SARCASM DETECTION BEYOND USING LEXICAL FEATURES

ADEWUYI, Joseph Oluwaseyi[1] and OLADEJI, Ifeoluwa David[2]

Department of Computer Science, University of Ibadan, Ibadan, Nigeria.

ABSTRACT

In current time, social media plateforms such as facebook, twitter, and so forth have improved and received substantial importance. These websites have grown into huge environments wherein users explicit their thoughts, perspectives and reviews evidently. Organizations leverage this environment to tap into people's opinion on their services and to make a quick feedback. This research seeks to keep away from using grammatical words as the only features for sarcasm detection however also the contextual features, which are theories explaining when, how and why sarcasm is expressed. A deep neural network architecture model was employed to carry out this task, which is a bidirectional long short-term memory with conditional random fields (Bi-LSTM-CRF), two stages were employed to classify if a reply or comment to a tweet is sarcastic or not-sarcastic. The performance of the models was evaluated using the following metrics: Accuracy, Precision, Recall, F-measure.

KEYWORDS

Sarcasm Detection, Deep Learning, Contextual features

1. INTRODUCTION

In contemporary time, social media platform such as Facebook, LinkedIn, Twitter, etc. have expanded and received vast admiration and significance. These sites have grown to be massive environments where users specify their ideas, views and opinions naturally. Social media sites became a well-established platform for users to specify their feelings and opinions on various topics, like events, individuals or products. Social media channels became a preferred platform to debate ideas and to interact with people worldwide. For instance, Facebook claims that it has 2.45 billion active users per month as of the third quarter of 2019, each one being a friend with 130 people on average. Similarly, Twitter claims to boast of 330 million monthly active users (as of 2019 Q1). For those, more than 40%, or more exactly 134 million, use the service on each day. Users post quite 340 million tweets and 1.6 billion search queries a day [1].

These days, social media networks are habitually the first place to get the reaction about contemporary occurrences and trends from user centre, allowing them to provide companies with vital data that can be used to position their products in the market as well as gather quick reaction from customers. When an occasion commences or an item is propelled, individuals begin tweeting, composing surveys, posting comments, etc. on social media. People go on social media sites to read reviews of a few items by other users before they make a decision whether or not to buy the merchandise. Organizations, groups, bodies also depend on these social media sites to know the response of users for their services and successively use the feedback to enhance their services.

DOI: 10.5121/ijdms.2020.12304

Companies and organizations leverage on this unique environment to tap into people's opinion on their products or services and to make available instantaneous customer assistance. Not shockingly, most big companies and firms have a social media nearness and a committed group that their service to the companies or the firms is just for promoting, after-sales services or feedback services, and customer help through social media [2]. By the means of vast speed and large quantity of social media information, organizations and companies got to perform diverse assignments like substance administration, estimation investigation, and extraction of important messages for the service representatives to reply to.

In any case, finding and confirming the authenticity of conclusions or surveys could be a formidable task, to research these various opinions may be a big task since there are some subtle difference forms of language such as sarcasm in which the meaning of a message is not always understandable and clear. This forces an additional burden on the social media team and text miner to recognize these messages and take action appropriately. It is hard to physically peruse through all the reviews and opinion of individual in order to determine which of the opinions communicated are sarcastic or which one is not. Additionally, it will be difficult for the common reader to recognize sarcasm in tweets or product reviews, which may end up misleading.

Encarta dictionary defines sarcasm as comments meaning the opposite of what they appear to mention, and intended to mock or deride. [3]. Another definition for sarcasm by Macmillan is that sarcasm the activity of claiming or writing the alternative of what you mean, or of speaking in an exceedingly way intended to make some other person feel stupid or show them that you are simply angry [4]. Sarcasm is a style of communication in which precise and intentional connotation are in contradictory. Sarcasm is commonly utilized to communicate a negative message utilizing positive words. In Natural Language Processing (NLP), such as sentiment analysis, opinion mining etc. automated detection of sarcasm is then very relevant, because sarcastic expression that contains positive terms can conveys a negative meaning and can be easily misinterpreted and misclassified by an automatic machine that performs this processing of the natural language.. Sarcasm also occurs when an individual implies something else from what he or she is talking about.

Sarcasm is a complex frame of discourse act in which the speakers indirectly communicate their message.. One essential characteristic of the sarcastic speech act is that it is sometimes hard to be aware of it [5]. The difficulty of sarcasm detection creates ambiguity and misinterpretation in day-to-day communication and causes problems for many NLP systems, such as online survey summarization frameworks, discourse frameworks or brand observing frameworks due to the inability of state-of-the-art sentiment analysis systems to identify comments that are sarcastic. For example-"I satisfy being cheated". Here "satisfy" Is expressing a positive feeling in a negative context. Definitely this post is indicated and suggested as sarcastic.

