Understanding Sarcasm from Reddit texts using Supervised Algorithms

Fahim Hasnat , Md. Mazidul Hasan , Abdullah Umar Nasib , Ashik Adnan , Nazifa Khanom , S M Mahsanul Islam , Md Humaion Kabir Mehedi , Shadab Iqbal , and Annajiat Alim Rasel

Department of Computer Science and Engineering
Brac University
66 Mohakhali, Dhaka - 1212, Bangladesh
{*md.fahim.hasnat, md.mazidul.hasan, abdullah.umar.nasib, ashik.adnan, nazifa.khanom, sm.mahsanul.islam, humaion.kabir.mehedi, shadab.iqbal*}*@g.bracu.ac.bd*
annajiat@gmail.com

2022 IEEE 10th Region 10 Humanitarian Technology Conference (R10-HTC) | 978-1-6654-0156-2/22/$31.00 ©2022 IEEE | DOI: 10.1109/R10-HTC54060.2022.9929882

***Abstract*—The use of satirical or ironic language for conveying a message is referred to as Sarcasm. Social networks such as Reddit, Twitter, etc. usually uses Sarcasm. Reddit which is an American website contains social aggregations of news, ratings of the content and discussions. These resources, which include links, text articles, photographs, and videos, are published to the platform by registered users and can be voted up or down. Posts that cover topics related to books, cooking, pets, news, politics, movies, religions, science, sports, fitness, video games, music, and image-sharing are organized as "communities" or "subreddits". The submissions receiving enough outvotes appear on the front page of the site and towards the top of the subreddit. This paper is about classifying a Reddit comment as sarcastic or non-sarcastic with the help of machine learning techniques. The dataset used in this study named as 'Sarcasm on Reddit' for classification according to genre-based and have followed some basic steps using supervised machine learning and deep learning algorithms for the classification of texts including pre-processing, feature extraction and modeling. In this approach, we have achieved 71%, 76% and 70% accuracy for LSTM, CNN, and Logistic Regression algorithms respectively.**

***Index Terms*—Sarcasm, Reddit, LSTM, CNN, Logistic Regression.**

I. Introduction

Sarcasm is an interesting linguistic phenomenon in which a sarcastic individual's intended meaning is contrary to the way it is communicated. Despite this contradictory nature, identifying a sarcastic phrase, mostly due to a knowledge of the context in which the phrase was uttered, the tone and body language of the sarcastic-individual, or the tendency of the individual to become sarcastic, is quite easy for those involved in the communication. However, it can be quite difficult for one to interpret the speaker's statement either as a true reflection of their intent, or as sarcastic wit without such cues to be able to derive a speaker's intent. A domain which lacks many of these cues is that of online discussions. People use positive words to communicate their bad thoughts, feelings, and opinions, making it difficult to recognize sarcasm. As a result, sarcasm has become a prevalent phenomenon on social networking sites [1].

From the community based forums to the comment sections of various media based websites, when given a textual comment by some individual, the only context available to determine the sarcastic nature of that comment is the overall post in which the comment was made [2], and perhaps in case of some platforms, the comment history of the individual. One of the largest pending natural language processing(NLP) problems of the time is developing truly conversational speech agents for understanding all the unique intricacies of human language. Humans frequently employ sarcasm in everyday discourse, whether venting, disputing, or even participating in lighthearted banter with friends. Detection of sarcasm is a must [3] for an agent for truly being conversational. Indeed, the difficulty of classifying a phrase as sarcastic or not is attributed to the contradictory nature of sarcasm.

