2020 International Conference for Emerging Technology (INCET)
Belgaum, India. Jun 5-7, 2020

Deep LSTM-RNN with Word Embedding for Sarcasm Detection on Twitter

Sayed Saniya Salim[1], Agrawal Nidhi Ghanshyam[2], Darkunde Mayur Ashok[3], Dungarpur Burhanuddin Mazahir[4], Bhushan S.Thakare[5]
[1,2,3,4,5] Computer Engineering, Sinhgad Academy of Engineering, Pune, India
[1]saniyasayed514@gmail.com, [2]agrawal.nidhi6@gmail.com, [3]mayur08darkunde@gmail.com, [4]burhandungarpur10@gmail.com,
[5]bhushan.thakare86@gmail.com

***Abstract*— In the world full of sarcastic people, it's becoming challenging for the people of 21st century to detect sarcasm using sentiment analysis efficiently. Sarcasm detection helps us to understand the bitter truth under the sugar coated sentences. It is widely used in various networking sites for understanding the true reviews and taking appropriate actions on the same if needed. Various methods, techniques and algorithms have been applied, although there's little or much drawback of using the same. For instance, algorithms like logistic regression has been used to detect sarcasm, which has a drawback, can't be used for continuous datasets. In our paper, we will be discussing about the approach we found appropriate and also provides an increased accuracy. Self designed dataset of sarcastic and non-sarcastic tweets is used for training of proposed model. Use of Recurrent Neural Network, Long Short Term Memory (LSTM) and Word Embeddings can make the sarcasm detection efficient and thereby making the statements from twitter easily classifiable.**

Keywords*— *Sarcasm; LSTM; RNN; Embedding; Tweets; Natural Language Processing;

I. INTRODUCTION

The word "Sarcasm" was first used in 1579, in a reference to The Sheperdess Calendar by Edmund Spenser. However, it is seen, that Sarcasm is often looked at as an insult. It's a way to address or compliment someone negatively in a polite way. For instance, "Sam is faithful as a Dog", here, Faithful is a good attribute, but Sam's comparison to a Dog is an insult. Sarcasm is at its peak in today's Virtual world, social media platforms, and where not. Therefore detecting sarcasm is one of the important task that needs to be addressed, so that we can know the real sentiment of the person. The use of exclamation marks, emoticons, capital letters make Sarcasm more prevalent.

Twitter is widely used in the sarcasm detection research as the tweets are not private and are available publicly. In this work, we have proposed a fusion model which includes Recurrent Neural Networks, Word Embedding and Long Short Term Memory. Initially, we use word embedding to map the words to vectors and then use the attribute features to initialize the LSTM cells, studies of different approach explains Neural network works significantly better than the other approaches used so far to detect sarcasm detection[1,2,3,4,5]. This paper is divided in the following way: the introduction to sarcasm detection task has been presented in chapter I, followed by chapter II, which discusses about our motivation, followed by the related work in chapter III, the proposed approach has been discussed in chapter IV, while chapter V, VI, VII, VIII discuss the experimental results, result analysis, the future scope and conclusion respectively.

II. MOTIVATION

Machine Learning is transcending various boundaries. Especially, neural network's cognitive ability and statistical analysis made its application possible in variety of disciplines. Our main motivation was to make sarcasm detection easier for the companies so that they can provide their customers with the real time customer assistance. It is one of the tedious task in sentiment analysis, yet the most prominent task. On various shopping websites like Flipkart, EBay, Amazon, sarcasm detection helps to understand the product review by the customer and the satisfaction. This can further be analyzed for the better market behavior to enhance the customer's experience.