Much research work on automatic detection of sarcasm has mainly been on Twitter Data and has mainly concentrated on finding information from the text of the social media post. Those models and methods handle sarcasm as a linguistic or grammatical phenomenon, without or with limited emphasis on the psychological features and other property of sarcasm. However, sarcasm has been studied to a great extent in psychological and behavioral sciences and theories explaining when, why, and how sarcasm is expressed [2]. These theories can be generalized and used for the automated identification of sarcasms on social media posts..

This research seeks to avoid the use of grammatical words as the only features for sarcasm detection but also the contextual features which theories are explaining when, how and why sarcasm is expressed. These contextual features consider the user's current and previous posts to detect or classify if a post is sarcastic. For example, a tweet written by a company about the specification of their new product and the services of the product. After some minutes of the post by the company, one of the user put a comment under the post that says "Wao! I Love this product, it is one of the best product have ever seen". Minutes after the comment, the company replied to the comment of the user, saying "Thank you customer, we are the best because we offer the best services". From that, different user begin to review the product, based on the first comment of the user but after few minutes again, the first user wrote a comment again under the response of the company reply and the user said " Do I mean LOVE? Your product that got spoilt some days after I purchased it" with the last comment/reply of the first user, it is clear that his first comment to the company post is a sarcastic comment, that can be misclassified because of the presence of the positive words (LOVE and BEST) and this will definitely affect other NLP work on the company analysis of the post. Relying only on the previous post of the first user alone and not connecting it to the current post will affect the classification.

This research used both linguistic and contextual features to detect sarcastic post in social media platform, this work designed a model aimed to detect sarcasm without the use of words and patterns of words alone.

2. LITERATURE REVIEW

The correctness of detection of sarcasms counts on every aspect of language; from the semantic to the lexical. Sarcasm detection requires several parameters in order to be successful; Lexical, Pragmatic and Hyperbole are examples of features that are often used. According to Saha et. al.[6], they said sarcasm detection is divided into three categories on the basis of text features that are being used for classification. The categories involve Lexical, pragmatic and hyperbolic feature based classification. Lexical feature based classification Includes text properties such as unigram, bigram, and n-grams. Pragmatic feature based classification refers to symbolic and figurative text. Examples-emotions, smilies etc. Hyperbole feature based classification involves text properties such as intensifiers, interjections, punctuation mark, quotes etc.

Classification based on the lexical feature involves text properties like unigram, bigram, and n-grams. An n-gram, is a connected sequence of n items from a simple sample of text. The items can be phonemes, syllables, letters, etc. according to the application. An n-gram of size 1 is called a unigram; size 2 is referred to as a bigram etc.

Bindra et. al. [7] Used bigrams and unigrams grouping a single word (e.g., extreme, fantastic, excellent, etc.) and double words (e.g.: very poor, really awesome, really nice, etc.). To extract them from the remaining text and each tweet was passed through tokenization, stemming, uncapitalization and by doing so, each and every n-gram was added to a binary feature dictionary. They investigated the applicability of pragmatic and lexical features in machine learning by classifying different positive, negative and sarcastic Tweets. The two standard classifiers that they used in sentiment classification are: logistic regression (LogR) and sequential minimal optimization (SMO) supporting vector machine.

Also, Peng et. al. [8] uses N-grams, such that specific tokens i.e. unigrams and bigrams are appended into a binary feature dictionary. Bigrams are obtained using the same library and are defined as duo of words that typically go together, examples include artificial intelligence, peanut butter, etc. they created the term frequency-inverse document frequency (TF-IDF) matrix, and then fed it into the Naive Bayes multinomial classifier. Likewise, Barbieri et. al. [9] does not include patterns of words as features for detecting sarcasm but made use of seven collections of lexical characteristics aimed at detecting sarcasm through its internal structure.

Pragmatic classification of features includes figurative and symbolic texts, such as smilies, emoticons. A range of authors have used pragmatic features to detect sarcasm. González-Ibáñez et. al. [10] used both lexical and pragmatic features, for the pragmatic features, they Used three pragmatic features, namely: I positive emoticons such as smileys; (ii) negative emoticons such as frowning faces; and (iii) User marking a response to another tweet (signaled tweets by <@user>). Likewise, Joshi et. al. [11] posed a computational approach that the root for sarcasm detection is in harnesses context incongruity. This work shows that a sarcasm detection system that is grounded in a linguistic theory, which is the theory of context incongruity. They define Incongruity as 'the state of being not in conformity, as with theories or principles'. Precisely, they used four kinds of features: (A) Lexical, (b) Pragmatic, (c) Implicit congruence, and (d) Explicit incoherence. Lexical features are unigrams acquired using feature selection techniques such as $\chi 2$ Test and Categorical Proportional Difference. Pragmatic features include expressions of laughter, emoticons, capital words and punctuation marks.

