A Novel Integrated Framework for Sarcasm Detection in Social Platform

Shawni Dutta, Akash Mehta, Samir Kumar Bandyopadhyay

Abstract: *Sarcasm is a form of speech which transforms the verbatim meaning of a sentence into its antonym. Sarcasm identification in social media is a crucial facet of the sentiment analysis process, since it deals with texts whose polarity is completely opposite from its utterance. Our paper provides an exhaustive review of the existing methodologies dedicated to the task of detecting sarcasm in texts posted on an online forum. A comparative analysis of the existing techniques, mentioning the datasets and the performance measure, is also provided. This paper also introduces a novel integrated framework for identifying sarcastic clues in tweets, and recognizing sarcastic users.*

Keywords : *Sarcasm Detection, Lexical Analysis, Natural Language Processing, Big Data, Deep Learning, MSELA.*

I. INTRODUCTION

Sarcasm is a satirical remark tempered by humor. Sarcasm has generally been used to create discrepancies and ambiguities in the minds of the listeners while mocking them or someone else. Sarcasm employs the use of ambivalence in order to keep the reader/listener guessing about the true intentions of the writer/speaker. Sarcasm, while speaking, is accompanied by a change in tone, body language and facial expressions. This makes it easier for sarcasm to be detected in a vocal mode of communication. When it comes to text however, the previously mentioned indicators are absent. Sarcasm detection [1] in texts is solely done on the basis of lexical structures, use of grammar and contextual information. This makes Sarcasm detection in texts a challenging task, thereby explaining the immense research interest in them.

Sarcasm is a way of expressing negative feelings using some positive words and phrases, or vice-versa. For example, "I just love to work in this romantic weather #sarcasm" utters the negative feelings using positive words. Sarcasm is often used as a weapon to make jokes, to be humorous, or to criticize or make remarks about any commodity, person or any events. Sarcasm is defined in many ways by different authors. According to [2], the situational differences between the text and the context are often regarded as Sarcasm. In [3], this linguistic phenomenon is referred as a collection of 6 tuples- speaker, listener, context, utterance, literal proposition, intended proposition. Sarcasm is defined as a player in the field of polarity switching [2] [4]. Depending on the usage, Sarcasm may belong into following categories such as- Sarcasm as wit (to indicate funny emotions), Sarcasm as whimper (to show irritation), Sarcasm as Evasion (to avoid delivering clear answer) [1].

With the initiation of Social media, detecting the wide array of sentiments online has become a very important task. Online Sentiment Analysis [5] in text involves the detection and study of human emotions present in scripts shared on an Online platform. This helps the computer to understand the emotions of a user while he/she was writing the piece of text. The applications of Online Sentiment Analysis [5] include detection of depression in users, classifying texts as true or false, detecting fake reviews online, detecting spams in online social forums and so on. Sarcasm is a special type of sentiment which needs to be detected in order to accelerate the domain of sentiment analysis task. Sarcasm detection would not only prosper as a stand-alone field, but also improve the efficiency of the other Online Sentiment Analysis [5] tasks mentioned above. By classifying texts as sarcastic, systems would not only be able to comprehend the user's points more clearly, but also be able to distinguish between lies and sarcastic posts – a distinction most self-learning systems fail to comprehend. Sarcasm Detection [1] also has its application in fake online review detection. Detecting sarcasm in online reviews help in obtaining clarity about the opinions on a particular product. This would help in improving the efficiency of a Recommender System. Like sarcasm, irony is also a type of sentiment that also behaves as a polarity switcher. Many researchers consider both of these as overlapped concepts. However, there are opinion differences- sarcasm is offensive and delivered with a cutting tone as compared to irony [6].

This paper provides a critical analysis of the various existing techniques for detecting sarcasm in texts and proposes a novel integrated framework for identifying sarcastic clues in tweets and recognizing users more prone to using sarcasm. The next section briefly discusses the existing research methodologies for identifying sarcasm. Section III provides a comparative study of the techniques discussed, including mentioning the datasets on which the experiments were performed and the performance measure obtained.

Retrieval Number: D7519049420/2020©BEIESP
DOI: 10.35940/ijeat.D7519.049420
Journal Website: www.ijeat.org

943

Published By:
Blue Eyes Intelligence Engineering
& Sciences Publication
© Copyright: All rights reserved.

