IJCRT.ORG ISSN : 2320-2882

INTERNATIONAL JOURNAL OF CREATIVE RESEARCH THOUGHTS (IJCRT)
An International Open Access, Peer-reviewed, Refereed Journal

Sarcasm Analysis Using Machine Learning

Sparsh Arora, Siddhant Bagat, Shrey Mathur, Shruti Agarawal

Student, Student, Student, Professor

Department of Information Technology,

Vidyalankar Institute of Technology, Mumbai, India

***Abstract*:** The growth of social media has been exponential within the recent years. Complete information is being put out on the domains directly through social media. Sarcastic tweets can mislead processing activities and end in wrong classification. This paper compares various classification algorithms like Random Forest, SVC, Logistic Regression, Linear SVC and Gaussian Naïve Bayes to detect sarcasm within the input given by the user, the simplest classifier is chosen and paired with various pre-processing and filtering techniques using emojis to supply the simplest possible output. The emojis being the central idea introduced through this paper, the obtained results may be used as an input for other research and applications.

1. INTRODUCTION

Sarcasm is also a nuanced form of communication where the individual states opposite of what's implied. one in every of the most challenges of sarcasm detection are its ambiguous nature. there's no prescribed definition of sarcasm. Another major challenge is that the growing size of the languages. Everyday many new slang words are being created and used on these sites. Hence, the prevailing corpus of positive and negative sentiments may not convince to be accurate in detecting sarcasm. Also, the recent developments in online social networks allow its users to use various emoticons with the text. These emoticons may change the polarity of the text and make it sarcastic.
Sarcasm is also a nuanced form of communication where the individual states opposite of what's implied. one in every of the most challenges of sarcasm detection are its ambiguous nature. there's no prescribed definition of sarcasm. Another major challenge is that the growing size of the languages. Everyday many new slang words are being created and used on these sites. Hence, the prevailing corpus of positive and negative sentiments may not convince to be accurate in detecting sarcasm. Also, the recent developments in online social networks allow its users to use various emoticons with the text. These emoticons may change the polarity of the text and make it sarcastic.

2. LITERATURE SURVEY

In [1], authors show the interest in caustic remark delectation inside the loudspeaker system. For capturing real time tweets, they use the Hadoop base framework, and processes that tweets they used the numerous six formulas like parsing based mostly lexicon generation algorithm (PBLGA), tweets contradicting with universal facts (TCUF), interjection word begin (IWS), positive sentiment with word combine (PSWAP), Tweets disaffirming with time- dependent facts (TCTDF), Likes dislikes contradiction (LDC), these algorithms area unit used identifies critical sentiment effectively. this system is generally appropriate for real time streaming tweets.

[2], authors use the machine system it's use for harnesses context incompatibility as a basis for caustic remark detection. caustic remark classifier uses four forms of features: lexical, pragmatic, express incompatibility, and implicit incompatibility options. They outline system on 2 text forms: tweets and discussion forum posts. For improvement of performance of tweet uses the rule base formula, and to boost the performance for discussion forum posts, uses the novel approach to use elicitor posts for caustic remark detection. this system conjointly introduces error analysis, the system future work
(a) Role of numbers for caustic remark, and,
(b) things with subjective sentiment.

In [3], authors used the machine learning method to caustic remark detection on Twitter in English and Czech. 1st work is caustic remark detection on Czech language. They used the two classifier most Entropy (Maxent) and Support Vector Machine (SVM) with completely different combos of options on each the Czech and English datasets. conjointly use the varied preprocessing technique like Tokenizing, POS tagging, no stemming and Removing stop words, its use for locating the matter of Czech language.

In [4], authors have inspected options of caustic remark on Twitter. {they're | they area unit} involved not simply with distinguishing whether or not tweets are critical or not, however conjointly take into account the polarity of the tweets. They even have compiled style of rules that improve the accuracy of sentiment analysis once caustic remark is assumed to be gift. investigator have developed a hash tag tokenizes for GATE technique thus sentiment and caustic remark found inside hash tag are detected additional simply. Hash tag tokenization technique is very helpful for detection of caustic remark and checks the polarity of the tweet i.e., positive or negative.