Hyperbole features based classification has been utilized for many work in identifying sarcasm in the text. It is the combination of the properties of text such as quotes, punctuations, intensifier, and interjection etc.

Clews [12] utilized string matching against positive sentiment and interjection lexicons to investigate whether or not the presence of both can be used to classify content as sarcastic or non-sarcastic.. Also, Bharti et. al. [1] proposed a two approaches for detection of sarcasm in the text data. The first is the use of a lexicon generation algorithm (PBLGA) based on parsing, and the second was the detection of sarcasm based on the frequency of the word interjection.. Also the method use for this two approaches is in two parts, they are 1. Part-of-speech (POS) Tagging and 2. Parsing and Parse Tree, parsing is the method of analyzing the grammatical structure of a language.

Bouazizi et. al [5] In their work, They suggested a method for detecting sarcasm using a pattern-based approach to Twitter data by considering various forms of sarcasm. The approach used for their work was to propose four sets of features which are (sentiment-related features, punctuation-related features, syntactic and semantic features and pattern-related features) that cover the different types of sarcasm which are (sarcasm as wit, sarcasm as whimper and sarcasm as avoidance). They divided words into two clusters for the pattern-based feature: the first cluster, called "CI," which contains words of which the content is important, and the second cluster named "GFI," which contains words of which the grammatical function is more important.

Felbo et. al. [13] presented a probabilistic modelling framework to identify, classify and learn sarcastic text characteristics through the training of a human-informed sarcastic benchmark using neural network. Their approach used the Parts-of-speech (POS) tagging concepts to identify specific words belonging to the defined constraints categorizations that are collectively stacked and aggregated as values to be fed into a two-layer multi-perceptron network to properly classify the text as sarcastic or not.. The feature categorization of their model are: keyword features, punctuation features, superlative features, preferentiality features, seasonal features.

According to Jyoti [14], the method he employed to capture sarcasm is that a model was built and tested using self-description of the user to obtain additional information about personality nature or character of Twitter authors. The features used were divided into three categories L-LEXICAL (features used includes: N-grams, Intensifiers, Capital Letters, Word-Count, Double-Quotes, Part-of-Speech Tags), S–SENTIMENT ({Sentiment Score and Contrast in Sentiments are used for the extraction), and T-TOPIC_MODELING, which uses Topic- Modeling with the aid of Latent Dirichlet Allocation (LDA) based features.

Jyoti [15] in his work, Proposed a model that explores those forms of Long Short-Term Memory (LSTM) networks capable of modeling both the discussion environment and the sarcastic response, using conditional LSTM networks and LSTM networks with a focus on context and outer response creating the LSTM model that reads only the response, they also developed a model that carefully analyzes the attention weights created by the LSTM models and discusses the results as compared to human performance on the task.

In the mid-90s, the German scientists SeppHochreiter and JuergenSchmidhuber proposed a variety of repetitive net with supposed Long Short-Term Memory Units, or LSTMs, as a response to the disappearing inclination problem. [16]. LSTMs help protects the blunder that can be backpropagated through time and layers. By keeping up an increasingly consistent mistake, they enable repetitive nets to keep on learning over many time ventures (more than 1000), in this way opening a channel to connect circumstances and end results remotely [17]. In LSTM, the hidden state of increasing position (ht) only encrypts the prefix context in a forward direction while not considering the backward context [17]. Two parallel moves (forwards and backwards) and the two LSTM's concatenated hidden states were used by Bidirectional LSTM to represent each position [18].

The bidirectional LSTM (BiLSTM) architecture is used to capture both past and future information by concatenating hidden state $\rightarrow h_t$ of forward LSTM and $\leftarrow h_t$ of backward LSTM. So you could define BiLSTM's hidden state as:

$$h_t = \overrightarrow{ht} + \overleftarrow{ht} \qquad (1)$$

Recently, Ghaeini et. al.[18], Proposed model called "Dependent Reading Bidirectional LSTM Network(DR-BiLSTM)" for modelling the joining between a premise and a hypothesis during encoding and inference, they also present an ensemble approach to syndicating models.

2.1. Conditional Random Field (CRF)

CRFs are a sort of discriminative probabilistic model, they are used to encrypt known associations among perceptions and create unsurprising clarifications and are regularly used for labelling or parsing of sequential data. Many work have been done using conditional random field. Zavalaet. al [19]used a new LSTM+CRF approach to remove adverse drug reactions from user reviews, a concatenating recurrent neural network and a CRF operating on scores extracted from this neural network.

