University of Ibadan
Journal of Science and Logics in ICT Research

Sarcasm Detection Using Lexical and Contextual Features of Deep Learning Architecture

ADEYEMO, Adesesan Barnabas sesanadeyemo@gmail.com

ADEWUYI, Joseph Oluwaseyi adewuyiseyi@gmail.com

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

Abstract

In contemporary time, social media sites such as Facebook, LinkedIn, Twitter, etc. have expanded and received vast admiration and significance. These sites have change into large environments where users express their ideas, views and opinions naturally. Companies and organizations leverage this unique environment to tap into people's opinion on their products or services and to make available instantaneous customer assistance. This research seeks to avoid the use of grammatical words as the only features for sarcasm detection but also the contextual features, which are theories explaining when, how and why sarcasm is expressed. The contextual features consider the user's current and previous posts to detect or classify if a post is sarcastic. 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), which is a probabilistic model for structured prediction helped to predict using the output of both forward and backward propagation of the LSTM. The performance of the models was evaluated using the following metrics: Accuracy, Precision, Recall, F-measure. The model has 0.9211 Accuracy, 0.92134232 precision, 0.9122 recall and 0.9131832 F-measure.

Keywords: Sarcasm Detection, Deep Learning, Bi-LSTM, Conditional Random Field (CRF), Contextual features

I. INTRODUCTION

In contemporary time, social media sites 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 have become a well-established platform for users to express their feelings and opinions on various topics, such as events, individuals or products. Social media channels have become a popular platform to discuss ideas and to interact with people worldwide. For instance, Facebook claims to have 1.59 billion monthly active users, each one being a friend with 130 people on average. Similarly, Twitter claims to have more than 500 million users, out of which more than 332 million are active. Users post more than 340 million tweets and 1.6 billion search queries every day [1].

These days, social media networks are habitually the first place to get the reaction about contemporary occurrences and trends from user center, allowing them to provide companies with vita data that can be used to position their products in the market as well as gather quick reaction from customers. When an event commences or a product is launched, people start tweeting, writing reviews, posting comments, etc. on social media. People go to social media platforms to read reviews from other users about a product before they make decision whether to procure the product or not. Organizations, groups, bodies also depend on these social media sites to know the response of users for their services and

ADEYEMO, Adesesan Barnabas and ADEWUYI, Joseph Oluwaseyi O. E (2019). Sarcasm Detection Using Lexical and Contextual Features of Deep Learning Architecture, ***University of Ibadan Journal of Science and Logics in ICT Research (UIJSLICTR)*****, Vol. 3, No. 1, pp. 47 – 59**

successively use the feedback to enhance their services.

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 surprisingly, most large companies have a social media presence and a dedicated team for marketing, after-sales service, and consumer assistance through social media [2]. With the high velocity and volume of social media data, organizations and companies need to perform different tasks like content management, sentiment analysis, and extraction of relevant messages for the service representatives to respond to.

However, finding and verifying the legitimacy of opinions or reviews is a formidable task, to analyze these various opinions is a big task because 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 difficult to manually read through all the reviews and peoples opinion and determine which of the opinions expressed are sarcastic or not. In addition, the common reader will have difficulty in recognizing sarcasm in tweets or product reviews, which may end up misleading them.

Encarta dictionary defines sarcasm as remarks that mean the opposite of what they seem to say and are intended to mock or deride [3]. Also One definition for sarcasm by Macmillan is the activity of saying or writing the opposite of what you mean, or of speaking in a way intended to make someone else feel stupid or show them that you are angry [4]. Sarcasm is a style of communication in which precise and intentional connotation are in contradictory. Sarcasm is often used to express a negative message using positive words. Automatic detection of sarcasm is then very important in Natural Language Processing (NLP) such as sentiment analysis, as a sarcastic phrase that includes positive words conveys a negative message and can be easily misunderstood and misclassify by an automatic system that perform this Natural Language processing. Sarcasm also occurs when an individual implies something else from what he or she is talking about.