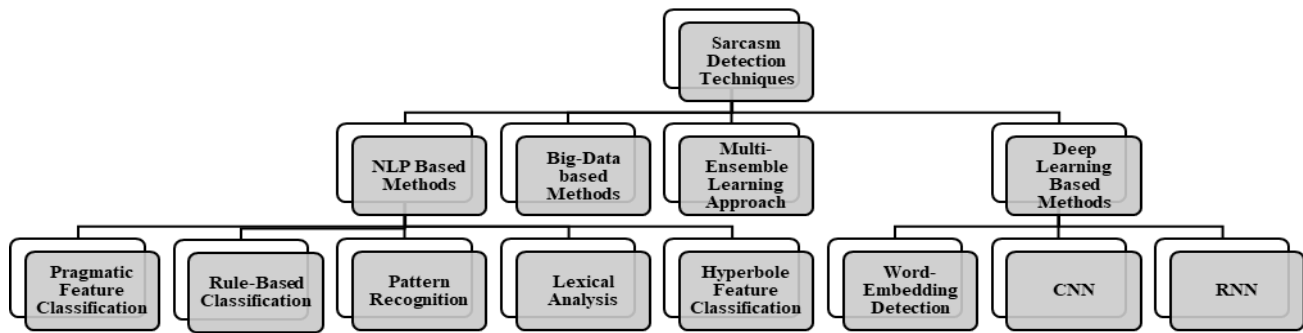# A Novel Integrated Framework for Sarcasm Detection in Social Platform

Figure 1. Classification of Sarcasm Detection Methodologies

Section IV discusses a novel integrated framework that assimilates the concept of discovering sarcasm in tweets and identifying sarcastic users. The paper concludes by discussing the future research scope of the problem examined.

II. LITERATURE REVIEW

In recent years, many researches have drawn attention to sarcasm detection on social media. The main objective of sarcasm detection tool is to distinguish a sarcastic text from a non-sarcastic one. In order to perceive that, researchers employed several techniques like NLP based methods, Big data-based approach, Deep learning-based approach and Multi-Ensemble Learning Approach. Figure 1 given below shows the classification of the various sarcasm detection Paradigms. Our paper delivers an all-inclusive review of the various researches done pertaining to each domain mentioned above.

A. Natural Language Processing Based Methods

Natural Language Processing (NLP) based methods were the first set of techniques that were applied to detect sarcasm in text. NLP based methods could be further classified into the following domains – Lexical Analysis, Pragmatic Feature Based Classification, Hyperbole Based Classification, Pattern Recognition and Rule-Based Classification. Lexical Analysis involves the extraction of features using the unigram, bigram or n-grams approach. Pragmatic features denote symbolic and figurative texts such as emoticons, smileys, @user, replies, etc. A hyperbolic text comprises of text properties, such as interjections, intensifier, punctuations, quotes, etc.

Kreuz and Caucci in their paper [7] perceive sarcasm by considering certain lexical clues. They hypothesize that the presence of certain interjections, adjectives, adverbs, and the use of certain punctuations are key indicators for sarcasm. This paper considers five indicators to be of interest – number of words, number of bold-faced words, interjections, adjectives/adverbs and punctuation marks. Experiments proved that interjections, adjectives/adverbs and punctuation marks are key sarcasm indicators compared to the rest.

Liebrecht et. al. in [4] make use of bigrams and trigrams as feature extractors for discovering sarcasm in text. A balanced winnow algorithm [4] is used to rank the extracted features, whose further analysis signals that sarcasm which is often articulated by the use of hyperbole, intensifiers and exclamations; and hence #hashtag sarcasm often fails to achieve true characteristics of sarcasm. This approach is Dutch language-centric and can also be extended for other languages.

Riloff et. al. in [8] contemplates that a sarcastic text is a combination of opposite emotions and polarity. In other words, if a tweet comprises of contrasting sentiments, it is highly likely that it is going to be a sarcastic tweet. They employed a bootstrapping learning framework that learns to differentiate between affirmative and deleterious sentiments from sarcastic tweets. This paper tries to exploit the syntactic structure of these tweets to obtain phrases, from a sentence, having contrasting polarity. The algorithm works with the following assumptions:

1. The positive sentiment is present in either the verb phrase or in predicative expressions;

2. The negative sentiment is present as the compliment to the verb phrase;

3. The positive sentiment occurs to the left of the negative sentiment;

4. The word "love" is used as a seed word to denote positive sentiment;

The n-grams that follow "love" is the negative sentiment phrase. Scores are assigned to the various n-grams obtained. The one with the highest score is taken as the negative phrase. Once the negative phrase has been obtained, the algorithm then proceeds to obtain the positive phrase through the n-grams [9] preceding the negative phrase. Verb and predicative phrases are considered as Positive Phrases, whereas the n-grams following it are considered as negative phrases. The algorithm learns the affirmative and deleterious phrases alternatively, until no new information can be obtained. However, this approach does not encompass all possible combination of sarcasm, in fact this contrasting method also labels some tweets as sarcastic which may not be sarcastic which provides insights into the future work of this paper.

Maynard and Greenwood employed a rule-based approach [10] where the contents of hashtags are analysed for sarcasm detection.

The hashtags are tokenised - multiple words are broken into individual words - and by studying the polarity of these words, sarcastic sentiment is recognized. One huge drawback of this method is the over-reliance on hashtags.

Only considering hashtag words limited the potential of their system, which would have greatly benefitted from the scrutiny of the entire textual body of the tweet.