Also, Paper et. al. [20] in their paper titled “Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition” proposeda model that established the fact that rudimentary Bi-LSTM model is not enough to beat the existing state of the art of NER solutions, they concluded that addition of CRF layer to the Bi-LSTM model drastically increases it’s quality.

Zhouet. al. [21] Suggested cross-sectional Bi-LSTM and CRF models The model has two bidirectional Long Short-Term Memory (LSTM) layers and a last Conditional Random Field (CRF) layer. They claimed that their model adds embedding of sense-disambiguation and an extended format of tag encoding to detect discontinuous entities, as well as overlapping or nested entities.

3. METHODOLOGY

This Section of the work discussed the generic framework and overall discussion of its components, the detailed discussions of each component part of the proposed framework and any necessary algorithms and diagrams, and finally, a discussion of the evaluation procedure of the proposed methodology. The proposed model will be improving on previous methodologies by bringing in the application of deep learning algorithm to improve the existing problem domain.

3.1. The Proposed Framework

Figure 1 shows the generic framework for this research, the framework is logically divided to five major phases, and they are:

1. Data Extraction
2. Dataset Prepossessing Phase
3. Representation of Dataset
4. Building of model
5. Training and evaluation phase

3.2. Data Extraction Phase

Data Extraction phase represents the raw data that has been collected from the primary source, which is the twitter data that was scraped using twitter API. In order to compare the results of this model with state-of-the-art, the dataset used by Joshi et. al.[11] was used for training this model. And for the testing this model, a set of 5199 tweet and comment from 10 different political tweets of two users was downloaded. Twitter API was used to scrap these data with the aid of Tweepy and Twitter library on Python Notebook. This extracted data serves as input to the proposed model.

3.3. Dataset Prepossessing Phase

Dataset Prepossessing Phase is the point that involves transforming the extracted data into a clear and logical format for the model. This phase is the point where the extracted data and issues surrounding it were resolved to make the data fit for the proposed model. For this phase, the following processes were carried out

3.4. Remove Stop-Word

In natural language processing (NPL), words that have no use in building up a model are referred to as stop words. Commonly used term (such as "is," "a," "to," "an") that a search engine is programmed to ignore, both when indexing search entries and when they are retrieved as a result of a search query.

For this work, we removed those words that are less than three letter word, also we removed every retweet and also, we made you of NLTK(Natural Language- Toolkit) in python that has a list of stop words stored in 16 different languages.

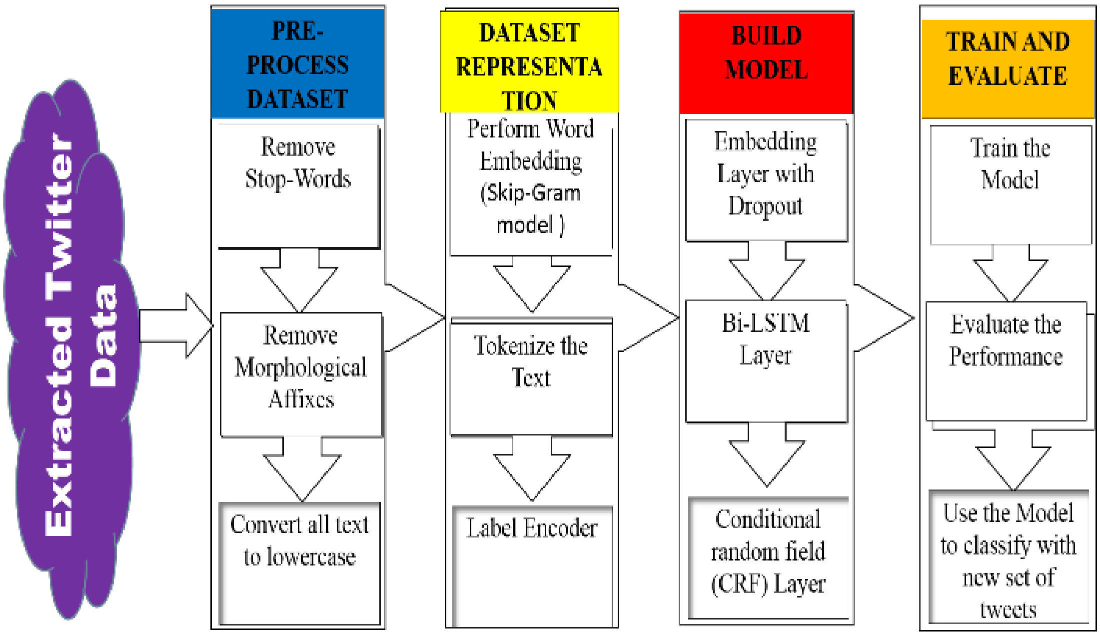

Figure 1: Generic diagram of the methodology

3.5. Remove Morphological Affix:

This is the stage of removing affix from modified or transformed words to their base root word. E.g., in the set {worker, working, works} the root is 'work'. Affix is taking in of suffix and prefix. A suffix is attached at the end of root word while prefix is attached beginning of the root word. We are removing affix, so that words with the same root will be seen as synonyms.