Sarcasm is a complex form of speech act in which the speakers convey their message in an indirect way. One essential characteristic of the sarcastic speech act is that it is sometimes hard to be aware of it [5]. The complexity in recognition of sarcasm causes confusion and misinterpretation in everyday communication and causes difficulties to many NLP systems such as online review summarization systems, dialogue systems or brand monitoring systems due to the failure of state of the art sentiment analysis systems to detect sarcastic comments. For example-"I loved being cheated". Here "love" expresses a positive sentiment in 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 extended and employed to automatically detect sarcasm 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. This 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.

II LITERATURE REVIEW

The accuracy of sarcasm detection count on every aspect of language; from the lexical to the semantic. Detecting sarcasm needs multiple parameters in place to be effective; 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 involves 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.
Lexical feature class of sarcasm detection have to do with text property such as unigram, bigram, and n-gram. 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 referred to as unigram; size 2 as bigram etc.

Bindra et. al. [7] Used bigrams and unigrams that grouped single word (example: seriously, great, amazing, etc.) and double words (example: really bad, super amazing, very good, 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 employed in sentiment classification are: logistic regression (LogR) and support vector machine with sequential minimal optimization (SMO). 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 constructed a term frequency-inverse document frequency (TF-IDF) vector, and then fed it into a multinomial Naive Bayes classifier. Likewise, Barbieri et. al. [9] does not include patterns of words as features for detecting sarcasm but made use of seven sets of lexical features aim to detect sarcasm by its inner structure

Pragmatic feature based classification involves figurative and symbolic texts such as smilies, emoticons. Various authors have used pragmatic features for sarcasm detection. 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) To User, which marks if a tweets is a reply to another tweet (signaled 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 congruity, and (d) Explicit incongruity features. Lexical features are unigrams acquired using feature selection techniques such as χ^2 Test and Categorical Proportional Difference. Pragmatic features comprises emoticons, laughter expressions, punctuation marks and capital words

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

Clews [12] utilized string matching against positive sentiment and interjection lexicons to test if the presence of both can be used to classify content as being sarcastic or not. Also, Bharti et. al. [1] proposed a two approaches for detection of sarcasm in the text data. The first is the use of a parsing-based lexicon generation algorithm (PBLGA) and the second was to detect sarcasm based on the occurrence of the interjection word. 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 proposed a method to detect sarcasm, using a pattern-based approach on Twitter data by considering different types 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). For the pattern-based feature, they divided words into two groups: the first group, named as “CI” which contains words of which the content is important and the second group named as “GFI” containing the words of which the grammatical function is more important.

Felbo *et. al.* [13] presented a probabilistic modelling framework of identifying, classifying and learning features of sarcastic text via training a neural network with human-informed sarcastic benchmarks. Their approach used the concepts of parts-of-speech (POS) tagging to identify specific words that belong to the categorizations of the defined constraints, which are collectively pile up and aggregated as values to be fed into a two layer feed forward multi perceptron network to correctly 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 look into some types of Long Short-Term Memory (LSTM) networks that can model both the discussion setting and the sarcastic response, they made use of the conditional LSTM network and LSTM networks with sentence level attention on context and response outers form the LSTM model that reads only the response, they also built a model that qualitatively analysis attention weights produced by the LSTM models with attention and discuss the results compared with human performance on the task.

In the mid-90s, a variety of repetitive net with purported Long Short-Term Memory units, or LSTMs, was proposed by the German scientists Sepp Hochreiter and Juergen Schmidhuber as an answer for the disappearing inclination issue [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 each position (h_t) only encrpts the prefix context in a forward direction while the backward context is not considered [17]. Bidirectional LSTM used two parallel passes (forward and backward) and concatenated hidden states of the two LSTMs as the representation of 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 the hidden state of BiLSTM could be defined as:

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

Recently, Ghaeini et. al.[18], proposed model called “dependent reading bidirectional LSTM network(DR-BiLSTM)” to model the connection between a premise and a hypothesis during encoding and inference, they also present an ensemble approach to syndicate the models.