Tungthamthiti et. al in [11] presented a method which identified indirect contradiction between sentiment and situation in texts by using a combination of the following approaches - sentiment analysis, concept level and common-sense knowledge expansion, recognizing coherence of sentences and performing classification using machine learning approaches. A feature vector was formed containing the N-grams [9] feature, Contradiction feature, Sentiment feature, Punctuation and special symbols feature. Later the feature vectors, generated from the tweets, were fed to a Support Vector Machine (SVM) [12] for training and then executing the final classification task. The use of ambiguous words in concept level expansion may lead to wrong classification of sarcastic tweets which limits their approach from achieving higher efficiency.

Tsur et al. [13] demonstrated a semi-supervised learning algorithm, for recognizing sarcasm that exploits both syntactic and pattern-based feature extraction. The algorithm consists of two stages: semi-supervised pattern acquisition, and sarcasm classification. Davidov et. al [14] has employed the semi-supervised learning algorithm [14] given by Tsur et. al. [13] on the Twitter and the Amazon review dataset, where a scale ranging from 1 to 5 is used to denote level of sarcasm. They employed a k-NN [15] based classification model that uses pattern-based features - such as high frequency and content words - as well as punctuation-based features - such as sentence length, number of capitalised words, exclamation marks, question marks, quotes present in the sentence - for sarcasm detection. They follow the automatic pattern extraction algorithm given by Davidov and Rapport 2006 [13]. They tried to determine whether #sarcasm is enough to detect sarcasm or not, and concluded that #sarcasm is often ambiguous and biased and frequently misclassifies sarcastic ones as non-sarcastic and vice-versa. The method proposed by Davidov was limited due to the fact that it relied heavily on the syntactical structure of the tweets, with little or no attention being paid to the contextual information available. Davidov et al. wished to extend their work to incorporate review rankings and summarization for brand monitoring systems. Taking the past tweets of the reviewers and other contextual information into account, would have greatly helped with the process.

González-Ibáñez et al. [16] realised that along with lexical features, pragmatic features - such as positive and negative emoticons, user mentions - are equally important to ensure the coherence between positive and negative tweets. They employed SVM [17] with sequential minimal optimization (SMO) and logistic regression (LogR) [18] for this purpose. This paper has a good potential in terms of results, however considering contextual features may increase the accuracy of the findings.

Unlike methods suggested by Davidov [14], González-Ibáñez [16], Riloff [8], Barbieri [19] preferred using features - such as punctuations, use of out of context words - over words and patterns of words. This model used seven set of features such as Frequency; Written-Spoken style uses; Concentration of adverbs and adjectives; Structural features like length, punctuation, emoticons; Sentiment gap between affirmative and deleterious terms; Synonyms; and Ambiguity. They make use of a supervised learning algorithm by employing Decision Tree classifier [20] in order to identify sarcasm. However, the efficiency can be enhanced by incorporating additional features such as language models.

Bharti et. al. [21] projected two methods for perceiving sarcastic tweets: a parsing-based lexicon generation algorithm; and the occurrences of interjection words. They employed bigram [22] and trigram [23] to generate bags of lexicons for sentiment and situation in tweets. In the first approach, contradiction between sentiment and situation plays a vital role, whereas the second approach focuses on dealing with identifying hyperbole - such as interjection word and intensifier - as it signals sarcastic tweets [7][4].

A. Rajadesingan et. al. [24] proposed a method that tends to detect sarcasm by considering the contextual features from the history tweets. They classify sarcasm into: Sarcasm as a contrast of emotions (Type-1 Sarcasm); Sarcasm as a complex form of expression (Type-2 Sarcasm); Sarcasm as means of conveying emotions (Type-3 Sarcasm); Sarcasm as a form of written expression (Type-4 Sarcasm). For detecting all the form of sarcasm, SCUBA [24] framework is proposed. For Type-1 sarcasm, not only does the framework find out the contrast in sentiments of phrases in a tweet, but it also compares the change in sentiments with respect to the past tweets. Standardized readability tests were employed to detect Type-2 sarcasm. The scores obtained were compared to the scores of the previous tweets. Type-3 sarcasm is categorized as - Mood based sarcasm and Affect/Sentiment based sarcasm. For mood-based sarcasm the mood of the user is gauged within a certain timeframe, with a sudden change signifying sarcasm; whereas the Affect/Sentiment based sarcasm is detected using the same method used for type-1 sarcasm. Type-4 sarcasm makes an assumption about the language skills of the user. It says that users who are sarcastic also tend to have a good control over the language. They tend to use #not or #sarcasm to denote sarcasm. Moreover, a change in tone and structural variation also denotes sarcasm.

Unlike the methods suggested in [24]; David Bamman and Noah A. Smith [25] considered some more information about the entities like author, relationship among author and their audience and conversational context, content of tweet. While analysing Author information, the features such as Author historical salient terms, Author historical topics, Profile information, Author historical sentiment, and Profile unigrams are taken into consideration. Author historical topics, Author historical salient terms, Profile unigrams, Profile information, Author/Addressee interactional topics, and Chronological interaction between author and addressee are captured while identifying audience features.

They also captured Pairwise Brown features between the original message and the response, Unigram features of the original message for modelling the contextual information of conversation.

