An Approach to Detect Sarcasm in Tweets

Jyoti Godara and Rajni Aron

March 23, 2020

An approach to detect Sarcasm in Tweets

Jyoti Godara[1] and Rajni Aron[2]

[1] Lovely Professional Universty, Punjab, India
[2] Lovely Professional University,Punjab, India
{jyotipoonia6,rajni3387}@gmail.com

Abstract. When academic and business ventures are discussed, electronic documents form the crucial part of receiving and transferring information. There is no use of online information if we cannot extract it and use it to cater our ventures. In order to frame up any summary, it is required to find the relevant text with complete omission of unnecessary information while keeping the focus on details and compile them into a document. The sentiment analysis is the approach used to evaluate users ' sentiments on websites, forums, comments, feedback as negative, positive or neutral. But, sometimes, people express their negative sentiment in a positive manner. This flips the polarity of the sentence and sentiment analysis performance is affected. Thus, detection of sarcasm is an important part of sentiment analysis. Input data features are extracted and data needs to be classified as sarcastic or not. To increase accuracy for the sarcasm detection from twitter data new features needs to add fort the training. This paper proposed to develop an ensemble classification method having base classifiers as Decision Tree, Naive Bayes and K-nearest Neighbor to increase various parametric values for the sarcasm detection.

Keywords: Sentiment Analysis · Sarcasm · Ensemble Classifier· Machine Learning.

1 Introduction

Internet era has changed the way individuals communicate their views and opinions. Millions of people use social networking sites to share thoughts, emotions, and views about their daily life [15]. There are around 823 new tweets posted on Twitter every second, 520 comments every minute ,294000 statuses and 138000 snapshots posted on Facebook and more than 1 million customer transactions per day are manged by Walmart [5]. Such huge data can be valuable for various applications such as election outcome forecast, analysis of market trend, e-governance, business intelligence as customers rely heavily on the online content for the decision making. This voluminous content is too high to be processed by human, thus it needs to be automated. This task of creating a system for gathering and analysing the sentiment about an entity's sentiment in the forum, comments, feed backs or tweets is called sentiment analysis.

It analyzes people's views, thoughts, emotions and perceptions towards things such as goods, services, organisations, person, issues, events, topics and their attributes . It can be applied at three different levels- document level, sentence level and aspect level [16]. From the sentiments expressed, we can easily express what the speaker has in his mind. Sentiments may be like- dislike, agree-disagree, good-bad, positive or negative sentiments or multiclass sentiments- strong positive, positive, neutral, negative, strong negative. Sometimes, analysis of sentiment can be misled by the use of terms which are of strong polarity but are used sarcastically, implying that the opposite polarity was intended.

Sarcasm is a specific type of sentiment where an individual conveys his negative feelings using positive exaggerated words in the text [**?**]. Example- "I love music, it can take you to another Place, for example, Meghan Trainor is playing in this cafe so now I am going to a different café." "Oh! He is out on a duck, what a legendry batsman." [21].In the example, person is expressing positive emotion but overall tweet is reflecting negativity towards the batsman. It is a very elegant way to indirectly convey message which makes it difficult to detect [7].

From business perspective, it is very crucial to understand product reviews, movie popularity and social opinions, as they may be suffered, if considered in the wrong category '[17]. Detection of sarcastic text will refine the performance of sentiment analysis system to make it more reliable and accurate as it will help to eliminate the intentional ambiguity raised in the text [19]. Detection of sarcasm while analysing the sentiment in the given text is a type of classification task. Various approaches for detection have been categorized into machine learning and lexicon based approach [15]. Machine learning approach initiates with the collection of training data sample. Then, a classifier on the training information is trained. After the selection of a classification method, a significant conclusion is made. Machine learning technique engages both supervised learning for labelled dataset and unsupervised learning for the unlabelled dataset. Few classifiers using supervised learning approach are Decision Tree, Neural Network, Support Vector Machine, Naïve Bayes, Maximum Entropy, while the unsupervised approach are clustering methods. Lexicon-based approach [10] uses the dictionary of sentiments with the terms of opinion and combines them with the data for polarity determinations. This assigns sentiment scores to the words of opinion explaining whether the terms in the dictionary are optimistic, negative and objective . Dictionary based lexical approach includes looking for synonyms or antonyms of the dictionary words while Corpus based approach provides dictionaries related to specific domains.