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. Zavala *et. al* [19] used a new approach of LSTM+CRF to remove adverse drug reactions from user appraisals, by concatenating recurrent neural network and a CRF that operates on the scores extracted by 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" proposed a 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.

Zhou et. al. [21] proposed a cross Bi-LSTM and CRF model, The architecture has two bidirectional Long Short-Term Memory (LSTM) layers and a last layer based on Conditional Random Field (CRF). They claimed that their model add sense-disambiguation embedding and an extended tag encoding format to detect discontinuous entities, as well as overlapping or nested entities.

II METHODOLOGY

This Section of the work will discuss 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.

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

DATA EXTRACTION PHASE

Data Extraction phase represents the raw data that has been collect 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 tweet of two user 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.

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 resolve to make the data fit for the proposed model. For this phase, the following processes were carried out

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 word (such as "the", "a", "an", "in") that a search engine has been programmed to ignore, both when indexing entries for searching and when retrieving them as the 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.

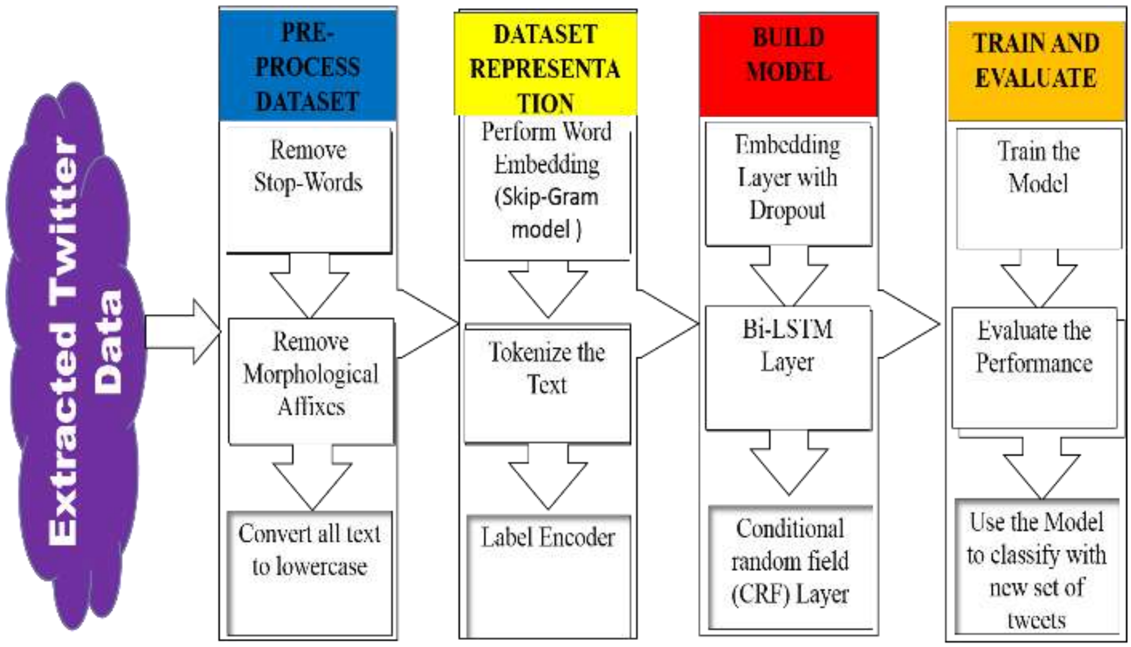

Figure 1: Generic diagram of the methodology

