You saved my dogs life Thanks a million!” to NLP model it is predicted as it’s not a sarcasm because here we are expressing our opinion in positive way. In this way our model will predict Sarcasm or ****non-Sarcasm for simple sentences.

**Fig 1.1** shows whether it is Sarcasm or not

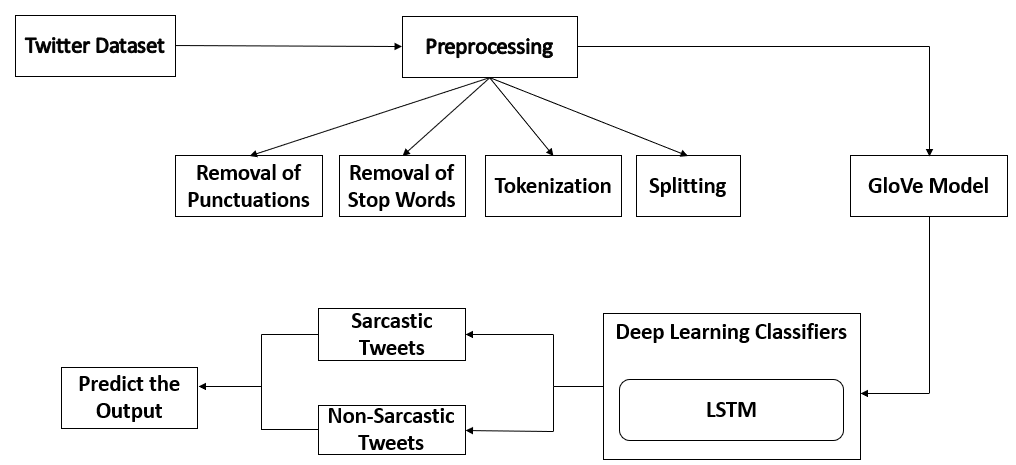
This section also discusses the literature on Sarcasm detection. The authors in [1] proposed propose a methodology to distinguish wry and non-wry tweets in view of the shoptalk and emoticons utilized in their tweets. They contemplated the characteristics for work related chatter and emoji utilized from the work-related conversation word reference and emoji word reference. Then, these characteristics are differentiated and different portrayal estimations like Random Forest, Gradient Boosting, Versatile Boost, Gaussian Naive Bayes, Logistic Relapse, and Decision Tree, to recognize the Sarcasm in tweets from the Twitter Streaming API. The author in [2] proposed a system to distinguish the mockery on twitter utilizing Simple Vector Machine (SVM), Maximum Entropy calculations. From the get go, they assembled the data and made into two datasets that are before adding the wry tweets to the readiness data and following adding wry tweets to the planning data. The author in [3] has proposed a strategy for identifying mockery in tweets utilizing assessment mining and robotized mockery identification. They removed the elements connected with feelings and accentuation and afterward the chi-square test is utilized to waitlist the most valuable highlights. In the second stage 200 top highlights are removed and joined with opinion related and accentuation related elements to track down mocking substance in the tweet. The author in [4] proposed irregular timberland and weighted outfit calculations to recognize the sarcasm in tweets and sober minded classifier to identify the feeling-based mockery. As demonstrated by the maker, Sarcasm is portrayed as the mix of good assessment or feeling joined to a pessimistic situation. Precision, survey, and precision is considered to learn the efficiency for both Irregular Forest Classifier and Weighted Ensemble estimation and saw that both the computations have nearly equivalated exactness. The author in [5] Programmed Sarcasm Detection: A Survey". In this work the creators suggested that portray datasets, approaches, patterns and issues in mockery location. they likewise talk about representative execution values, portray shared errands and give pointers to future work, as given in earlier works. As far as assets to comprehend the cutting edge, the review presents a few helpful outlines - most unmistakably, a table that total summarizes papers along various aspects like the sorts of elements, explanation strategies and datasets utilized. The author in [6] proposed an approach that classifies tweets. The approach used Hadoop Ecosystem to improve its efficiency and enhance scalability. Hadoop ecosystem use Map Reduce parallel processing paradigm on distributed processing platform. The author in [7] used various information analysis methods such as different hashtags analysis, Twitters network-topology, and various events over the network, influence impact for sentiment analysis. The above-mentioned approaches considered less features for classifying polarities for non-sarcastic sentences and lack in achieving good classification accuracy. The author in [8] used collected tweets on electronic products from the Twitter micro-blogging site and used preprocessing on tweets to classify sarcasm. The approach used unsupervised learning algorithm to determine the positive, negative and neutral polarities. In this paper, the approach presents two stages. Thus, the objective of the proposed approach lies in:

• Classification of sarcastic and non-sarcastic tweets

• Classification of the non-sarcastic tweets into positive polarity and negative polarity.