Wallace et al. [4] claim that any classifier will face an upper bound on classifying a single phrase, without context, as the phrase can be intended to be either sarcastic or non-sarcastic. As such [2], current classification models which do not utilize features that provide sufficient insight into the context of a phrase tend to have poor classification accuracy. Sarcasm is employed to convey a meaning different from the literal one, usually in satirical context. It is complex and a bit difficult to comprehend since the actual

message in the text has to be interpreted by the user. Sarcasm [5] is utilized for humor as well as criticism of ideas, people or events. This detection is required for assessing people's true feelings, as they frequently use sarcasm to portray a reversed emotion in contrast to the true definition. Sarcasm is kind of creativity and this detection is required to reveal the creativity of people. Also, sarcasm detection and removing them from social media can reduce the toxicity level in social media. This experiment is mainly to perform sarcasm detection from text in which there is no limit for the maximum characters. The text can be a sentence, a word or even a paragraph. Hence, we have worked on the Reddit dataset. Our proposed method use a supervised approach to learn the sarcastic patterns from Reddit texts.

II. Related Works

At the time of background research, we discovered a variety of supervised machine learning and deep learning models for identifying sarcasm from Reddit texts.

P. Verma et al. [6] examined current sarcasm detection research, noting many solutions and difficulties. In this work, they analyzed the available models for detecting sarcasm.

Using Reddit and Twitter data, A. Fiallos et al. [7] suggested a system for automatically categorizing data. A labeled dataset of "42,100", Reddit publications was used to train their model. The researchers next put their trained model to the test on a dataset of tweets from, 1,573 profiles, with an average of 100 tweets per user. They had a 75.62% accuracy rate during this process.

S. Svoboda et al. [8] created a new sarcasm recognition model derived from previous sarcasm recognition research and tested it on new data from an electronic discussion forum. For this project, he used two different English data sets. He made the first one, and the author's history was downloaded from Reddit for the second. The most significant addition was the identification of the most attractive characteristics in a big and fresh data set, which must enhance the conclusions more resistant to data noise. The algorithm was slightly better at detecting non-sarcastic comments than detecting sarcastic comments, which had an accuracy of 69.5 percent.

The models that used bidirectional encoder representations (BRP) from Transformers to capture syntactic and semantic information across conversation sentences fared better than the current models, according to A. Avvaru et al. [9]. They achieved f-measures of 75.2% for Twitter and 62.1% for Reddit data-sets.

M. J. Adarsh et al. [10] attempted to bring out the negativity in positive remarks while simultaneously bringing out the positives in negative words by using Sentiwordnet to generate polarity scores. They used the term frequency-inverse document frequency (TF-IDF) approach to transform text files into vector models based on the frequency of words in the text.

S. S. Salim et al. [11] suggested a sarcastic comment identification system using the RNN-LSTM model. It is made up of 4 primary components, which are dataset, dataset for pre-processing, wordEmbedding and model RNN-LSTM. The dataset they used contains both sarcastic and non-sarcastic comments made over tweeter that they developed themselves to train the proposed algorithm. They trained their algorithm using 15,000 comments which are non-sarcastic and 11,000 tweets containing sarcasm. With additional epochs, the proposed model improved. It far outperformed standard SVM classifiers and other machine learning algorithms. They correctly identified sarcasm with 85.23% of accuracy and non-sarcasm with 86.47% of accuracy.

S. Porwal et al. [12] use tensorflow to demonstrate a deep neural network sarcasm detection algorithm from Twitter. LSTM was used to create an automatic feature extraction method. They trained their algorithm with a twitter dataset and achieved 91% accuracy.

Using keras' tokenizer framework, P. Shrikhande. et al. [13] created a model to identify sarcasm. They train their proposed model using two different datasets from Kaggle and Onion, which mostly feature humorous news headlines. They acquire an accuracy of 86.13 percent after testing their model.

III. Proposed Workflow

The primary goal of this research is to use machine learning techniques to classify a Reddit comment as sarcastic or non-sarcastic. Topic-based and genre-based are two varieties of text classification. We have used the 'Sarcasm on Reddit' dataset for genre-based classification and have followed some basic steps using supervised machine learning algorithms for text classification, including pre-processing, feature extraction and modeling.