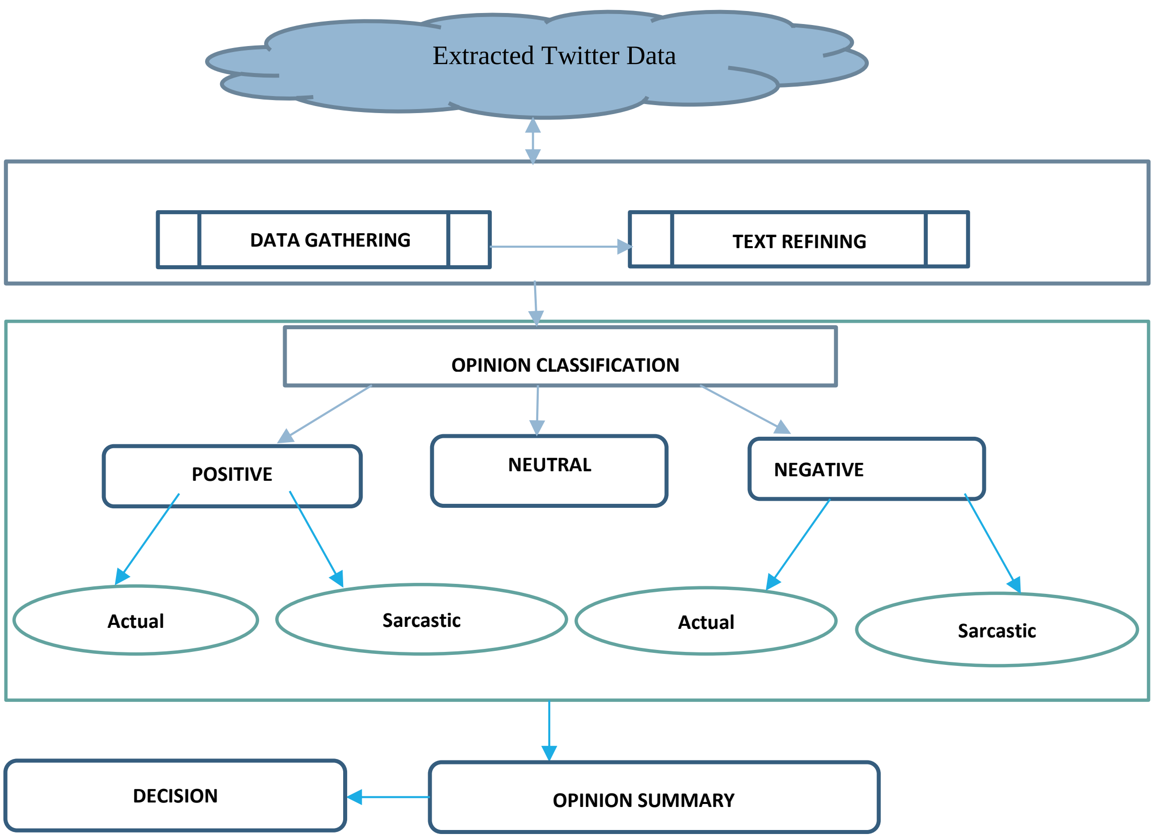

Figure 2: Existing framework

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. 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.

Convert All Text To Lowercase:
Text often has a diversity of capitalization showing the beginning of sentences, stressing of proper nouns. The most common approach 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.

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 representation of document vocabulary was converted to vector. Word embedding is a process of capturing context of a word in a document, semantic and syntactic similarity, relation with other words, etc. converting them to vector representations of a particular word. Word2Vec the most common method to learn word embedding using shallow neural network, it was developed by Tomas Mikolov in 2013 at Google. Word2vec is a mainly computationally-effectual predictive model for learning word embedding from raw text. It is of two main type, 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 source context words ('the boy likes playing'), while the skip-gram does the inverse and predicts source context-words from the target words. This inversion might seem like an arbitrary choice, also skip-gram treats each context-target pair as a new observation, and this tends to do better when working with larger 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 contained in the norm_bible variable for training the model. The implementation process was on five stages: Build the corpus vocabulary, Build a skip-gram [(target, context), relevancy] generator, Build the skip-gram model architecture. Train the Model and Get Word Embeddings.