945

Word unigrams and bigrams, Brown cluster unigrams bigrams, Unlabelled dependency bigrams, lexicalized and Brown clusters, Part-of-speech features, Pronunciation features, Capitalization features, Tweet whole sentiment, and Tweet word sentiment, Intensifiers are the features those are considered while content of the tweet is analysed.

Debanjan Ghosh and Weiwei Guo and Smaranda Muresan [26] framed sarcasm detection task, known as Literal/Sarcastic Sense Disambiguation (LSSD), that identifies the sense of individual word as sarcastic or literal. The following two challenges were handled in this paper- Bagging clique of objective words; and automatically detecting whether the target word is taken on face value or as sarcastic depending on context used. While handling the first issue, opposite phrases were detected using a crowdsourcing and unsupervised methods. For the latter one addressed, they used several distributional semantic approaches and employed Support Vector Machines (SVM) [17] classifier with a modified kernel using word-embedding to achieve better results. Muresan et al. managed to extract contextual information from certain keywords, however they were limited to the current tweet only. Their work could have been expanded to deal with tweets written in the past. Clues revealed from the past tweets could have helped improve the efficiency of their sarcasm detection technique.

Most of the papers we have studied so far concentrate on modelling features from a single tweet, which bound the performance of the algorithms. The techniques discussed in [43] try to leverage the contextual information about the author of the tweet to detect sarcasm more efficiently. The Tweets were sieved through torrents of posts. This allowed a wider context to be considered. Three types of contextual information were considered – History, Conversation and Topic based Context. Two feature engineering methods were used – Bag of Words [28] and Word Clusters – to model the features. These features are then employed to perform sequential classification using the SVMhnn [43] algorithm. Experimental Results confirmed that sequential classification efficiently detected the contextual information and was able to show an increment in the performance of the sarcasm detection algorithm. Wang et al. did not consider the personality traits of the author as viable contextual information.

Shanthibala et. al. [29] works on a framework that records user behavior and identifies personality traits along with context information. In the first phase, they investigated different personality traits by analyzing the user tweet pattern, and classified them into clusters based on their behavior. While identifying the user behavior following areas such as Principle of Inferability, Emphasis on Specific word, Familiarity of using Social media platforms, Follower and following aspects are taken into consideration. The Second phase deals with context evaluation where different opinions are examined with respect to the users. Initially, all the users are assigned a score zero, and a score of +1 is assigned if their tweet contains any sarcastic emotions. If this score surmounts a certain threshold, that particular user is labelled as 'sarcastic'.

Lukin & Walker [30] presented a pattern-based approach that automatically identifies sarcastic and nastiness patterns on unannotated online dialogues. A high precision sarcastic post classifier, followed by a high precision non-sarcastic post classifier is trained using bootstrapping [30] method.

B. Deep Learning Based Methods

Joshi et al. [31] focused on the subtle contextual information which does not fit in with its surroundings. These contextual oddities that have not been captured by previous methods have proven to be critical for the detection of sarcasm in texts. This technique [31] uses the similarity between word-embedding in order to detect sarcasm. What sets this method [31] apart from its peers is that it does not neglect the findings of its predecessors, rather it compliments them by adding the features based on the similarity of the word-embedding. The word embedding similarity is calculated using 2 methods – unweighted similarity features (UWS) and weighted similarity features (WS). This method [31] takes into account four types of word embeddings – LSA, GloVe, Dependency Weights and Word2Vec.

Silvio Amir et al. [2016] [32] highlighted the importance of taking the contextual features into account in addition to the lexical and syntactical features. The technique discussed [32] suggested that sarcasm is a trait that may be embedded in the user, and hence considering the user characteristics would help in detecting sarcasm. The user-embeddings were augmented along with the lexical and syntactical rules in order to detect sarcasm. Silvio Amir et al. [32] didn't spend much time in feature engineering for the extraction of the user-traits. The paper uses a CNN [33] in order to obtain the contextual features. The past tweets of the user are simply fed to the CNN[33], which learns the user characteristics, which is then in turn augmented along with the lexical and syntactical information.

Ghosh and Veale [2016] [34] assimilated CNN [33], LSTM [35] and a DNN [36] in order to sense sarcasm in text. A recursive SVM [17] was also developed [34] that was fed labelled syntactic and semantic information. The results generated by the two models [34] were then compared. In the neural network, the text is taken as input and converted into a vector. This vector is then fed into a CNN [33], which reduces the frequency variation and identifies the various discriminating words. These words are then fed into the LSTM [35] – a RNN [37] capable of extracting temporal contextual information.

The output of the LSTM [35] is then fed into a DNN [36] layer, which generates a high order feature set as output. This high order feature set is then fed into a SOFTMAX [38] layer that performs the final classification. They concluded that Neural Network Model [39] outperforms well rather than SVM Model [17].

C. Big-Data Based Approach

Due to massive growth of tweet data, Bharti et. al. [40] used a Hadoop based framework where Real time tweets are captured and processed with Apache Flume and Apache Hive in the Hadoop framework. Then, a clique of algorithms is proposed to detect sarcasm in the Hadoop framework as well as non-Hadoop framework.