3.6. Convert All Text To Lowercase:

Text often has a diversity of capitalization showing the beginning of sentences, stressing of proper nouns. The widely used method is to reduce everything to lower case for simplicity. Converting all text to lowercase was done to avoid different variation in input capitalization (e.g. 'Nigeria' vs. 'nigeria') that can result to giving us different types of output or unexpected output. This may probably happen if the dataset has a mixed-case occurrences of the word 'Nigeria' and there is insufficient evidence for the Deep learning network to effectively learn the weights for the less common version.

3.7. Representation of Dataset

This is the phase where the preprocessed data are converted to the form which the deep learning network can easily work with. It is at this stage that Word embedding was done. This is the point where the representation of document vocabulary was converted to vector. Word embedding is a process of capturing a word's context in a document, semanthropic and syntactic similarity, relationship with other words, etc. and also convert them to vector representations of a particular word. Word2Vec was developed by Tomas Mikolov at Google in 2013 as the most common method for learning word embeddings using shallow neural network. Word2vec is a mainly computationally-effectual predictive model for learning word embedding from raw text. It is of two main types, the Continuous Bag-of-Words model (CBOW) and the Skip-Gram model.

For this work, the Skip Gram model was employed. CBOW predicts target words (e.g. 'football') from the source background words ('the boy likes to play'), while the skip-gram does the reverse and predicts context-words from the target words. This reversal might appear like a subjective choice, moreover skip-gram treats each context-target combine as a modern perception, and this tends to do way better when working with bigger datasets. This research work, used Skip Gram model for the word embedding stage. Implementing the Skip-gram Model, this work leveraged on Bible corpus of Tomas Mikolov, (2013) Which is included in the training variable norm bible for the model. The implementation process was on five stages: Create the vocabulary of the corpus, Skip-gram [(target, context), relevant] generator, Construct the model architecture of the skip-gram. Train the Model and Get Word Embeddings.

3.8. The BI-LSTM-CRF Model

In this research work, we propose a different way of dealing with different sentence types so as to make it easier to extract and predict sarcasm in the sentences. In particular, we investigate the relationship between the users post on twitter and the reply and comment that follow such post, investigating using previous and current response. This research work proposes a framework for improving sarcasm detection using both the linguistic and contextual feature, this contextual feature are when and what was in the previous and current reply of the user, together with the original post that generated the reply and comment.

For example, a tweet written by a company about the specification of their new product and the services of the product. After some minutes of the post by the company, one of the user put a comment under the post that says “Wao! I Love this product, it is one of the best products of all time”. Minutes after the comment, the company replied to the comment of the user, saying “Thank you customer, we are the best because we offer the best services”. From that, a different user begins to review the product, based on the first comment of the user but after few minutes again, the first user wrote a comment again under the response of the company reply and the user says “ Do I mean LOVE? Your product that got spoilt some days after I purchased it” with the last comment/post of the first user, it is clear that his first comment to the company post is a sarcastic comment, that can be misclassified because of the presence of the positive words (LOVE and BEST) and this will definitely affect other NLP work on the company analysis of the post. Relying only on the previous post of the first user alone and not connecting it to the current post will affect the classification.

Based on this observation, a deep neural network architecture model was employed to carry out this task, which is a bidirectional long short-term memory with conditional random fields (Bi-LSTM-CRF), two phases were employed to classify if a reply or comment to a tweet is sarcastic or non-sarcastic. In the first phase, classification was carried out separately using the comment and the reply alone. In the second phase, the classification considers both the reply and the context of the reply with the original tweet. For these two phases, experiment was carried out using the Bi-directional Long-Short Term Memory (Bi-LSTM).

The inclusion of Conditional Random Field (CRF), a probabilistic model for structured prediction, is another type of probabilistic discriminative model, representing a Single log-linear distribution over structured outputs in function of a particular input sequence of observations. Inclusion of CRF will help to predict from the output of both forward and backward propagation of the LSTM. Bi-LSTM-CRF is one of the deep neural sequence models in which a bi-directional long-short term memory layer (Bi-LSTM) and a conditional random field (CRF) layer are stacked together for sequence learning, as shown in Figure 2 Bi-LSTM combines a forward long-term memory layer (LSTM) and a backward LSTM layer in order to learn information from both the preceding and the following tokens. LSTM is a kind of Recurrent neural network (RNN) architecture that has a hidden units, with long-short term memory modules.