THE Bi-LSTM-CRF MODEL

In this research work, we propose a different way in 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 propose 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 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 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), which is a probabilistic model for structured prediction, is another kind of discriminative probabilistic model, which represents a single log-linear distributions over structured outputs as a function of a particular observation input sequence. 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 deep neural sequence models, where a bi-directional long short-term memory (Bi-LSTM) layer and a conditional random fields (CRF) layer are stacked together for sequence learning, as shown in figure 2 Bi-LSTM incorporates a forward long short-term memory (LSTM) layer and a backward LSTM layer in order to learn information from preceding as well as following tokens. LSTM is a kind of recurrent neural network (RNN) architecture with long short-term memory units as hidden units.

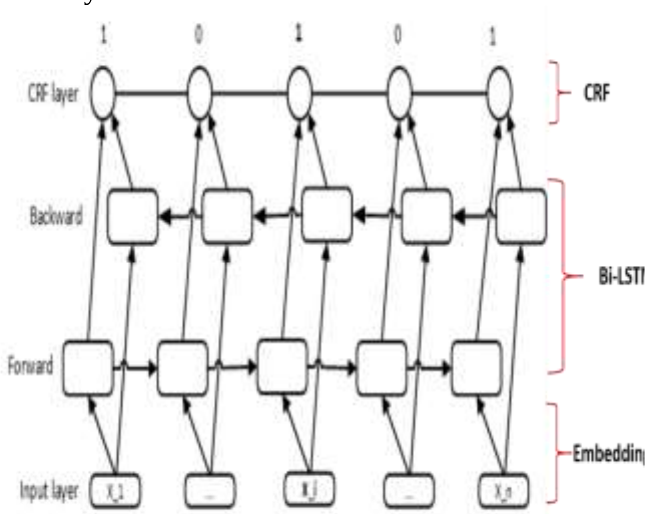

Figure 3 Framework of the Bi-LSTM-CRF

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

In order to compare the results of this model with state-of-the-art, the dataset created by [1] which as previously mentioned was also used by [22] for training this model 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 tweet and comment from 10 different 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.

Preprocessing the data collected, retweets, duplicates, quotes, tweets that contain only hashtags and URLs or are shorter than three words were removed but the user ID was not remove, so as to know if the user has comment on the post earlier or if that is the user first conversation on a tweet. To build the conversation context, for each train and test dataset, the "reply to status" parameter in the tweet was used to determine whether it was in reply to a previous tweet: if so, the last tweet (i.e., "local conversation context") to which the original tweet was replying to was downloaded as well.

The parameters used to build the model 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.

IV. RESULTS 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 5.

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 4: Parameters used for the model

Table 1: Showing the performance matrices of the model

Accuracy	0.9211			
Precision	0.92134232			
Recall	0.9122			
F_score	0.9131832			
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 4. The result shows the number of sarcastic reply for each data and the accuracy for each data. The graphical representation is shown in figure 6.

	data	number of reply	unique user	Sarcasm	accuracy
0	data 1	437	302	76	0.89450
1	data 2	664	443	199	0.90340
2	data 3	488	278	111	0.91430
3	data 4	534	302	65	0.91990
4	data 5	744	499	189	0.90670
5	data 6	451	370	89	0.91230
6	data 7	554	345	137	0.92340
7	data 8	643	431	102	0.91440
8	data 9	235	207	32	0.91720
9	data 10	449	369	94	0.91340
10	average				0.91195

Figure 5: showing the accuracy of the model on the test dataset for 10 different tweet