Ensemble approaches in machine learning puts together the effect of several machine learning methods on the problem to obtain a greater predictive ability in solitude than its comprising algorithms [14]. The overarching objective of this approach is to combine outputs of base classifiers, in order to produce a single aggregated output which exceeds any of the isolated base classifiers. The first stage in the classifier ensemble process is to produce various base classifiers. One strategy is to apply N learning technique, with a single collection of training data, to obtain N different models of classification . An alternative approach is to build N datasets from the training data and use single learning algorithm for each set. [13]. Bagging is the technique of development of multiple training datasets from a single dataset.. Boosting applies different datasets and same learning method. Stacking is used to combine various models which were built using different learning models on single data.

In proposed method, ensemble classifier will be generated from the machine learning classifiers to improve the accuracy for detecting sarcasm.

The following sections are organized as follows: We conducted literature surveys in Section II. It gives an overview of research undertaken in this field. We also outlined an approach in Section III. The analysis finishes in Section IV.

2 Literature Review

This section includes existing literature related to the sentiment analysis, ensemble classifier and detection of sarcasm in the textual data.

Rui Ren, et.al (2019)integrated sentiment analysis using SVM classifier into a machine learning framework [20].]. In addition, more accurate and practical sentiment indexes were generated by taking into account the day-to-week impact. It was possible to improve the prediction accuracy of movement direction of the SSE 50 Index up to 89.93% as per the obtained experimental results. A growth of 18.6% was seen in the accuracy rate after including sentiment variables. In the meantime, the proposed model also provided support to the shareholders in efficient decision making.

Farkhund Iqbal, et.al (2019) proposed a new feature reduction approach based on Genetic Algorithm (GA) [2]. This hybrid technique had the ability of reducing the dimension of feature-set up to 42% without bothering accuracy. The proposed feature reduction approach was compared with more extensively used feature lessening algorithms based on Principal Component Analysis (PCA) and Latent Semantic Analysis (LSA). The proposed approach showed improved accuracy rate of 15.4% and 40.2% over PCA and LSA based approaches respectively.

Nora Al-Twairesh, et.al (2019) made an attempt to discover about sentiment-particular word embeddings from Arabic tweets [1]. These word embeddings were later used for classifying the sentiments of Arabic Tweets. Furthermore, a novel feature which was an ensemble model of surface and deep features was recommended in this work. The surface features were retrieved physically. On the other hand, the deep features were generic word embeddings and sentiment-specific word embeddings. A lot of tests were carried out for testing the efficiency of ensembling surface and deep features. These tests also examined pooling functions, embeddings dimension, and cross-dataset models.

Xianghua, et.al (2018) recommended a lexicon-enhanced LSTM model. Initially, This model used sentiment lexicon as additional pre-training information for a word sentiment classifier [9]. Afterward, this classifier got the sentiment embeddings of words. These embeddings included non-lexicon words. It was possible to represent words more accurately by combining the sentiment embedding and its word embedding. Also, a novel technique was described in this work for finding the attention vector in wide-ranging sentiment analysis devoid of any target. This phenomenon could enhance the LSTM ability for getting inclusive information of sentiment. The tested outcomes depicted that the recommended models outperformed the other accessible models.

Liang-Chih Yu, et.al (2018) recommended a word vector refinement model. Refining existing pretrained word vectors was the main aim of this model [22]. For this purpose, this model used real-valued sentiment intensity scores given by sentiment lexicons. The concept of the recommended model was to improve every word vector for bringing it nearer to both semantically and sentimentally alike words and more far from sentimentally unlike words in lexicon. The tested outcomes revealed that the suggested model could improve both traditional word embeddings and already suggested sentiment embeddings for classifying binary, ternary, and fine-grained sentiments. Two data sets called SemEval and Stanford Sentiment Treebank were used in this work to perform tests.