III. RELATED WORK

A. Sarcasm Detection

Sarcasm detection is the necessity of the digital world, Growing network of the interconnected social media entities has lead in the surge of big data. Natural Language processing is escalating branches of machine learning, that in return cause many researches to get involve in sentiment analysis. In last few years many researchers have done work on twitter dataset for classification and opinions mining nevertheless various techniques are used in this area for betterment. Lexical features are studied by Kreuz and Caucci [6] for sarcasm detection, their studies shows punctuations and stop words are effective in detecting sarcasm. Kreuz and Caucci [6] and Carvalho et al. [7] preferred unigram lexical feature extraction for sarcasm detection. Further, Lukin and Walker [8] put one step forward and used n-gram feature extraction and pattern based method. Bharti et al. [9] used various algorithm such as Parsing based lexical generation algorithm (PBLGA), Likes and Dislikes Contradiction (LDC), Tweet contradict universal facts (TCUF),Tweet contradict time dependent facts(TCTDF) to extract features such as lexicons, universal facts, hyperbole also classification is performed by using machine learning methods where PBLGA with decision tree approach gives higher performance and recall of 0.76.

Bouzizi [10] proposed a pattern based approach to detect sarcasm in twitter, Here 4 sets are proposed based on feature which is further used for classification of tweets. Basak et al. [11] automated the task of classification public shaming tweets into different classes as abusive, shaming tweets etc. Here SVM classifier is mainly used as classifying approach and block shame web application designed for blocking of attackers.

978-1-7281-6221-8/20/$31.00 ©2020 IEEE

B. Neural Network

The starting researches in field of sentiment analysis were frequently in Feature extraction, Pattern recognition and classification based on extracted features. In recent studies it is observed that the combination of method provides better performance in sarcasm detection. A recent study of hazarika et al. [12] came up with a new approach known as CASCADE that is Contex-tuAlSarCasmDEtector; it's a hybrid way of both content and driven techniques. Basically content driven techniques task is to classify and finding the lexicons, pragmatic features to detect sarcasm. In Last 10 year's contextual sarcasm is widely used in social media websites and platforms. Various statements, discussion, chats, reviews are mostly afflict by slangs, regional language, grammar mistakes and contextual information. This kind of information is difficult and inefficient to extract and thus more efficient technique and extraction clues are required.

Davidov et al. [13] used a semi supervised approach with two different datasets of twitter and Amazon. Further these data sets are pre-processed by feature selection, pattern extraction, and pattern selection, pattern matching and annotated the statements accordingly. KNN cluster classification algorithm & SASI algorithm is used for testing datasets which provides 91.2% precision with F score 0.82. Performance of both datasets are compared and examined on different parameters. Different architecture are proposed for sarcasm detection using neural network, deep CNN model of dos Santos and Gatti [14]and neural CRF model of Zhangetal. [15] are one of good architecture proposed using Neural network. Power of neural network is clearly visible by comparing results of existing models with neural network model.

IV. Proposed Approach

The proposed approach of sarcasm detection is RNN-LSTM model , It has mainly four main elements , That are 1)Dataset 2)Pre-Processing Dataset 3)Word Embedding 4)RNN-LSTM model. The architecture and main elements are explained further. Two separate datasets were used to train the model, one for training and other for testing. Initially we used the training dataset .The training dataset was pre-processed. The pre-processing involved the removal of newlines, punctuations, white spaces, special symbols, redirection to links and any such which has no significance in determining the sarcasm of the sentence, if any. Once the training dataset is pre-processed, a new pre-processed dataset is obtained. This pre-processed dataset is used in further steps. Word Embedding is applied on the pre-processed dataset. Label and integer encoding is done, followed by the training of Neural Network, Preciously RNN in this project , and finally word to vector creation is done. This word to vector creation is further stored into a vector matrix. The vector matrix is used by the LSTM, where a separate model is created to store the weights of the LSTM model. Lastly, the classification is done. Then the testing dataset is applied to determine the accuracy for the sentences.

A. Dataset

To train dataset appropriately for proposed model, dataset is created by mining various sarcastic tweets from public accounts on twitter. Hash tags and various sarcastic profile pages on twitter are visited, and manually context driven sarcastic tweets are mined. Also non sarcastic tweets, Facts, True opinion, conversational tweets and re-tweets are included in dataset. Similarly testing dataset is created.