The rest of the paper is organized as follows. The related works are presented in the next section. In Sect. 2, we present the proposed approach and in Sect. 3, experimental results are discussion and in sect. 4, conclusion for this paper in the last section.

1. Proposed approach

The dataset considered in this approach consists of 9,104 tweets divided into sarcastic and not sarcastic tweets. The sarcastic tweets are labelled as #irony, #sarcasm. All the tweets are of English language and are unstructured. So, dataset cleaning and pre-processing are very much required. Then cleaning and preprocessing there we removed the unwanted, duplicated data. In data processing it is a task that convert the data from one given from to one more meaningful form. That makes it more meaning full and informative. It used algorithms to perform mathematical modeling and statistical knowledge. This entire process can be automated. Their output may in graphs, charts, tables and images. In Tokenizing we will be removing comma and punctuations symbols and in splitting we split the data into Training data (80%) and Testing data (20%). Then we will be performing feature there we are going with LSTM. Using a recurrent neural network (RNN) model for sarcasm detection because it automatically extracts features required for machine learning approaches. Along with the recurrent neural network, this model also uses long short-term memory (LSTM) cells on tensor flow to capture syntactic and semantic information over Twitter tweets to detect sarcasm. The GloVe model is an unsupervised learning algorithm is to obtaining vector representation of words and it also allows us to tale corpus of the text and transform each corpus into high dimensional space. Then finally its it will predict the output whether it is sarcasm or not.

**Fig 2.1:** Overview architecture of the Sarcasm detection

2.1 Twitter Dataset

This Project is based on the data set that is derived from tweets in the twitter data set. The data set has 9,104 tweets data contained Sarcasm and not Sarcasm points with no missing values. This Tweeter data is gathered and stored for the preprocessing. The below table represent the Tweeter dataset description. [https://data.word/dat-society/twitter -user-data]

|  |  |  |  |
| --- | --- | --- | --- |
| Data Set | No of data contained | Sarcasm data | Not Sarcasm data |
| Twitter Data | 9,104 | 6,482 | 2,622 |

**Table 2.1:** Dataset Description

2.2 Preprocessing

The preprocessing is the primary step in sentiment analysis. In preprocessing first, we need to do data cleaning that means it may have many irrelevant and missing parts to handle this it involves handling of missing data and noisy data and duplicated data. Once Data cleaning is done next, we need to remove all the punctuations means it help the NLP model to treat each sentence equally. For example, if we consider Data and Data! Here we are treated equally after removal of punctuations. Then perform removal of stop words its generally filtered out before processing into NLP model. It also doesn’t add much information about sentences. In Tokenization we will breaking the sequence of strings into small pieces such as words, keywords, phrases, symbols and other elements also replacing sensitive data with unique identification of symbols. In splitting it’s a process of dividing text into sentence. We can also splitted to sentences by “.” Or “/n” characters.

2.3 GloVe Model

Global Vectors (GloVe) is an unsupervised learning algorithm that represents words as vectors. Aggregated global word to word co-occurrence statistics from a corpus is used for training. The representations form linear substructures of the vector space. GloVe offers pre-trained word vectors for a variety of subjects. This study made use of the Twitter vector which contained 2 billion tweets, 27 billion tokens, 1.2 million vocabularies uncased with 50 dimensions.

2.4 Feature Extraction

In feature extraction stage the proposed approach uses RNN and LSTM which is a deep neural network for feature extraction and classification. A function is used to find sentiment score is formed. In this function, a word, its pos-tag, and a flag are passed. The flag is passed to identity if we are calculating the score for positive words or negative words, words having highly positive or negative emotional content. For the calculation of each feature the tweet is tokenized in words and then for each word the count is calculated: Positive word count: for each word, sentiment score is calculated using senti synset module in wordnet in the NLTK library which gives the sentiment score. If the score is greater than zero then it is treated as a positive word. Negative word count: it is calculated the same as positive word count but if the score is less than zero then it is treated as a negative word.