They employed several approaches like Parsing based lexicon generation algorithm, Interjection word start, Positive sentiment with antonym pair, Tweets which contradicts universal facts, Tweet contradicting time dependent facts, and Likes-dislikes contradiction.

The last three approaches can be implemented using Hadoop based framework after obtaining suitable data sets, which provides insights about future work of this paper.

D. Multi Ensemble Learning Approach

Liu et. al. [27] presented a method which was different from others because it first highlighted the class imbalance distribution problem of sarcastic and non-sarcastic phrases. The physiognomies of both English and Chinese sarcastic sentences and a clique of features for both of the languages are explored. For English sentences, following features like punctuation marks, lexical features, Syntactic features and Semantic imbalance rate are considered. For Chinese language following factors such as Rhetorical feature, Homophony feature, and Construction feature are included as a feature set. A novel multi-strategy ensemble learning approach (MSELA) which assimilates sample ensemble strategy, classifier-ensemble strategy and weighted voting strategy to eliminate sarcasm class imbalance distribution problem is introduced in this paper. The Sample-ensemble strategy employs weighted random sampling which eliminates class imbalanced problem. Classifier-ensemble strategy incorporates numerous classification procedures such as Naive Bayes, SVM and Maximum Entropy. Finally, the weighted voting strategy is employed which considers both the accuracy of different classification algorithms and the importance of sample subsets, to achieve fairer weighted voting and higher prediction accuracy. They also established that MSELA outperforms well in average case rather than other ensemble methods like FullT, Bagging, AdaBoost [41], Random Forest [42], EUM, and Logistic Regression [18].

III. COMPARATIVE ANALYSIS

Table 1 presents a comparative analysis of the existing sarcasm detection algorithms. All of the algorithms were applied on the twitter data set. Experimental results show that techniques that take into account the contextual information do a much better job than the ones who don't. Experimental results also show a huge amount of untapped potential in Deep Learning-based methods. They can be improved upon to increase the efficiency and accuracy of the system.

Table 1. Comparative Analysis of Sarcasm Detection Techniques

Technique Used	Paper Reference	Performance Measure
NLP Based Method	[7]	Accuracy: 71%
	[4]	AUC: 0.79
	[8]	F – Score: 0.51
	[10]	Accuracy: 83.1%
	[11]	Accuracy: 79.43%
	[13]	F-Score: 0.827
	[14]	F-Score: 0.827 (Twitter) 0.766 (Amazon)
	[16]	F-Score: 0.97
	[19]	F-Score: 0.95
	[24]	Accuracy: 83.46%
	[25]	Accuracy: 81.2%
	[26]	F1-Score: 76.1%
	[30]	Accuracy: 68.7%
Deep Learning Based Methods	[31]	LSA: 80.06%
		GloVe: 80.26%
		DW: 81.19%
		Word2Vec: 81.20%
	[32]	Accuracy: 87.2%
	[34]	F-Score: 0.921
Big Data-Based Approach	[40]	F-Score: 0.90
Multi-Ensemble Learning Approach	[27]	AUC Score: 0.840

IV. PROPOSED FRAMEWORK

Our proposed framework not only aims to recognize

sarcastic tweets but also sarcastic users. The model consists of two phases – in phase one a Tweet is classified as either Sarcastic or Non-Sarcastic, and in phase two the output of the previous phase is used to identify sarcastic users.

The working in the two phases is described below in great details:

Proposed Method - Detecting Sarcasm within a tweet

This phase analyses the tweet content to identify sarcastic clues. The entire working in this phase can be divided into multiple stages:

- **Data Collection**: The tweets are collected from Twitter using Twitter API.
- **Data Preprocessing**: The collected tweets are pre-processed by eliminating stop-words, URLs, etc. present in the tweet content.
- **Feature Extraction**: The syntactic, semantic, lexical features are extracted and a feature vector is formed.
- **Content Analysis in RNN-1**: The feature vector is then fed into a Recurrent Neural Network (RNN) that provides content analysis of a given tweet.
- **Score Generation by RNN-2**: The output of the previous Recurrent Neural Network, is fed into another RNN that considers interfering factors such as History of Conversation and Topical Context. Then imposing some score calculation function, a score is obtained for each tweet. If that score goes beyond a predefined threshold, that tweet is labelled as sarcastic.