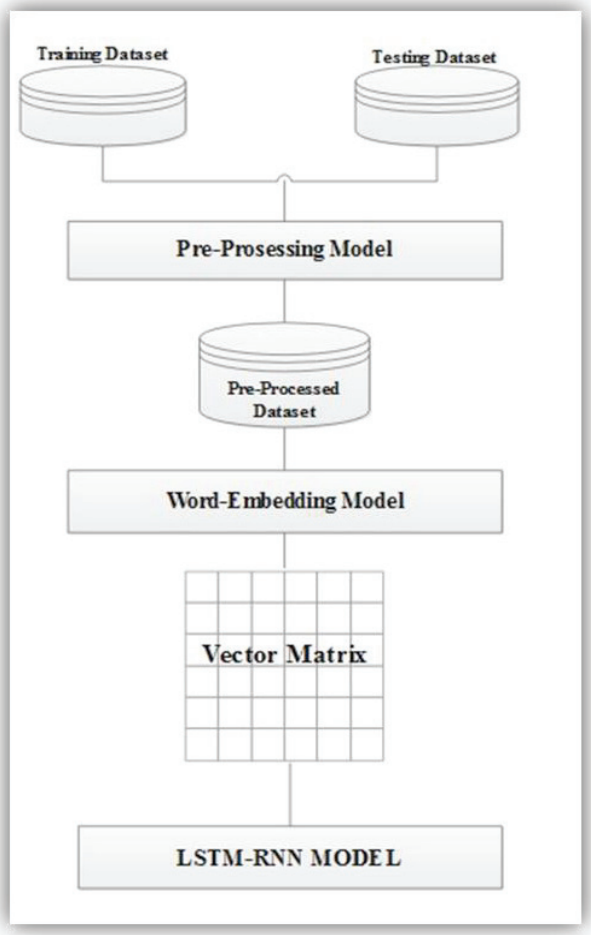

Fig. 1. Architecture of the proposed system

B. Pre-Processing Model

Before feeding the mined tweets to algorithm, Data need to be transformed to clean and required format. For example before processing tweet from dataset is like this: ['former Versace store clerk sues over secret black code for minority shoppers', 'gourmet gifts for the foodie 2014']. These tweets from dataset are fed to pre-processing for feature extraction also to convert raw tweets into cleaned one. Various unnecessary elements are removed from tweets. Tokenization and Padding sequence are the two components mainly used for pre-processing.

C. Pre-Processed Dataset

The text is converted into sequence and these processed tweets are further used for word embedding model. Transformed tweets are padded with zeros and each tweet is made of equal length.

D. Word Embedding

Neural network is capable of processing numerical data rather than textual data and thus word embedding plays important role for converting text into numerical form. Processing huge amount of text and analyzing is tedious task and word embedding handles it well. In word embedding

words are mapped to vectors of real number. We have used embedding size of 128. Generated vectors are further feed to neural network layers.

Word embedding also serves for grammatical correlation between words, which means possibility of two words coming together. Model leans and predict next word according to count of two word coming together. For example Possibility of 'Sharp' followed by 'Knife' or 'Mind' is high rather than any other word.

E. RNN-LSTM Module

The RNN-LSTM module is capable of moving in both directions forward and backward. In forward direction information is passed to next node for getting familiar with the previous node. For example 'I am good at mathematics and thus I am perusing masters in math'. Here context of first node of good at math can be pass to further node for understanding possibility of 'math' related word occurrence.

Understanding context is necessary in sarcasm detection also extracting feature is one of crucial task, neural network is capable of doing this task better. Neural network can implicitly extract features also as it works on activation function its intensity increases with every node.

Training on dataset with full of feature rich data is first step toward building strong model and using this features in best way is another big step. Using LSTM in model is beneficial as LSTM cell are capable of remembering and forgetting information. Fig 2 shows the diagrammatic representation of RNN and LSTM nodes. RNN take input as (X_{t-1}) state and produces (h_{t-1}) state using 'tanh' activation function, further information is passed to next node. LSTM has gates similar as gates in electronics, It stores information and when this gates are arranged in particular manner it work as counter. LSTM process information in similar way it stores needful information and forgets the rest.