Ahmet Hayran, et.al (2017) suggested a new scheme to classify the sentiments microblog posts in automatic manner [12]. The recommended scheme was based on strong feature demonstration and combination. This work used word embedding method as the feature depiction. On the other hand, Support Vector Machine was used as a classification model. The tested outcomes depicted that that the proposed technique efficiently reduced the size of tweet depiction and enhanced the precision of sentiment classification. The suggested approach showed accuracy rate of 80.05% in the classification of sentiments and proved its supremacy over other existing techniques.

Nadia F.F da Silva, et.al (2014) generated a single classifier from multiple classifiers- Naïve Bayes, Maximum Entropy and Support Vector Machine to improve the accuracy of tweet classification [6].

Isidoros Perikos, et.al (2016) proposed a sentiment analysis method for automated emotion detection in a text using classifier ensembles [18].Ensemble classifier was based on three key classifiers-a learner from Naïve Bayes, a maximum entropy and knowledge-based system. Results showed that ensemble schema performed better in recognizing emotion present in the text and determining text polarity than the sole classifiers. Pornima Gidhe, et.al (2017) stated that a lot of researches were conducted on dataset including # sarcasm tag. These researchers mostly used structural and sentiment attributes. However, some cases did not include # sarcasm tags [11]. Emotional and semantic similarity features were needed to transcribe these types of cases. In this work, a novel scheme had been recommended for the classification of sarcastic and non-sarcastic sentences. For this purpose, structural, emotional and semantic similarity features with MLP-BP approach were used in this work.

Paras Dharwal, et.al (2017) stated that it was a very complex task to detect sarcasm in automatic manner in Sentiment Analysis [8]. The detection task included composite linguistic analysis and machine learning algorithms. In this work, a lot of sarcasm analysis techniques were reviewed. These techniques were used to filter the sarcastic statements from a text. These techniques also analyzed the application of Automatic sarcasm detection in the classification of product review texts and tweets.

Santosh Kumar Bharti, et.al (2017) stated that several sarcasm detection approaches had been recommended for detecting sarcastic within text [3]. Detecting sarcasm in English text was the main aim of almost all techniques. A novel context-based pattern had been recommended in this work to detect sarcasm within Hindi tweets. In the proposed pattern, sarcasm was represented as a conflict between a tweet and the reference of its relevant news. The suggested technique used Hindi news as the reference of a tweet within the similar timestamp and achieved 87% accuracy rate.

Shubhadeep Mukherjee, et.al (2017) considered different text independent feature sets.This kind of features included n-grams and function words. [17]. In this work, Naïve Bayes and fuzzy clustering algorithms were used for the testing of several feature sets. The tested results depicted that it was advantageous to add some features capturing the blogging style of the microblog writers to detect sarcasm. Accuracy rate of about 65% was achieved by the recommended approach in this work.

Mondher Bouazizi, et.al (2016) proposed a novel scheme based on pattern for twitter sarcasm detection [4]. In this work, four feature sets had been utilized. These feature sets covered the different sorts of identified sarcasm. These feature sets were utilized for classifying tweets as sarcastic and non-sarcastic. The recommended scheme showed accuracy and precision rate of 83.1% and 91.1% respectively. This work studied the significance of every suggested feature set. This work also evaluated its additional value to the classification.

As per the existing research work, we have analysed that hybrid classifier is performing better than standalone classifiers , thus an ensemble classifier is proposed to increase the accuracy to detect sarcasm.

3 Proposed Methodology

This article proposes a prototype for detecting sarcasm during sentiment analysis of tweets. The basic steps are also described in Fig 1.

3.1 Data Collection

The data sets needed for the experiments will be obtained from Twitter's official website using appropriate APIs and programs in the form of tweets — a micro blogging network that allows its users to write up to 140 characters. Tweets of #sarcasm or #sarcastic or #not of a certain length are taken as sarcastic datasets.It also extracts an appropriate number of regular tweets that have positive or negative sentiment scores. These corpora would then be used at a later stage to train the systems.