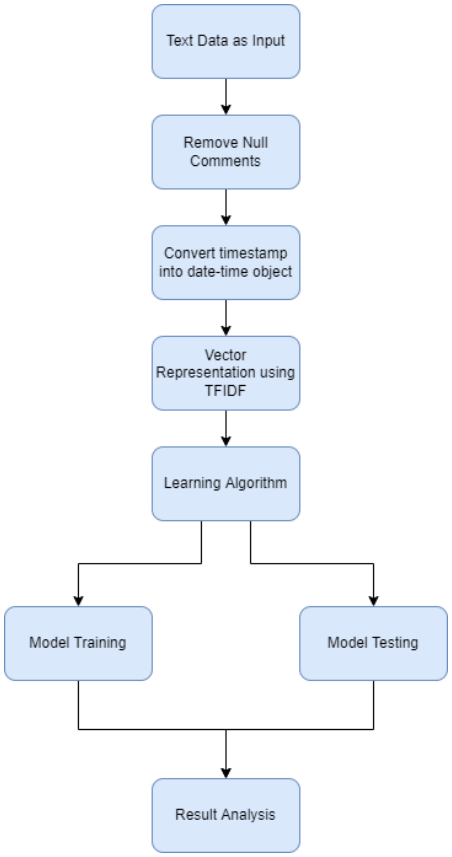

Fig. 1: Proposed workflow for understanding sarcasm from Reddit text

The procedure depicted in fig.1 demonstrates the proposed model. As part of data pre-processing, we have eliminated the containing null comments from the input text data to ensure consistency and correctness. Then timestamp objects are converted to date-time objects for further data shaping. Following that, using TF-IDF, the data is represented in vector format before going to train the model and testing with a completely blind set of data. Finally, based on the achieved experimental results, the proposed model is analyzed.

When preparing the dataset for fitting with different machine learning algorithms in the first step, it is necessary to pre-process the texts for better and more generic results. We start with text data and then filter out any null comments.

Features are extracted from the documents for the second stage. And employed the TF-IDF information retrieval technique, which considers the frequency of a phrase TF and it's inverse document frequency (IDF). The timestamp is converted to a Date-Time object, which is then vectorized using TF-IDF. This method aids in the classification of terms from high to low frequency. The dataset has now become suitable for learning to use various algorithms.

In the proposed model, we have split our data in two parts. For training the model, we used 70% of our dataset and rest of them for testing the model. We have used three machine learning algorithms named Logistic Regression, Bidirectional LSTM and CNN, and finally the last stage of our process is the analysis of the results.

IV. Dataset

In this section, the collected dataset is briefly described, also the source and authenticity of it. Moreover, the pre-processing of the dataset along with how the uniqueness among the data is identified is also demonstrated here.

A. Data Collection

In this paper, a dataset used called "Sarcasm on Reddit" prepared by Dan Ofer from Kaggle [14]. M. Khodak, et al. was the one who first acquired the information for their article "A Large Self-Annotated Corpus for Sarcasm" [15]. In this data set, there are 10,10,828 rows consisting of features including: "comment", "author", "subreddit", "score", "ups", "downs", "date", "created-utc" and "parent-comment". A label column in the dataset classifies 1 as "sarcastic" and 0 as "non-sarcastic."

Fig. 2: Data distribution in classes

B. Data Shaping

The dataset is pre-processed by first deleting the null comments and then converting the timestamp into a Date-Time object. By doing so the size of the rows decreases slightly to 10,10,773. The dataset is balanced as the proportion of the sarcastic and

non-sarcastic comments are the same, i.e. - 50% as displayed in fig. 2.

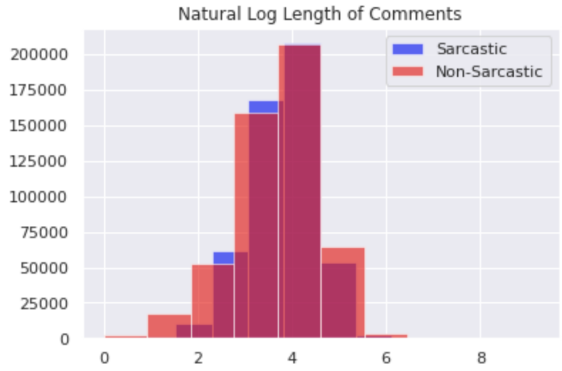

Fig. 3: Natural log length of comments

Since the dataset is skewed, log transformations are being made. According to the graph fig. 3, the length of the sarcastic comments is normally distributed, whereas the non-sarcastic comments are slightly negatively skewed.