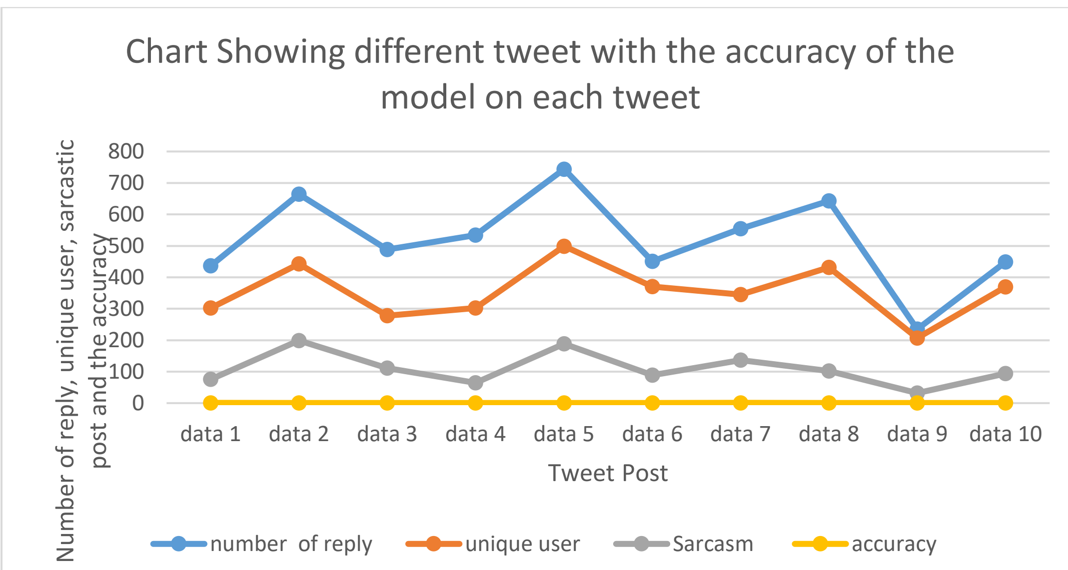

Figure 6: graphical representation of the sarcasm detection on 10 set of data and the accuracy

EVALUATION

Evaluating the new model with the current model that uses contextual features for sarcasm detection, it was discover that the new model perform better in term of accuracy, precision and recall and F-score as shown in Table 2. It can be deduced from the Table 2 that the new method (BiLSTM-CRF) achieves a better accuracy and precision result on average, with Debanjan Ghosh et al., [15] being the next best algorithm.

Table 2: Showing the comparison of the result of existing model and the new model

Author	Model	Accuracy	Precision	Recall	f- Score
Aditya Joshi (2015)	SVM	0.80	0.77	0.51	0.61
Supriya Jyoti et al(2017)	SVM	0.7011	0.7373	0.6063	0.6654
Supriya Jyoti et al(2017)	L. R	0.6975	0.7354	0.6168	0.6975
Debanjan Ghosh et al., (2017)	$LSTM^{conditional}$	0.8001	0.7725	0.7653	0.7630
New Model	Bi-LSTM-CRF	0.9115	0.91	0.91	0.9114787

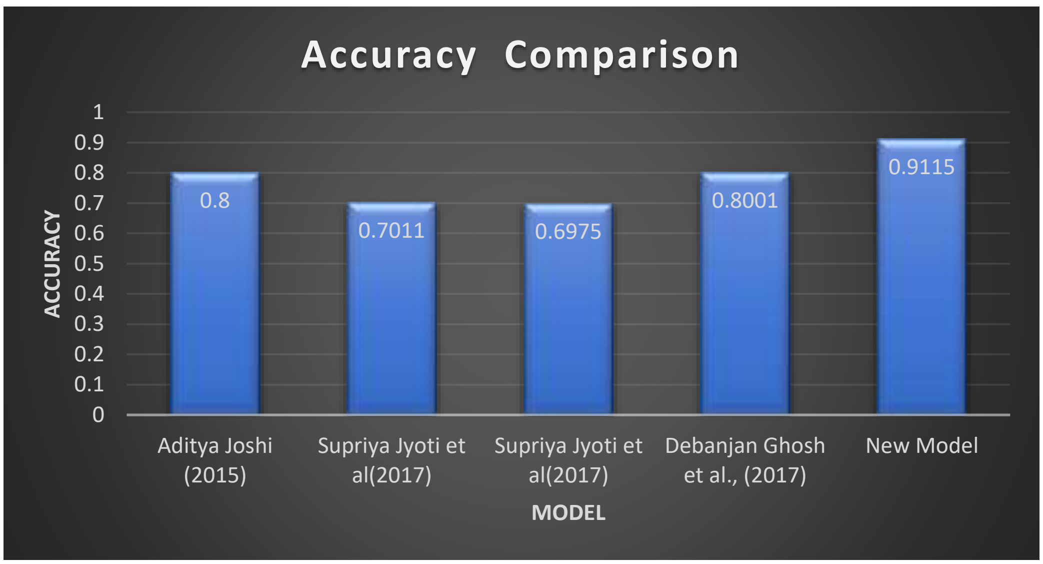

Figure 7: Showing the comparison in the accuracy of the new model and existing model