In [5], authors area unit used 2 ways like lexical and pragmatic factors that area unit used for differentiate between caustic remark from positive and negative sentiments expressed in Twitter messages. They conjointly created corpus of critical Twitter messages inside that determination of the caustic remark of each message has been created by its author. Corpus is employed to visualize critical utterances in Twitter to utterances that show positive or negative attitudes while not caustic remark.

In [6], author has developed a true statement recognizer to visualize caustic remark on Twitter consists of a positive sentiment contrasted with a negative scenario of caustic remark in tweets. They apply novel bootstrapping formula that instinctively learns lists of positive sentiment phrases and negative scenario phrases from critical tweets. They prove that intercede contradicting contexts exploitation the phrases learned through bootstrapping. Rule-based approaches attempt to establish caustic remark through specific proof. These evidences area unit captured in terms of rules that rely upon indicators of caustic remark. target distinguishing whether or not a given figure of speech (of the form '* as a *') is supposed to be critical. They use Google search thus on see however doubtless a figure of speech is. They gift a 9-step approach wherever at every step rule; a figure of speech is valid exploitation the number of search results. Strength of this approach is that they gift an error analysis admire multiple rules [9]. The hash tag sentiment might be a key indicator of caustic remark. Hash tags area unit typically used by tweet authors to spotlight caustic remark, and hence, if the sentiment expressed by a hash tag does not trust remainder of the tweet, the tweet is foretold as critical. They use a hash tag tokenizer to separate hashtags made up of concatenated words.

In [7], this work is that the most up-to-date advancement of the caustic remark detection and irony identification on twitter sentiment analysis. It uses existent algorithms like SVM, Adaboost and call tree classifiers and performs Tokenization, Stemming and Lemmatization part-Of Speech (POS)-tagging, Feature choice then model analysis. However, the paper will embody emoji and slang dictionaries which might improve the accuracy.

In [8], authors have introduced enhancements by as well as the history of tweets and author profiles which is able to aid within the classification method. The paper presents accuracies starting from seventieth and upwards for various eventualities. supported additional demanding analysis and referral of varied publications, the conclusion obtained was that the analysis as yet, have elapsed the very fact that emojis and slangs play an enormous role in tweets of the trendy day. The new trend of tweets that extensively use slangs, short-forms and emojis area unit throwing gift algorithms off-balance as they are doing not complete such changes within the trend of tweets. The paper derives inspiration from numerous mentioned publications and multiple others in building of this classifier model. we have a tendency to propose numerous amends like inclusion of the slang and emoji dictionaries for classification which can lead to additional accuracy.

In [9], the authors introduction to the sector of caustic remark detection was mentioned within the works of Buscaldi et al. that explains well what a critical tweet is. The article on options that create classification doable and provides in-depth clarification relating to however the varied options contribute to classification and results.