2.5Deep Learning Classifier

This is the third stage in sentiment analysis. Here, the proposed approach performs classification in two stages. In the first stage of classification, RNN and LSTM classifies tweets into sarcastic and non-sarcastic tweets. In second stage, classifying the non-sarcastic tweets into positive, negative and neutral polarities.

**2.5.1 Classification of LSTM model**

Long-Short Term Memory (LSTM) is composed of a cell, an input gate, an output gate and a forget gate. The cell stores the history of values at regular intervals while the gates are responsible for the flow of information.

• Forget gate: Determines useful data to keep from previous step

• Input gate: Determines what information is relevant to add from the current step

• Output gate: Determines what the next hidden state should be

The control flow is similar to the control flow in Recurrent Neural Networks (RNN). The output of the previous gate is fed as input, along with the original input, to the next gate. LSTM differs from standard feed forward neural networks as it has feedback connections. These connections enable the network to not only process single data points but sequences of data as well.

1. Results and Discussion

This proposed approach uses the Tweeter dataset [https://data.word/dat-society/twitter -user-data]. This dataset consists of 9,104 tweets which is divided further the sarcasm data contain 6,482 tweets and non-sarcasm data contain 2,622 datasets.

Here we discuss various metrics that are used for evaluating the performance of classifiers. Here, the metrics that have been used for experimentation are explained in detail. We used the Confusion Matrix to measure the performance of the classification. The confusion matrix has two dimensions, such as actual and predicted, where both dimensions have values like True Positive (TP), True Negative (TN), False Positive (FP), and False Negative (FN). TP represents positive class data points that were correctly classified by the model. TN represents negative class data points that were correctly classified by the model. FP represents negative class data points that were incorrectly classified as belonging to the positive class by the model. FN represents positive class data points that were incorrectly classified as belonging to the negative class by the model.

Further, we evaluated using precision which estimates correctly predicted cases that actually turned out to be positive. This is depicted in Eq. (1).

Precision = (1)

As we used benchmark dataset, recall is estimated to know how many of the actual positive cases were able to predict correctly with the proposed model. This is depicted in Eq. (2)

Recall = (2)

Finally, the F1 score is a measure of accuracy that can be interpreted as a weighted average of accuracy and recall. This is depicted in Eq. (3)

F1 Score = (3)

The overall accuracy of the instances which are correctly classified to the total number of the instances. This is depicted in Eq. (4)

Accuracy = (4)

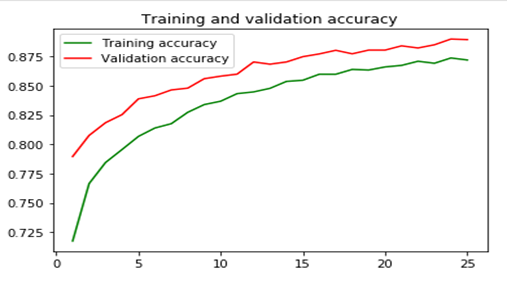
When we try to increase the precision of our model, the recall decreases, and vice versa. We estimated the F1 score because it can capture both trends in a single value. We also traced receiver operating characteristics (ROC), which are based on varying threshold values. A ROC is a probability curve. The capability of distinguishing the classes is calculated by the AUC-ROC metric. The higher the AUC, the more accurate the model. It can be represented by plotting TPR (True Positive Rate), i.e., sensitivity or recall, vs FPR (False Positive Rate) mathematically. We estimated log loss (Logistic regression loss or cross-entropy loss). It is basically defined as the probability estimate. It measures the performance of a classification model. The input is a probability value between 0 and 1.