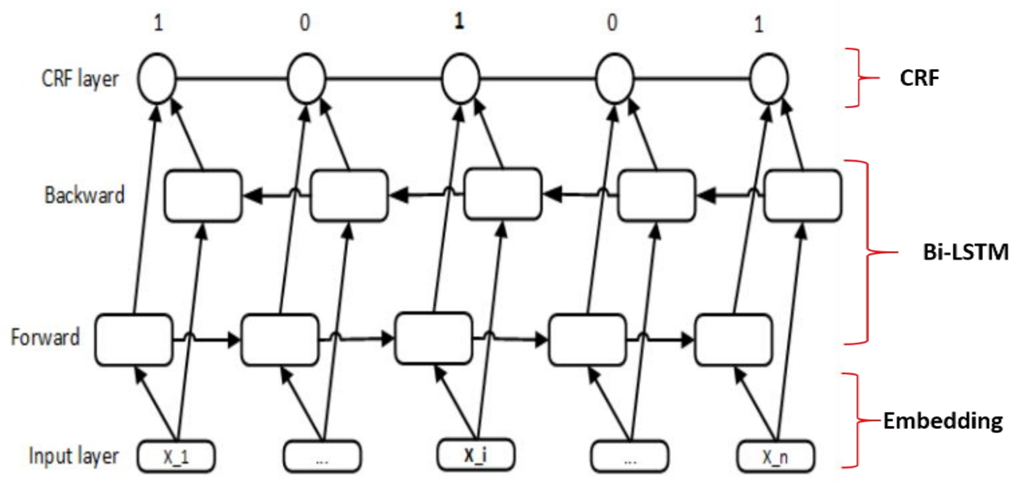

Figure 2 Framework of the Bi-LSTM-CRF

3.9. Experimental Analysis

This section of the research work discusses the details of the experiments carried out. This section put forward the yielded results of the results of the analysis and discuss the prediction process of the work.

Data and Parameters

To compare the results of this model with state-of-the-art, the dataset generated by [1] which was also used by [22] to train this model as mentioned above was chosen.. The datasets are comprised of 25,991 tweets which comprises of both tweets and the comments: one balanced and one imbalanced. The balanced data set contains 12,215sarcastic, that is, post and comment with hash tag such as, #sarcasm, #sarcastic, #iron and for the test data, 5199 tweets and comment from 10 different tweets of two users were downloaded. Twitter API was used to scrap these data with the aid of Tweepy and Twitter library on Python Notebook.

Pre-processing of the collected data, retweets, duplicates, quotes, tweets containing only hashtags and URLs or less than three words have been removed but the user ID has not been removed to determine whether the user has commented on the post earlier or whether that is the user's first conversation on a tweet. To create a conversation context, the 'reply to status' parameter in the tweet was used for each train and test dataset to determine if it was in response to a previous tweet: if so, the last tweet (i.e. 'local conversation context') to which the original tweet replied was also downloaded.

```
Layer (type)                 Output Shape              Param #
=================================================================
embedding_1 (Embedding)      (None, 30, 256)           8676352
_________________________________________________________________
Conditional RF (CRF)         (None, 28, 256)           196864
_________________________________________________________________
lstm_1 (LSTM)                (None, 26, 256)           525312
_________________________________________________________________
lstm_2 (LSTM)                (None, 256)               525312
_________________________________________________________________
dense_1 (Dense)              (None, 256)               65792
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 514
_________________________________________________________________
activation_1 (Activation)                              (None, 2)      0
```

Figure 3. Parameters used for the model

Figure 3. Shows the parameters used to build the model, the set of parameters used for this model consists of parameters of Bi-LSTM built on Kerasthey are the configuration variables that is internal to the model and whose value can be estimated from data. The parameter used are shown in the figure 3.

4. RESULT AND EVALUATION

To evaluate the model, precision, recall and F1- measure were used as the evaluation metrics. From the model result in Table 1, the accuracy of the model is 0.9211, with 0.92134232, 0.9122and 0.9131832 precision, recall and f-score respectively. Also from the test data of 10 different tweets with their reply, the output of each user with the accuracy for each user is presented in figure 4.