3. PROPOSED SYSTEM

A) SYSTEM ARCHITECTURE

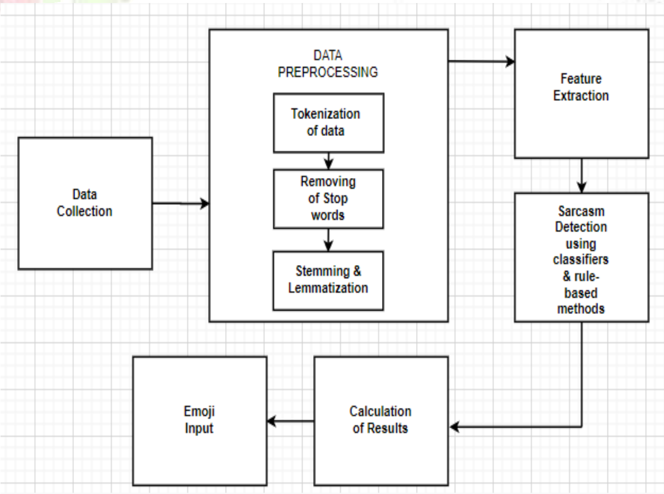

Figure 1 – System Architecture

B) PROPOSED WORK

The proposed system recognizes the sarcastic emotions of the individuals with the use of MNNB (Multi-Nominal Naïve Bayes) algorithm and also to identify the type of sarcastic emotions using SVM, random forest & decision tree method. . In this work, the first step is to extract data from the datasets; feature extraction, sarcasm detection and identification of categories of sarcasm have been performed. Anaconda version 4.2.0 software is used for the implementation. Jupyter Notebook is used as the important tool. Python is used for the implementation of the proposed system. Twitter datasets is taken from the corpus or by using the tweepy software depending upon the requirements. The dataset is incorporated, and feature extraction is performed. Feature extraction is carried out using Count Vectorizer which is used to convert the text format into vector format. Multinomial Naïve Bayes is used to find sarcastic and non-sarcastic sentences**.**

C) IMPLEMENTATION

a. FEATURE EXTRACTION

Count vectorizer is a method which used in feature extraction. It is accustomed to convert the text into binary format. It offers a modest technique to both tokenize an assortment of text documents and build a jargon of known words, but also to encode new documents using that vocabulary. Next is Tf-idf Vectorizer which will tokenize documents, learn the jargon and converse document frequency weightings, and allow them to encode new documents. Consecutively, if the users already have a learned Count vectorizer, he/she can use it with a Tf-idf transformer to just calculate the inverse document frequencies and start encoding documents.

b. SARCASM DETECTION

The sarcasm detection in the proposed model is done using classifiers such as Decision Tree, Random Forest, Gradient Boosting, Adaptive Boosting, Logistic Regression and Gaussian Naive Bayes. Decision Tree Classifier is a simple and widely used classification technique. Decision Tree Classifier poses a series of carefully crafted questions about the features that are supplied to the algorithm. Based on the answers received there are more questions are posed and ultimately class labels are assigned based on the cumulative answers. Random Forest Classifier works by building multiple decision trees and obtaining class labels. Gradient boosting generates learners during the learning process. It builds first learner to predict the values/labels of samples and calculates the loss (the difference between the outcome of the first learner and the real value). It then builds a second learner to predict the loss after the first step. The step continues to learn the third, fourth and so on, until a certain threshold. Adaptive boosting requires users to specify a set of weak learners (alternatively, it will randomly generate a set of weak learners before the real learning process). It will learn the weights of how to add these learners to be a strong learner. The weight of each learner is learnt by whether it predicts a sample correctly or not. If a learner predicts a sample incorrectly, the weight of the learner is reduced a bit. The process is repeated until convergence. Naive bayes and logistic regression both are log-linear models; that is, in both cases the probability of a document belonging to a class is proportional to exp(w·x), where w is a classifier parameter and x is a feature vector for the document. The main difference is that, in naive bayes, the model is specified so that both the data and the labels depend on w, while in logistic regression only the labels depend on w. These classifiers are given the newly formed dataset as the input. The various features and their weights are considered by the algorithm They are trained for various splits ranging from 60:40, 70:30 and 80:20 of training to testing data. The most accurate algorithm from the best performing split is selected for testing and the same is used for the real-time functioning of the model. For ours, linear SVC worked best showing over 0.57 accuracy.

c. MATHEMATICAL DERIVATION

Notation: X is your sample dataset, it contains N samples, P features. X_{ij} is the ith sample, jth feature. X_i is the vector describe all the P features of sample i. X+ is the sample from class"+", X- is the sample from class"-". Y is the label vector; Y_i has 2 values +1 and -1.