The top 5 popular subreddits have been determined which are: “AskReddit”, “politics”, “worldnews”, “leagueoflegends” and “pcmasterrace”. Finally, in order to gain a deeper knowledge of human behavior, we have attempted to determine whether Reddit users are more sarcastic on a certain day of the week. In fig. 4, we can see the counts of sarcastic comments per day.

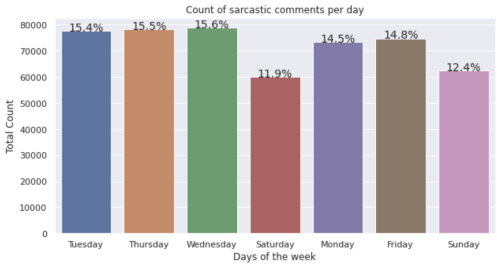

Fig. 4: Counts of sarcastic comments per day

V. Experimental Analysis

A. Feature Extraction

The process of extracting certain information that is important for a particular procedure is referred to feature extraction. Feature extraction is basically used to retrieve a subset of new features from the original feature set through the use of functional mapping while maintaining quite so much data feasible. To be specific, it encapsulated into creating variables which captured unseen business understandings and then made sound decisions about which variable is to be picked for the predictive models. The TF-IDF (Term Frequency – Inverse Document Frequency) vectorizer is chosen for this task because it considered context better than others.

TF-IDF comprises two values, one is TF and the other one is IDF. The standard means to determine TF is to take the raw frequency of a term and divide by the maximum frequency of any term in the document. IDF determines the rare commodity of a word in a given document. Scoring on the words and vectorization of the document are made with the help of a bundle of word dictionaries. This was accomplished by counting each unique word in the paper, as well as the number of times the word appeared. The more frequently a word is used, the more value it gains. Another significant feature is the ability to distinguish two papers that are nearly identical by calculating a unique value. For example, the word “not” can distinguish two papers that are nearly identical, which is recognized using TF-IDF by counting the vectorizers frequencies.

B. Evaluation

The evaluation methods which are presented here, quite well known. First, the analyzation of each result is performed and then checked the accuracy of each algorithm that we used. Then the comparison of the accuracy is done to find out which algorithm performed better here. We have calculated it with the number of correct predictions divided by the total number of prediction. In table I, the training and testing accuracy of the algorithms are demonstrated.

$$Accuracy = \frac{Number of Correct Predictions}{Total Number of Predictions} \quad (1)$$

For achieving the desired result, the dataset is tuned in such a way which can easily fit and train the data into all the algorithms. The null comments also removed and cleaned out other unnecessary features in the dataset. To calculate the accuracy of our trained and tested model, the total number of correct prediction is divided by the number of total predictions (1).

Logistic regression: Logistic regression is the baseline supervised machine learning algorithm for classification in natural language processing, and also has a very close relationship with neural networks.

Long short-term memory: LSTM [16] is a one of the type of recurrent neural networks which is better than traditional recurrent neural networks in terms of memory. LSTMs performs fairly better

because it has a good hold over memorizing certain patterns. In text classification, LSTM has a feature by which it can memorize a sequence of data.

Convolutional neural network: CNN [17] is basically an artificial neural network mostly used in analyzing visual imaginary in deep learning. CNNs are translation invariant, which means they can recognize patterns regardless of where they appear in the phrase because the local order of words is not critical in text categorization.

In the case of sentiment analysis, each filter/kernel identifies a specific feature, such as whether the phrase includes positive ("excellent", "wonderful") or negative ("bad", "awful") words. The presence or absence of a few key phrases inside a sentence determines the majority of text categorization tasks, such as sentiment analysis. CNN's, which excel at extracting local and position-invariant features from data, can successfully simulate this.