Table 1: Showing the performance matrices of the model

Accuracy	0.9211			
Precision	0.92134232			
Recall	0.9122			
F_Score		Precision	Recall	FScore
	0	0.920	0.912	0.901
	1	0.917	0.917	0.915
Micro avg		0.919	0.913	0.912
Micro avg		0.921	0.913	0.912
Weighted Avg		0.920	0.913	0.912

The test data of tweets and comment was tested and the result of the test is shown in figure 3 Twitter boasts 330 million monthly active users (as of 2019 Q1). Of these, Over 40 per cent of these, or more accurately 134 million, use the service on a regular basis. The result shows the number of sarcastic reply for each data and the accuracy for each data. The graphical representation is shown in figure 5.

	Tweet	number of reply	unique user	Sarcasm	accuracy
0	Tweet1	537	472	114	0.90670
1	Tweet 2	684	513	211	0.89550
2	Tweet 3	538	468	98	0.91430
3	Tweet 4	631	543	127	0.91990
4	Tweet 5	782	623	192	0.92340
5	Tweet 6	632	572	101	0.91230
6	Tweet 7	554	453	87	0.92390
7	Tweet 8	693	543	140	0.91440
8	Tweet 9	883	553	137	0.91820
9	Tweet 10	458	398	111	0.91340
10	average				0.91195

Figure 4: showing the accuracy of the model on the test dataset for 10 different tweet

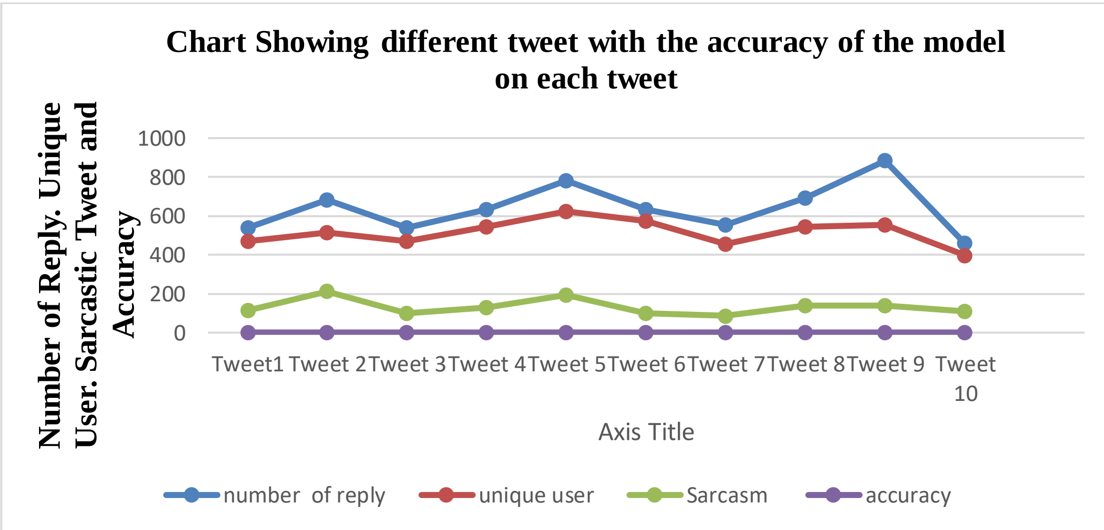

Figure 5: graphical representation of the sarcasm detection on 10 set of data and the accuracy

5. Conclusion

Sarcasm is a complex form of speech act in which the speakers convey their message in an indirect way. One important aspect of the sarcastic speech is that it is often difficult to be conscious of it. The complexity in recognition of sarcasm causes confusion and misinterpretation in everyday communication and causes difficulties to many NLP systems. This research work makes a complementary impact to the existing work of modeling sarcasm detection by considering the lexical and the contextual feature in detecting sarcasm in social media. For this research, the particular contextual feature used is by looking at a particular post with the comments that follow such post to know when the polarity of a comment changes from another and also when the polarity of the comment change from the original post. This research work shows how lexical feature to contextual feature can be usefully fused together to yield an improved sarcasm detection.

To achieve this, the Bi-Directional Long Short-Term Memory with Conditional random field (Bi-LSTM-CRF) architecture was used to build the model that extracted both the lexical and contextual features and predict. In this architecture, two phases of classification was done, in the first phase, classification was carried out separately using the comment of each user. In the second phase, the classification considers both the reply and the context of the reply with the original tweet. For these two phases, experiment was carried out using the Bi-directional Long-Short Term Memory (Bi-LSTM). The inclusion of Conditional Random Field (CRF), which is a probabilistic model is for structured prediction base on the two classification. The result of the model shows the model gave us has 0.9211 accuracy, the average precision of the model is 0.92134232, while the recall is 0.9122 and the f-score is 0.9131832 which is a slight improvement on existing model in 10-fold cross validation.