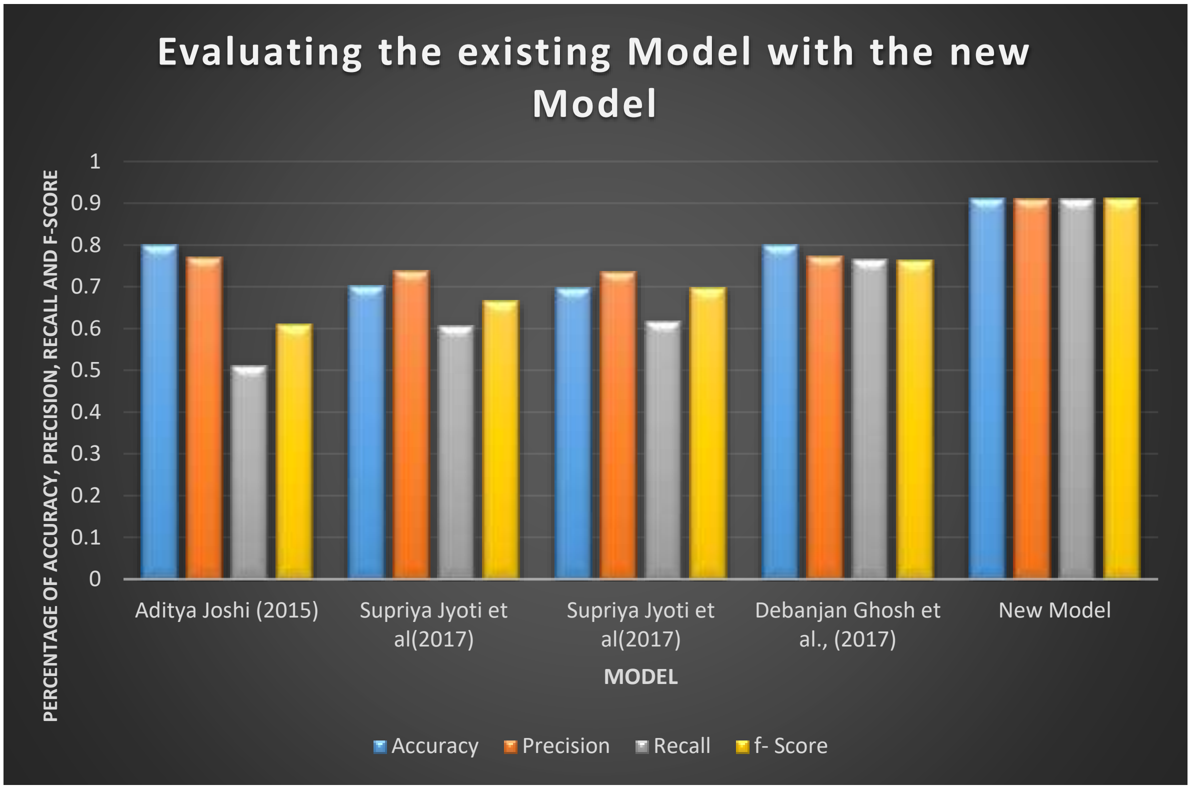

Figure 8: Graphical representation of the comparison with different model

V. CONCLUSION

Sarcasm is a complex form of speech act in which the speakers convey their message in an indirect way. One essential characteristic of the sarcastic speech act is that it is sometimes hard to be aware 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.

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. (2017). *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