VI. Result Analysis

Fig. 5, shows the graphical representation of obtained accuracy. Moreover, the f1 scores of different algorithms are demonstrated in fig. 6. The figures show that the accuracies are the same. Here, all the shown test accuracies are plotted which were found from the test dataset.

We used three algorithms here, where LSTM and logistic regression's performance are very close. However, the most prominent is CNN's result because it extracts higher-level information using convolutional layers.

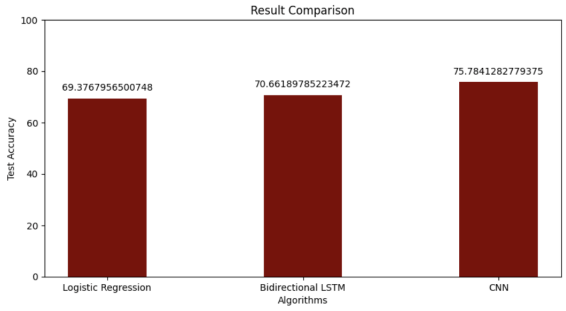

Fig. 5: Graphical representation of obtained accuracy

CNN is picked for our intent classification assignment. For this, the model is trained with 10 epochs. By calculating the accuracy, this model is evaluated. For the same dataset, the accuracy is 76% for testing the dataset. This is much better than any of the models presented here.

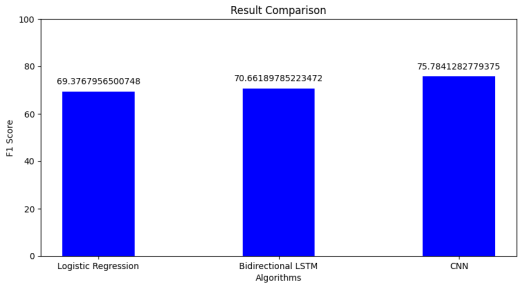

Fig. 6: F1 score comparison

The accuracy of logistic regress is lowest, and this can be the cause of linearity between the dependent and independent variables. Logistic regression cannot tackle non-linear issues because it has a linear decision surface. In real world circumstances, it is quite uncommon to have separable data. A logistic regression model is calculated by plugging numbers into the logistic regression equation, in this way calculated the result and found 69% accuracy for the test dataset and for the training dataset 72% accuracy.

In LSTM, the data set and trained the model is fed for bidirectional LSTM Algorithms. For the test dataset, the accuracy is 71%, and on the other hand, for the training dataset, the accuracy is 76%. Table I, represents the same thing.

TABLE I: Train and Test accuracy comparison

Serial	Model	Train Accuracy	Test Accuracy
01	LSTM	0.759	0.706
02	CNN	0.705	0.757
03	LR	0.722	0.693

There are two things that explain why Linear Regression is not suitable for classification. The first one is that Linear Regression deals with continuous values, whereas classification problems mandate discrete values. The second one is regarding the shift in threshold value when new data points are added. With the help of grid search, the hyperparameters are tuned and thus improve the accuracy. Moreover, performing cross validation while grid searching can increase the accuracy.

When training a network, bidirectional LSTM dropout is a regularization strategy in which input and recurrent connections to LSTM units are probabilistically omitted from activation and weight updates. In LSTM, it is very difficult to determine the dropout. Moreover, LSTMs are sensitive to different random weight initialization. In this regard, proper tuning of the model is needed to improve

the accuracy. As well as, applying hierarchical encoding of a document, e.g., one LSTM over paragraph/sentence content and another LSTM over encoded paragraphs, can increase the accuracy.

Translational invariance is an issue for CNN. Translational invariance means that if an object's orientation or location changes slightly, the neuron that is intended to recognize that item may not fire, reducing accuracy.Misclassification can be reduced with data augmentation, and a big dataset is needed to train a CNN model properly.

VII. Conclusion and Future Work

Researchers in the field of natural language processing are currently using sarcasm detection extensively. Since sarcasm detection from text data requires a common ground between the reader and the author in written form of data, in this approach, we have used genre-based classification. Pre-processing, feature extraction, and modeling using supervised machine learning algorithms are some basic procedures for text classification. Following these basic steps, the proposed model can detect whether a comment from a user on Reddit is a sarcasm or not. Among our proposed model, we have obtained the best accuracy using CNN.