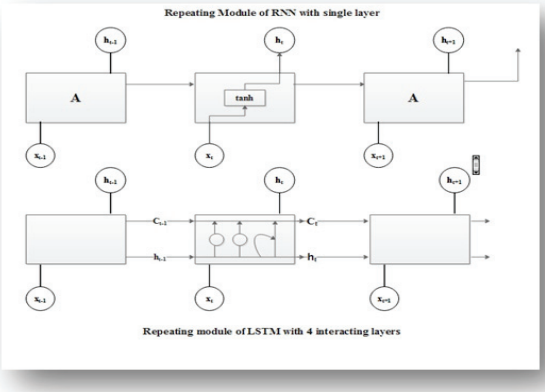

Fig. 2. RNN-LSTM model

The model is implemented using Tensorflow and Keras libraries. Our model consist of two embedding layers, one spatial dropout layer and LSTM layers. As neural network's layers affect its accuracy, we have selected moderate no of layers to handle accuracy as well as computation. This model is used with a front end as web page. This Web page is build using flask framework and model is deployed as web application for its demonstration.

V. Experimental Results

We have trained our model with 15000 Non-sarcastic and 11000 sarcastic tweets. Although the hardware constraints the model behaved well and trained well. Model had achieved 88 percent of accuracy only with 15 epochs.

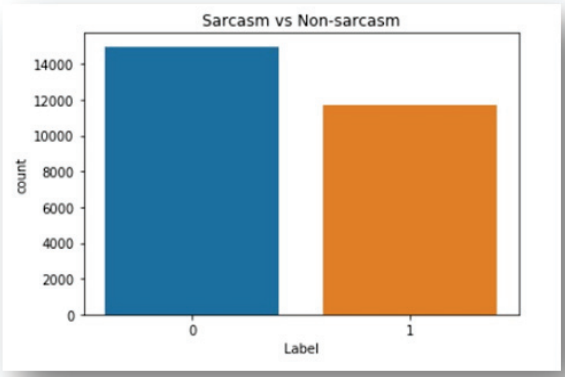

Fig. 3. Sarcasm vs Non-Sarcasm

Loss of model gradually decreases with increased in no of epochs

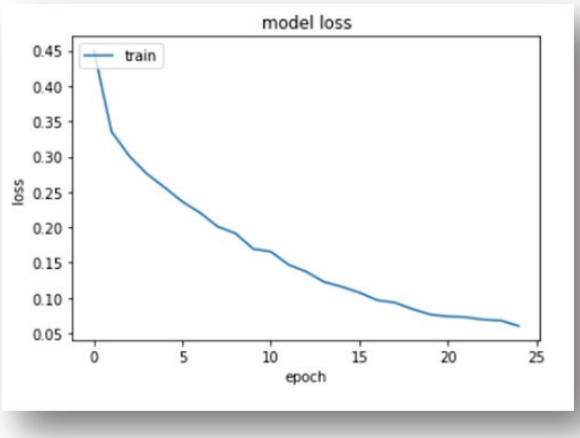

Fig. 4. Model Loss

On testing dataset of 15000 tweets,

Positive Count: 650

Negative Count: 850

Positive Correct: 554

Negative Correct: 735

Sarcasm accuracy 85.23 %

Non-Sarcasm accuracy 86.47 %

With accuracy above 80 percent and less no of epoch it is efficient and clearly better model then other algorithmic model for sarcasm detection.

VI. Result Analysis

The proposed model gets better with more number of epochs, It is certainly better than plain SVM classifier or other machine learning algorithms.

VII. Future Scope

One of the future scopes which is not much looked on is datasets. More work on dataset can be done, as currently people are using most of the textual data from social

networking sites such as Twitter and Reddit. Bigger and diverse and varied datasets can be used instead of the ever growing DL models which are complex. A good data creates a good model, and till date, the datasets have their own limitations and biases. Apart from speaking about the dataset, Sarcasm detection plays an important role, be it implemented in the chatbot or auto replies.

In the days ahead, RNN along with LSTM can be applied for automated clustering of sentiment types. It can also be applied into the dependency rules of the clustering. Using this, it can be further expanded to detect various non-literal forms of sentiments like irony and satire.