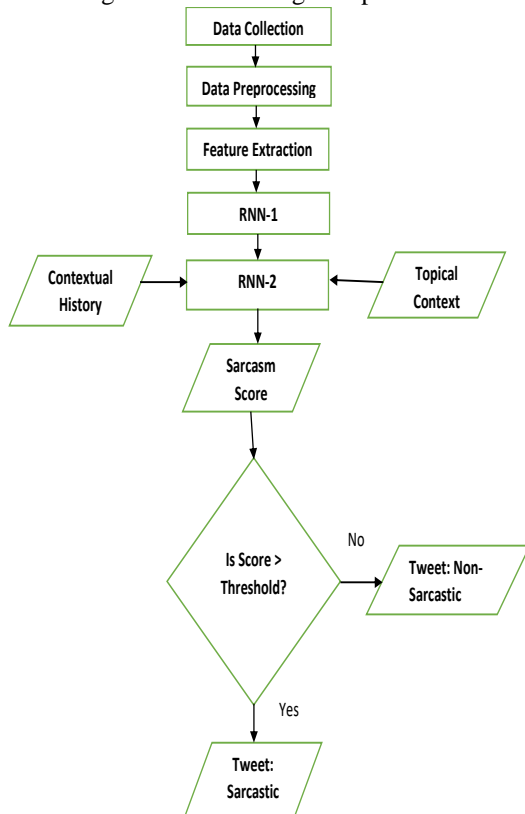The flowchart shown in figure II below displays the flow of control amongst the various stages in phase 1.

Figure 2. Flowchart for Phase-1

Phase 2 - Identifying sarcastic user

In this case, we assume that a user can be regarded as sarcastic if most of his/her tweets are sarcastic. For a user, if some tweets contain sarcasm those particular tweets can be regarded as sarcastic. But, for identifying a user as sarcastic, all of his/her tweets are taken into consideration. These tweets are then fed into the phase-1 of our framework as input, and are labelled as either sarcastic or non-sarcastic. Next, the tweet scores generated in phase-1 are averaged and a user score is obtained. If that score is greater than a predefined value, that user will be regarded as sarcastic.

V. CONCLUSION

This paper provided an exhaustive discourse on the existing sarcasm detection methodologies. The existing techniques were roughly divided into three categories – Feature Engineering based methods, Deep Learning based techniques, and Big Data based approach. The feature engineering-based methods involve a study of the target text and the extraction of the lexical clues, hyperboles and pragmatic features. These features were then employed to perform the final classification using either a supervised or semi-supervised paradigm. The feature engineering methods are significantly popular but costly, and they may miss out on certain sarcasm clues, if done manually. To overcome these problems, neural-network based deep learning approaches were preferred. These approaches made the use of CNNs and RNNs for feature extraction, which were then later fed into a SOFTMAX layer for final classification. Feature engineering methods were then extended into distributed system, in order to deal with the huge voluminous structure of tweet data.

Based on our understanding of researches pursued in this field, the following areas can be earmarked for future work and can be employed to extend our framework:

1. Language Independent Sarcasm Detection: Many authors as in [27][4] followed a specific culture-based sarcasm detection technique which can be further improved by incorporating several languages, thus making the sarcasm detection tool language independent.

2. Developing better unsupervised approaches: Many researchers have always preferred working with either supervised or semi-supervised systems. This is because it is very difficult to differentiate between sarcastic and non-sarcastic comments without any external help. An unsupervised method will work only with the use of specific hashtags – which has its own set of drawbacks. Having a better unsupervised approach can help to obtain a highly efficient sarcasm detection engine.

3. Improving upon the existing deep learning paradigms- Deep learning-based sarcasm detection field can be still utilized and extended while designing an effective sarcastic text identifier tool.

4. Implementing them in a distributed framework: the utter volume of textual data available on the internet makes it compulsory for the development of a highly efficient and distributed scheme adept of detecting sarcasm.