In order to achieve the performance of proposed system we have dome come calculations using Accuracy, Precision, Recall and F1 score methods. It gives clear observation that LSTM and traditional classifier like SVM and Random Forest have nearly equal Precision and Accuracy. The table below shows the performance metrics.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Approach | Precision | Recall | F1 Score | Accuracy |
| LSTM | 0.824 | 0.847 | 0.835 | 0.918 |
| SVM | 0.812 | 0.823 | 0.796 | 0.895 |
| Random Forest | 0.841 | 0.873 | 0.855 | 0.794 |

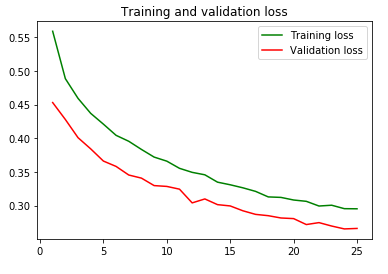
**Table 3.1:** Performance metrics comparison

We tried the for various models that includes LSTM, SVM and Random Forest that performs with the accompanying estimations: the model was run for 25 epochs with 3 stages for each stage utilizing 8 overlap cross-approval method. We accomplished exactness of 92% of accuracy. the diagram plotted on Number of Epochs versus Accuracy shows that there is no much contrast guaranteeing that the model is thoroughly preparation. As we can see there is increase in the training validation with increase in the training accuracy also. The graph is showed below.



**Fig 3.1:** Training and validation Accuracy

Further we calculated the training loss which indicates how well the model is fitting the training data, while the validation loss indicates how well the model fits new data. Fig. 4 depicts the loss and validation loss graph on number of iterations and the cost. As there is decrease in the loss, we can also see decrease in the training validation also as increase the number of epochs. The graph is showed below.



**Fig 3.2:** Training and validation Loss

This above graph shows how well is our model is fitting for the training data as there is increase in the accuracy and decrease in the loss as we trained with more data. We can also increase our accuracy by changing the weights and bias in each epoch.

1. Conclusion

In this work, a framework is proposed that have been carrying out significant work in the field of the sarcasm detection. The framework utilizes sarcastic tweets, 9,104 tweets containing #sarcasm, and #not in dataset. The proposed model in this paper presents a study of Deep Neural Networks, for the detection of Sarcasm detection in twitter. The framework utilizes the LSTM model. The methodology has shown great results and it is seen that RNN classifier has more exactness than other classifiers. It has been observed that the performance was quite impressive to determine for sarcasm sentences. From fig 3 and fig 4 we can see the training and validation accuracy curve plotted which is majorly used in classification algorithms. This proposed methodology of detection of the twitter had been given a best accuracy, precision and F1 score over the other machine learning algorithms. Thus, Deep Neural Networks will be very supportive in the analysis of the twit. the model was run for 25 epochs with 3 stages for each age utilizing 8 overlap cross-approval method. We accomplished exactness of 92% of accuracy. In future we are coming up with the model with

simple sentence, compound sentence and complex sentence with emojis using LSTM. Here in the model, we’ve achieved the accuracy.

References

1. ACM Reference format:Aditya Joshi, Pushpak Bha.acharyya, and Mark J Carman. 2017. Automatic Sarcasm Detection: A Survey.ACM Comput. Surv. 0, 0, Article 1000 ( 2017)
2. Multi-Modal Sarcasm Detection in Twitter with Hierarchical Fusion Model Yitao Cai, Huiyu Cai and Xiaojun Wan Institute of Computer Science and Technology, Peking University
3. The Role of Conversation Context for Sarcasm Detection in Online Interactions Debanjan Ghoshx Alexander Richard Fabbriy Smaranda Muresanz xSchool of Communication Information, Rutgers University, NJ, USA.
4. A Statistical Approach for Sarcasm Detection Using Twitter Data by:-Rahul Gupta, Jitendra Kumar, Harsh Agarwal and publish:- 2020, International Conference on Intelligent Computing and Control Systems.
5. Sentiment analysis for sarcasm detection on streaming short text data. by Anukarsh G Prasad; Sanjana S, Skanda M Bhat, B S Harish. In the year 2017, 2nd International Conference on Knowledge Engineering and Applications.
6. Magnets for Sarcasm: Making Sarcasm Detection Timely, Contextual and Very Personal –by Aniruddha Ghosh and Tony Veale. In the year 2018, International Conference on Intelligent Computing and Control Systems.
7. Reactive Supervision: A New Method for Collecting Sarcasm Data-by Boaz Shmueli, Lun-Wei Ku and Soumya Ray 2017. Automatic Sarcasm Detection: A Survey.ACM Comput. Surv. 0, 0, Article 1000 ( 2017)
8. A COMPREHENSIVE STUDY ON SARCASM DETECTION TECHNIQUES IN SENTIMENT ANALYSIS by Sindhu. C, G.Vadivu, Mandala Vishal Rao –In 2018 International Journal of Pure and Applied Mathematics.