VIII. Conclusion

Sarcasm detection, being one of the fascinating subjects, evaluates a sentence for features like patterns, words and then later classifies the sentences as "Sarcastic or Non-Sarcastic" using the classifier also termed as classification framework. Although CNN provides a quick way to classify, but RNN and LSTM are common for textual usage.

Speaking about the dataset, to train our model we have used 26000 tweets dataset and using NLP the same was preprocessed. We have first created an abstract from the list of strings into the lists of integers. Further then, labels and features were created from the sequences. Then after, LSTM model was built with word embedding, LSTM and Dense layers. We loaded the pre-trained embedding and then trained our model to predict the next work in sequence.

We streamed the data (tweets) and the model gave the prediction of whether the tweet is sarcastic or not. The test dataset, of 1500 tweets was also used to predict the efficiency of the model.

References

[1] O. Tsur, D. Davidov, and A. Rappoport.Icwsm-a great catchy name: Semi-supervised recognition of sarcastic sentences in online product reviews. In ICWSM, 2010.

[2] R. Gonza´lez-Iba´n˜ez, S. Muresan, and N. Wacholder.Identifying sarcasm in twitter: A closer look. In ACL (Short Papers), pages 581–586. Citeseer, 2011.

[3] C. Liebrecht, F. Kunneman, and A. van den Bosch.The perfect solution for detecting sarcasm in tweets# not.WASSA 2013, page 29, 2013.

[4] E. RiloR, A. Qadir, P. Surve, and Silva.Sarcasm as contrast between a positive sentiment and negative situation. In EMNLP, pages 704–714. ACL, 2013.

[5] P. Rockwell. The eRects of cognitive complexity and communication apprehension on the expression and recognition of sarcasm. Hauppauge, NY: Nova Science Publishers, 2007.

[6] Roger J Kreuz and Gina M Caucci. 2007. Lexical influences on the perception of sarcasm. In Proceedings of the Workshop on computational approaches to Figurative Language, pages 1–4.

[7] Paula Carvalho, Lu´ısSarmento, M´ario J Silva, and Eug´enio De Oliveira. 2009. Clues for detecting irony in user-generated contents: oh...!! it's so easy;-). In Proceedings of the 1st international CIKM workshop on Topic-sentiment analysis for mass opinion, pages 53–56.

[8] Stephanie Lukin and Marilyn Walker. 2013. Really? well. apparently bootstrapping improves the performance of sarcasm and nastiness classifiers for online dialogue. In Proceedings of the Workshop on Language Analysis in Social Media, pages 30–40.

[9] S.K. Bharti , R. Pradhan , K.S. Babu ,S.K. Jena," Trends in Social Network Analysis", , Lecture Notes in Social Networks,2017,pp 51-76

[10] M. Bouazizive T. Ohtsuki, "Sarcasm Detection in Twitter: "All Your Products Are Incredibly Amazing!!!" - Are They Really?," in GLOBECOM 2015 - 2015 IEEE Global Communications Conference, San Diego, 2015

[11] Rajesh Basak, ShamikSural ,NiloyGanguly, veSoumya K. Ghosh, "Online Public Shaming on Twitter: Detection, Analysis, and Mitigation",2019

[12] DevamanyuHazarika,SoujanyaPoria. Proceedings of the 27th International Conference on Computational Linguistics, pages 1837–1848Santa Fe, New Mexico, USA, August 20-26, 2018.

[13] D. Davidov , O. Tsurve A. Rappoport, "Semi-supervised recognition of sarcastic sentences in Twitter and Amazon," in Association for Computational Linguistics, Stroudsburg, 2010.

[14] Cicero dos Santos and MairaGatti. 2014. Deep convolutional neural networks for sentiment analysis of short texts. In Proceedings of COLING 2014, pages 69–78.

[15] Meishan Zhang, Yue Zhang, and Duy Tin Vo. 2015.Neural networks for open domain targeted sentiment. In Proceedings of the EMNLP, pages 612–621.