REFERENCES

1. M. Bouazizi and T. Otsuki, "A Pattern-Based Approach for Sarcasm Detection on Twitter," IEEE Access, vol. 4, pp. 5477–5488, 2016, doi: 10.1109/ACCESS.2016.2594194.
2. R. Giora, "On Irony and Negation," Discourse Process., vol. 19, no. 2, pp. 239–264, 1995, doi: 10.1080/01638539509544916.
3. S. L. Ivanko and P. M. Pexman, "Context Incongruity and Irony Processing," Discourse Process., vol. 35, no. 3, pp. 241–279, 2003, doi: 10.1207/s15326950dp3503_2.
4. C. Liebrecht, "The perfect solution for detecting sarcasm in tweets # not," no. June, pp. 29–37, 2013.
5. A. U. Hassan, J. Hussain, M. Hussain, M. Sadiq, and S. Lee, "Sentiment analysis of social networking sites (SNS) data using machine learning approach for the measurement of depression," Int. Conf. Inf. Commun. Technol. Converg. ICT Converg. Technol. Lead. Fourth Ind. Revolution, ICTC 2017, vol. 2017-Decem, pp. 138–140, 2017, doi: 10.1109/ICTC.2017.8190959.
6. E. Sulis, D. Irazú Hernández Farías, P. Rosso, V. Patti, and G. Ruffo, "Figurative messages and affect in Twitter: Differences between #irony, #sarcasm and #not," Knowledge-Based Syst., vol. 108, pp. 132–143, 2016, doi: 10.1016/j.knosys.2016.05.035.
7. R. J. Kreuz and G. M. Caucci, "Lexical influences on the perception of sarcasm," pp. 1–4, 2007, doi: 10.3115/1611528.1611529.
8. E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as Contrast between a Positive Sentiment and Negative Situation," no. Emnlp, 2013.
9. A. Tripathy, A. Agrawal, and S. K. Rath, "Classification of sentiment reviews using n-gram machine learning approach," Expert Syst. Appl., vol. 57, no. October 2017, pp. 117–126, 2016, doi: 10.1016/j.eswa.2016.03.028.
10. D. Maynard and M. A. Greenwood, "Who cares about sarcastic tweets ? Investigating the impact of sarcasm on sentiment analysis."
11. P. Tungthamthiti, K. Shirai, and M. Mohd, "Recognition of Sarcasm in Tweets Based on Concept Level Sentiment Analysis and Supervised Learning Approaches," pp. 404–413, 2014.
12. B. T. Pham, D. Tien Bui, I. Prakash, L. H. Nguyen, and M. B. Dholakia, "A comparative study of sequential minimal optimization-based support vector machines, vote feature intervals, and logistic regression in landslide susceptibility assessment using GIS," Environ. Earth Sci., vol. 76, no. 10, 2017, doi: 10.1007/s12665-017-6689-3.
13. O. Tsur and A. Rappoport, "ICWSM – A Great Catchy Name : Semi-Supervised Recognition of Sarcastic Sentences in Online Product Reviews," no. 9, pp. 162–169.
14. D. Davidov and O. Tsur, "Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon," no. July, pp. 107–116, 2010.
15. G. Song, J. Rochas, L. E. El Beze, F. Huet, and F. Magoulès, "K Nearest Neighbour Joins for Big Data on MapReduce: A Theoretical and Experimental Analysis," IEEE Trans. Knowl. Data Eng., vol. 28, no. 9, pp. 2376–2392, 2016, doi: 10.1109/TKDE.2016.2562627.
16. R. González-ibáñez and N. Wacholder, "Identifying Sarcasm in Twitter : A Closer Look," no. 2010, pp. 581–586, 2011.
17. F. Luo, C. Li, and Z. Cao, "Affective-feature-based sentiment analysis using SVM classifier," Proc. 2016 IEEE 20th Int. Conf. Comput. Support. Coop. Work Des. CSCWD 2016, pp. 276–281, 2016, doi: 10.1109/CSCWD.2016.7566001.
18. M. J. L. F. Cruyff, U. Böckenholt, P. G. M. van der Heijden, and L. E. Frank, "A Review of Regression Procedures for Randomized Response Data, Including Univariate and Multivariate Logistic Regression, the Proportional Odds Model and Item Response Model, and Self-Protective Responses," Handb. Stat., vol. 34, pp. 287–315, 2016, doi: 10.1016/bs.host.2016.01.016.
19. F. Barbieri, H. Saggion, and F. Ronzano, "Modelling Sarcasm in Twitter , a Novel Approach," pp. 50–58, 2014.
20. D. M. Magerman, "Statistical decision-tree models for parsing," pp. 276–283, 1995, doi: 10.3115/981658.981695.
21. D. K. Bharti, K. S. Babu, and S. K. Jena, "Parsing-based Sarcasm Sentiment Recognition in Twitter Data," no. August, 2015, doi: 10.1145/2808797.2808910.
22. M. J. Collins, "A new statistical parser based on bigram lexical dependencies," pp. 184–191, 1996, doi: 10.3115/981863.981888.
23. S. Martin, J. Liermann, and H. Ney, "Algorithms for bigram and trigram word clustering," Speech Commun., vol. 24, no. 1, pp. 19–37, 1998, doi: 10.1016/S0167-6393(97)00062-9.
24. A. Rajadesingan, R. Zafarani, and H. Liu, "Sarcasm Detection on Twitter," pp. 97–106, 2015, doi: 10.1145/2684822.2685316.
25. D. Bamman and N. A. Smith, "Contextualized sarcasm detection on twitter," Proc. 9th Int. Conf. Web Soc. Media, ICWSM 2015, pp. 574–577, 2015.
26. D. Ghosh, W. Guo, and S. Muresan, "Sarcastic or Not : Word Embeddings to Predict the Literal or Sarcastic Meaning of Words," no. September, pp. 1003–1012, 2015.
27. P. Liu, W. Chen, G. Ou, T. Wang, D. Yang, and K. Lei, "Sarcasm Detection in Social Media Based on Imbalanced Classification," pp. 459–471, 2014.
28. R. Shekhar and C. V. Jawahar, "Word image retrieval using bag of visual words," Proc. - 10th IAPR Int. Work. Doc. Anal. Syst. DAS 2012, no. March, pp. 297–301, 2012, doi: 10.1109/DAS.2012.96.
29. K. Rajeswari and P. Shanthibala, "SARCASM DETECTION USING MACHINE LEARNING TECHNIQUES," vol. 9, no. L, pp. 26368–26372, 2018, doi: 10.24327/IJRSR.
30. R. Justo, T. Corcoran, S. M. Lukin, M. Walker, and M. I. Torres, "Extracting relevant knowledge for the detection of sarcasm and nastiness in the social web," Knowledge-Based Syst., vol. 69, no. 1, pp. 124–133, 2014, doi: 10.1016/j.knosys.2014.05.021.
31. A. Joshi, V. Tripathi, K. Patel, P. Bhattacharyya, and M. Carman, "Are Word Embedding-based Features Useful for Sarcasm Detection ?," no. 2013, pp. 1006–1011, 2016.
32. S. Amir, B. C. Wallace, P. Carvalho, and J. Silva, "Modelling Context with User Embeddings for Sarcasm Detection in Social Media," pp. 167–177, 2016.
33. R. Girshick, "Fast R-CNN," Proc. IEEE Int. Conf. Comput. Vis., vol. 2015 Inter, pp. 1440–1448, 2015, doi: 10.1109/ICCV.2015.169.
34. A. Ghosh and T. Veale, "Fracking Sarcasm using Neural Network Fracking Sarcasm using Neural Network," no. June, 2016, doi: 10.13140/RG.2.2.16560.15363.
35. X. Shi, Z. Chen, and H. Wang, "Convolutional LSTM Network," Nips, pp. 2–3, 2015, doi: [].
36. J. Li, R. Zhao, J. T. Huang, and Y. Gong, "Learning small-size DNN with output-distribution-based criteria," Proc. Annu. Conf. Int. Speech Commun. Assoc. INTERSPEECH, no. September, pp. 1910–1914, 2014.
37. J. Koutn and K. Greff, "A Clockwork RNN," 1989.
38. E. Jang, S. Gu, and B. Poole, "Categorical reparameterization with gumbel-softmax," 5th Int. Conf. Learn. Represent. ICLR 2017 - Conf. Track Proc., pp. 1–13, 2019.
39. A. K. Yadav and S. S. Chandel, "Solar energy potential assessment of western Himalayan Indian state of Himachal Pradesh using J48 algorithm of WEKA in ANN based prediction model," Renew. Energy, vol. 75, pp. 675–693, 2015, doi: 10.1016/j.renene.2014.10.046.
40. S. K. Bharti, B. Vachha, R. K. Pradhan, K. S. Babu, and S. K. Jena, "Sarcastic sentiment detection in tweets streamed in real time: a big data approach," Digit. Commun. Networks, vol. 2, no. 3, pp. 108–121, 2016, doi: 10.1016/j.dcan.2016.06.002.
41. G. Haixiang, L. Yijing, L. Yanan, L. Xiao, and L. Jinling, "BPSO-Adaboost-KNN ensemble learning algorithm for multi-class imbalanced data classification," Eng. Appl. Artif. Intell., vol. 49, no. October, pp. 176–193, 2016, doi: 10.1016/j.engappai.2015.09.011.
42. M. Belgiu and L. Drăgu, "Random forest in remote sensing: A review of applications and future directions," ISPRS J. Photogramm. Remote Sens., vol. 114, pp. 24–31, 2016, doi: 10.1016/j.isprsjprs.2016.01.011.
43. Wang, Z., Wu, Z., Wang, R., & Ren, Y. (2015, November). Twitter sarcasm detection exploiting a context-based model. In international conference on web information systems engineering (pp. 77-91). Springer, Cham.