Since, the proposed model have worked on a text dataset limited to the user comments on Reddit, scope of improvement was also restricted to the received content form comments. However, with the availability of more data, it can work on behavioral modeling frameworks, semi-supervised pattern extraction to extract implicit sentiment and hashtag-based supervision. We are also working on comparison between these approaches to find out the most comprehensive techniques among them. The future approaches might not only stay limited to determine sarcasm, but also extract other emotions and psychological behaviors.

References

[1] R. Jamil, I. Ashraf, F. Rustam, E. Saad, A. Mehmood, and G. S. Choi, "Detecting sarcasm in multi-domain datasets using convolutional neural networks and long short term memory network model," *PeerJ Computer Science*, vol. 7, p. e645, 08 2021.

[2] K. Clarkson and M. Reza, "Was that sarcasm? -a survey of machine learning models for classifying sarcastic comments on reddit using word embeddings," 12 2018.

[3] R. S. Nick Guo, "Finding sarcasm in reddit postings: A deep learning approach."

[4] B. C. Wallace, D. K. Choe, L. Kertz, and E. Charniak, "Humans require context to infer ironic intent (so computers probably do, too)," pp. 512–516, Jun. 2014. [Online]. Available: https://aclanthology.org/P14-2084

[5] A. A. B. Lakshmanan, "Sarcasm detection in reddit," *International Journal of Recent Engineering Research and Development (IJRERD)*, vol. 3, pp. 46–54, 2 2018.

[6] P. Verma, N. Shukla, and A. Shukla, "Techniques of sarcasm detection: A review," pp. 968–972, 2021.

[7] A. Fiallos and K. Jimenes, "Using reddit data for multi-label text classification of twitter users interests," pp. 324–327, 2019.

[8] S. Svoboda, "Sarcasm detection in reddit comments," *Amsterdam School of Economics*, 8 2018.

[9] A. Avvaru, S. Vobilisetty, and R. Mamidi, "Detecting Sarcasm in Conversation Context Using Transformer-Based Models," pp. 98–103, Jul. 2020. [Online]. Available: https://aclanthology.org/2020.figlang-1.15

[10] A. M. J and P. Ravikumar, "Sarcasm detection in text data to bring out genuine sentiments for sentimental analysis," pp. 94–98, 2019.

[11] S. S. Salim, A. Nidhi Ghanshyam, D. M. Ashok, D. Burhanuddin Mazahir, and B. S. Thakare, "Deep lstm-rnn with word embedding for sarcasm detection on twitter," pp. 1–4, 2020.

[12] S. Porwal, G. Ostwal, A. Phadtare, M. Pandey, and M. V. Marathe, "Sarcasm detection using recurrent neural network," pp. 746–748, 2018.

[13] P. Shrikhande, V. Setty, and D. A. Sahani, "Sarcasm detection in newspaper headlines," pp. 483–487, 2020.

[14] D. Ofer, "Sarcasm on reddit," May 2018. [Online]. Available: https://www.kaggle.com/datasets/danofer/sarcasm?group=bookmarked

[15] M. Khodak, N. Saunshi, and K. Vodrahalli, "A large self-annotated corpus for sarcasm," May 2018. [Online]. Available: https://aclanthology.org/L18-1102

[16] K. M. Hasib, N. A. Towhid, and M. G. R. Alam, "Online review based sentiment classification on bangladesh airline service using supervised learning," in *2021 5th International Conference on Electrical Engineering and Information Communication Technology (ICEEICT)*, 2021, pp. 1–6.

[17] Q. A. R. Adib, M. H. K. Mehedi, M. S. Sakib, K. K. Patwary, M. S. Hossain, and A. A. Rasel, "A deep hybrid learning approach to detect bangla fake news," in *2021 5th International Symposium on Multidisciplinary Studies and Innovative Technologies (ISMSIT)*, 2021, pp. 442–447.