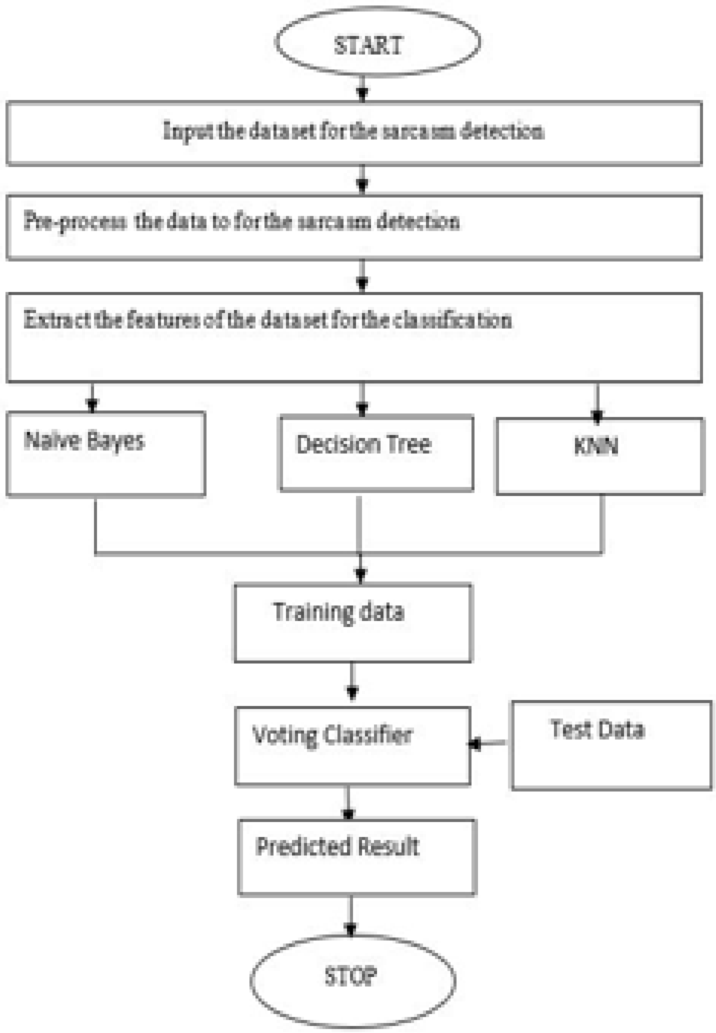

Fig. 1. Proposed Methodology

3.2 Data Pre-processing

The collected dataset will be then pre-processed to be converted into computer readable format. Following steps will be used:

- DataCleaning:URLs, @mention, retweets, hashtags, ampersands, extra white spaces,hexacharacters, double quotations, emoticons, numbers, and lines are eliminated. All uppercase letters are transformed to lowercase cases.Redundant tweets are removed.Tweets of fewer than three words are also deleted.
- Tokenization: Word sequences are broken down to symbols, phrases, words, or other elements that are useful and are known as tokens.
- POS (Parts Of Speech) tagging: It is used to extract important words which have a greater impact on the classification of emotions, such as adjective, adverb, verb, noun. Each word is be assigned a POS tagger.
- Removal of stop words: Stop Words such as the, in, at, that, which, and on. are removed as they will increase the work required by the software to parse them, while providing minimal benefit.
- Stemming and Lemmatization: Stemming reduces the derived word to its original word stem. Eg. Attending is stemmed to attend. The main objective of this technique is to eliminate included suffixes and amount of words. Lemmatization removes the affixes and also adds the missing character to the root to make the complete word. Eg. Stemming : decided- decid; Lemmatization: decide

3.3 Feature extraction

- Point-wise Mutual Information: The measure of mutual information provides a traditional way of modeling the mutual information between the features and classes.
- Chi-square and PMI: These are two distinct ways to measure the association between words and groups.
- Punctuation marks:Punctuation includes, ; : ? !, emoticons, quotes, capitalized words, etc. In certain cases, Sarcastic sentences show a certain pattern. Punctuation marks play a significant role in understanding such sarcastic sentences. For example, Wednesday is latin for “almost friday. ”
- Lexical features: These are significant when sarcastic statements are delivered and identified. These can be based on factors such as counter-factuality (e.g., yet), temporal compression (e.g., suddenly), repeated sequences (e.g., n sequences in a sentence-grams, grams skip).
- Semantic characteristics: While some attention has been given to semantic characteristics (e.g., ADV+ADJ+N) and temporal inconsistency (e.g., dislike, not). They are of vital importance in recent investigations to demonstrate inconsistency in the sense of sentences. Sarcasm requires the background to be reliably known.