AUTHORS PROFILE

Shawni Dutta was born in Kolkata, India, in 1995. She completed her M.Sc. in Computer Science at St. Xavier's College Kolkata (Autonomous). Currently she is working as a College Whole Time Teacher at The Bhawanipur Education Society College Kolkata. Her research interests include Image Processing, Natural Language Processing, Cryptography and Machine Learning. She received eminent Academic Awards as well as Social Awards. She published a number of papers in the International Journals.

Akash Mehta was born in Kolkata, India, in 1992. He completed his M.Tech. in CSE at the University of Calcutta, Department of Computer Science and Engineering, in 2017. He completed his M.Sc. in Computer and Information Science, at the University of Calcutta, Department of Computer Science and Engineering, in 2015. Currently he is working as a College Whole Time Teacher at The Bhawanipur Education Society College Kolkata. His research interests include Image Processing, Deep Learning, Parallel Processing and Computer Vision.

Prof. (Dr.) Samir Kumar Bandyopadhyay is presently designated as Academic Advisor, The Bhawanipur Education Society College Kolkata. He is also Senior Professor in the Department of Computer Science and Engineering of University of Calcutta. He did his Bachelor and Master Degree in Engineering Degree and Medical Science from University of Calcutta. Prof. Bandyopadhyay did three Ph.D. in Medical Science. First he did his Ph.D. in Cardiology from University of Calcutta. Prof. Bandyopadhyay also did his second Ph.d. on Breast Cancer from Bristol University, UK. Finally, he obtained his Ph.D. degree in Brain Tumor from MIT, USA. He served as Registrar, Vice-Chancellor of State University in India. His fields of research interest are: Biomedical Image Processing, Biomedical Engineering, Mobile Computing, Pattern Recognition, Graph Theory, Image Processing, Handwritten Signature Verification, Graphical Password Verification, Steganography and Watermarking, Medical Imaging, Forensic Science, Crime Science