REFERENCES

[1] Bharti, S. K., Babu, K. S., & Jena, S. K. (2015). Parsing-based Sarcasm Sentiment Recognition in Twitter Data. Proceedings of the 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining 2015 - ASONAM '15, 1373–1380. https://doi.org/10.1145/2808797.2808910

[2] Rajadesingan, A., Zafarani, R., & Liu, H. (2015). Sarcasm Detection on Twitter. Proceedings of the Eighth ACM International Conference on Web Search and Data Mining -WSDM '15, 97–106. https://doi.org/10.1145/2684822.2685316

[3] Microsoft Encarta, 2008

[4] Macmillan English Dictionary, 2007.

[5] Bouazizi, M., & Otsuki, T. (2016). A Pattern-Based Approach for Sarcasm Detection on Twitter. IEEE Access, 4, 5477–5488. https://doi.org/10.1109/ACCESS.2016.2594194

[6] Saha, S., Yadav, J., & Ranjan, P. (2017). Proposed Approach for Sarcasm Detection in Twitter. Indian Journal of Science and Technology, 10(25), 1–8. https://doi.org/10.17485/ijst/2017/v10i25/114443

[7] Bindra, K. K., Prof, A., & Gupta, A. (2016). Tweet Sarcasm : Mechanism of Sarcasm Detection in Twitter. 7(1), 215–217.

[8] Peng, C., Pan, J. W., Edu, C. S., & Edu, J. S. (2015). Detecting Sarcasm in Text : An Obvious Solution to a Trivial Problem.

[9] Barbieri, F., Saggion, H., & Ronzano, F. (2014). Modelling Sarcasm in Twitter, a Novel Approach. Proceedings of the 5th Workshop on Computational Approaches to Subjectivity, Sentiment and Social Media Analysis, 50–58

[10] González-Ibáñez, R., Muresan, S., & Wacholder, N. (2011). Identifying sarcasm in Twitter: a closer look. Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics: Human Language Technologies: Short Papers-Volume 2, (2010), 581–586. https://doi.org/10.1.1.207.5253

[11] Joshi, A., Sharma, V., & Bhattacharyya, P. (2015). Harnessing Context Incongruity for Sarcasm Detection. Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Short Papers), 51(4), 757–762. https://doi.org/10.1016/j.ipm.2014.09.003

[12] Clews, P. (2017). Rudimentary Lexicon Based Method For Sarcasm. International Journal of Academic Research and Reflection, 5(4), 24–33.

[13] Felbo, B., Mislove, A., Rahwan, I., Lehmann, S., & Science, I. (2016). Using millions of emoji occurrences to learn any-domain representations for detecting sentiment, emotion and sarcasm.

[14] Jyoti, S. (2017). Twitter Sarcasm Detector (Tsd) Using Topic Modeling on user description. International Conference on Computer Science Networks and Information Technology (August), 94–104.

[15] Ghosh, D., Richard, A., & Smaranda, F. (2016). The Role of Conversation Context for Sarcasm Detection in Online Interactions.

[16] Zhao, Z., Chen, W., Wu, X., Chen, P. C. Y., & Liu, J. (2017). LSTM network : a deep learning approach for short-term traffic forecast. 68–75. https://doi.org/10.1049/iet-its.2016.0208

[17] Qian, Q., & Huang, M. (2015). Linguistically Regularized LSTM for Sentiment Classification. of arXiv:1611.03949v2 [cs.CL], 76-87

[18] Ghaeini, R., Hasan, S. A., Datla, V., Liu, J., Lee, K., Qadir, A., … Farri, O. (2018). DR-BiLSTM : Dependent Reading Bidirectional LSTM for Natural Language Inference ∗. 1, 1460–1469.

[19] H, U. M., Zavala, R. M. R., Mart, P., & Segura-bedmar, I. (2018). A Hybrid Bi-LSTM-CRF model for Knowledge Recognition from eHealth documents.

[20] Paper, C., & Burtsev, M. S. (2018). Application of a Hybrid Bi-LSTM-CRF Model to the Task of Russian Named Entity Recognition Application of a Hybrid Bi-LSTM-CRF model to the task of Russian Named Entity Recognition. (September). https://doi.org/10.1007/978-3-319-71746-3

[21] Zhou, P., & Qi, Z. (2016). Text Classification Improved by Integrating Bidirectional LSTM with Two-dimensional Max Pooling. 2(1), 3485–3495.

[22] Bamman, D., & Smith, N. A. (2015). Contextualized Sarcasm Detection on Twitter. Icwsm (International AAAI Conference on Web and Social Media), 574–577. https://doi.org/10.1145/2684822.2685316