Those features are selected which are highly co-related to the class.

3.4 Classification

In the proposed methodology, ensemble classifier will be generated by selecting Naïve Bayes, Decision Tree and K nearest Neighbor as baseline classifiers.

- Naive Bayes is a probabilistic classification model. It makes assumption about the strong independence among the features. This classifier needs less number of training data for computing the parameters to carry out prediction task [8]. This is the major benefit of this approach. Due to the independence of features, this classifier computes merely the variance of the feature rather than computing the complete covariance matrix. The conditional probability for each class given a review is P(c) for a text review received 'd ' and for a class ' c ' (positive, negative); It is expressed as:

$$P(c) = \frac{(P(d) * P(c))}{P(d)} \tag{1}$$

- A decision tree resembles a flowchart-like tree structure. In this approach, every interior node depicts a test on a feature and every branch demonstrates the test result. Every leaf node or terminal node represents a class label. The testing of attribute values of the tuple is performed against the decision tree for a given tuple X. The tracing of a route is done from the root to a leaf node. This node holds the class prediction for the tuple [9].
- K-Nearest Neighbors is a very fundamental and important classifier. This classification algorithm is related to the supervised learning approach. This classifier is extremely useful in pattern recognition, data mining and attack revealing. Some prior data known as training data is used to classify coordinate into classes.

Ensemble classifier will be generated by combining the base classifiers using stacking approach. Initially all instances from the validation set are classified by all the base classifiers. A model is built by combining the decisions of base classifier through majority voting technique.

3.5 Evaluation

The ensemble model will be classifying the new test data in the sarcastic or non-sarcastic data. Performance will be evaluated by the confusion matrix. The parameters of accuracy, precision, recall, and F-measures are based on the classification's true positive, true negative, false positive, and false negative outcomes. True Positive: The correctly defined values are considered true positive ones. True Negatives: The attributes that are wrongly placed into a separate class are True negatives. False positives: False positives are the values that do not belong to a certain class but are mistakenly listed in that class. False Negatives: False negatives are the values that neither belonged to a certain class nor were included in that class.

$$Precision = \frac{(TruePositive)}{(TruePositive + FalsePositive)} \tag{2}$$

$$Recall = \frac{(TruePositive)}{(TruePositive + FalseNegative)} \tag{3}$$

$$Fmeasure = \frac{(2 * recall * precision)}{(precision + recall)} \tag{4}$$

$$Accuracy = \frac{(TruePositive + TrueNegative)}{(TruePositive + FalsePositive + TrueNegative + FalseNegative)} \tag{5}$$

4 Conclusion

Sentiment analysis examines the polarity of the sentiments from the data.Due to presence of sarcastic text, the accuracy is affected. Thus, detection of sarcasm in the text is very crucial during analysis of sentiments. In this paper, an approach to detect sarcasm in tweets is proposed by generating an ensemble classifier combining outputs of Naive Bayes, K- nearest Neighbor and Decision Tree using majority voting approach. Performance will be improved using this approach. In further studies, this model can be implemented on product review tweets or other social networking websites. It can also be extended to stock markets, news articles and political discussions.

References

1. Nora Al-Twairesh and Hadeel Al-Negheimish. Surface and deep features ensemble for sentiment analysis of arabic tweets. *IEEE Access*, 7:84122–84131, 2019.
2. T Anuprathibha and CS KanimozhiSelvi. Penguin search optimization based feature selection for automated opinion mining.
3. Santosh Kumar Bharti, Korra Sathya Babu, and Rahul Raman. Context-based sarcasm detection in hindi tweets. In *2017 Ninth International Conference on Advances in Pattern Recognition (ICAPR)*, pages 1–6. IEEE, 2017.
4. Mondher Bouazizi and Tomoaki Otsuki Ohtsuki. A pattern-based approach for sarcasm detection on twitter. *IEEE Access*, 4:5477–5488, 2016.
5. Himadri Tanaya Chidananda, Debashis Das, and Santwana Sagnika. Sentiment analysis using n-gram technique. In *Progress in Computing, Analytics and Networking*, pages 359–367. Springer, 2018.
6. Nadia FF Da Silva, Eduardo R Hruschka, and Estevam R Hruschka Jr. Tweet sentiment analysis with classifier ensembles. *Decision Support Systems*, 66:170–179, 2014.
7. Anandkumar D Dave and Nikita P Desai. A comprehensive study of classification techniques for sarcasm detection on textual data. In *2016 International Conference on Electrical, Electronics, and Optimization Techniques (ICEEOT)*, pages 1985–1991. IEEE, 2016.

8. Paras Dharwal, Tanupriya Choudhury, Rajat Mittal, and Praveen Kumar. Automatic sarcasm detection using feature selection. In *2017 3rd International Conference on Applied and Theoretical Computing and Communication Technology (iCATccT)*, pages 29–34. IEEE, 2017.
9. Xianghua Fu, Jingying Yang, Jianqiang Li, Min Fang, and Huihui Wang. Lexicon-enhanced lstm with attention for general sentiment analysis. *IEEE Access*, 6:71884–71891, 2018.
10. Xingtong Ge, Xiaofang Jin, Bo Miao, Chenming Liu, and Xinyi Wu. Research on the key technology of chinese text sentiment analysis. In *2018 IEEE 9th International Conference on Software Engineering and Service Science (ICSESS)*, pages 395–398. IEEE, 2018.
11. Pornima Gidhe and Leena Ragha. Sarcasm detection of non# tagged statements using mlp-bp. In *2017 International Conference on Advances in Computing, Communication and Control (ICAC3)*, pages 1–4. IEEE, 2017.
12. Ahmet Hayran and Mustafa Sert. Sentiment analysis on microblog data based on word embedding and fusion techniques. In *2017 25th Signal Processing and Communications Applications Conference (SIU)*, pages 1–4. IEEE, 2017.
13. Anna Jurek, Yaxin Bi, Shengli Wu, and Chris Nugent. A survey of commonly used ensemble-based classification techniques. *The Knowledge Engineering Review*, 29(5):551–581, 2014.
14. Monisha Kanakaraj and Ram Mohana Reddy Guddeti. Performance analysis of ensemble methods on twitter sentiment analysis using nlp techniques. In *Proceedings of the 2015 IEEE 9th International Conference on Semantic Computing (IEEE ICSC 2015)*, pages 169–170. IEEE, 2015.
15. Vishal Kharde, Prof Sonawane, et al. Sentiment analysis of twitter data: a survey of techniques. *arXiv preprint arXiv:1601.06971*, 2016.
16. M Kiruthika, Sanjana Woonna, and Priyanka Giri. Sentiment analysis of twitter data. *International Journal of Innovations in Engineering and Technology*, 6(4):264–273, 2016.
17. Shubhadeep Mukherjee and Pradip Kumar Bala. Sarcasm detection in microblogs using naïve bayes and fuzzy clustering. *Technology in Society*, 48:19–27, 2017.
18. Isidoros Perikos and Ioannis Hatzilygeroudis. Recognizing emotions in text using ensemble of classifiers. *Engineering Applications of Artificial Intelligence*, 51:191–201, 2016.
19. Saurabh Porwal, Gaurav Ostwal, Anagha Phadtare, Mohini Pandey, and Manisha V Marathe. Sarcasm detection using recurrent neural network. In *2018 Second International Conference on Intelligent Computing and Control Systems (ICICCS)*, pages 746–748. IEEE, 2018.
20. Rui Ren, Desheng Dash Wu, and Tianxiang Liu. Forecasting stock market movement direction using sentiment analysis and support vector machine. *IEEE Systems Journal*, 13(1):760–770, 2018.
21. Shubham Rendalkar and Chaitali Chandankhede. Sarcasm detection of online comments using emotion detection. In *2018 International Conference on Inventive Research in Computing Applications (ICIRCA)*, pages 1244–1249. IEEE, 2018.
22. Liang-Chih Yu, Jin Wang, K Robert Lai, and Xuejie Zhang. Refining word embeddings using intensity scores for sentiment analysis. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 26(3):671–681, 2017.