$\frac{1}{2}||\omega||^2$ function is our target function, but it comes with a constrained function: $Y_i\,(\vec{\omega}\cdot\vec{X}_i+b) = 1$

Now, let's pack them together.

$$\mathbf{L = \frac{1}{2}\,||\omega||^2 - \sum\alpha_i[Y_i(\vec{\omega}\cdot\vec{X}_i+b) - 1]}$$

$$\frac{\partial L}{\partial w} = 0, \text{ we will have } \vec{\omega} = \sum\alpha_i Y_i X_i$$

$$\frac{\partial L}{\partial b} = 0, \text{ we will have } \sum\alpha_i Y_i = 0$$

Plug these 2 equations back to L,

$$L = \frac{1}{2}\,(\sum\alpha_i Y_i\vec{X}_i)(\sum\alpha_j Y_j\vec{X}_j) - \sum\alpha_i[Y_i(\sum\alpha_j Y_j\vec{X}_i\cdot\vec{X}_j + b)\ 1]$$

$$= -\frac{1}{2}\sum_{ij}\alpha_i\alpha_j Y_i Y_j\vec{X}_i\cdot\vec{X}_j + \sum\alpha_i$$

From this we can see, the classifier is only depending on the dot product of the sample.

d. EVALUATION METRICS

In this work, confusion matrix is used to calculate true positive and false positive rate which is used to evaluate the precision and recall. Confusion matrix consists of true positive, true negative, false negative, false positive depending upon that the graph is plotted. It will be in the diagonal arrangement true positive rate is 0.92 and false positive rate 0.9. A confusion matrix may be a table that's often won't to describe the performance of a classification model (or "classifier") on a group of test data that the true values are known. It allows the envision of the performance of an algorithm. It permits simple identification of confusion between classes example: - one class is commonly mislabeled as the other.

4. RESULTS

Classifier	Accuracy
SVC	0.906
Linear SVC	0.950
Random Forest Classifier	0.554
Decision Tree Classifier	0.455

Figure 3 - Results

Accuracy = $\frac{TP + TN}{TP+TN+FP+FN}$

Where TP = True Positive, TN = True Negative
FP = False Positive, FN = False Negative.

In the table given above, one can see that we have used different classifiers to verify their accuracy which can be better than the existing systems. Simple Vector Machine (SVM) classifier gives the accuracy of 0.5(approx.). When using linear SVC, it gives us the accuracy of 0.577. when we use random forest classifier it gives us the training accuracy of 0.999 and test accuracy of 0.554. when we use decision tree classifier in given accuracy of 0.458. as you can see the best accuracy was obtained in linear SVC so we used earlier SVC in our program. Out of all the given systems already present our program gives more accuracy. Also, it helps to understand the exact sarcasm used in the sentence with the help of emojis.
Here, we have created a webpage using HTML5 and CSS for accepting input and output display. Our machine learning code is stored in the "pickle" format and flask –framework is further used to integrate it into a web application. Our webpage displays an input bracket- to accept input. Once you enter the input, press the "predict" button underneath it to generate output. The output consists of an emoji sticker and a text message relative to the input previously provided. Example: –
INPUT: Every time I imagine that someone I love or I could contact a serious illness, even death.
OUTPUT: Negative Sarcasm

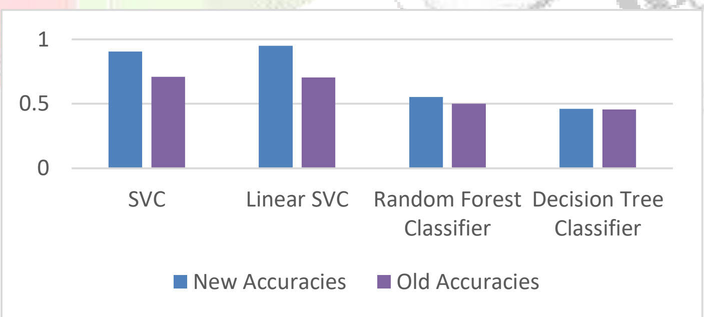

Figure 4 – Comparison

5. CONCLUSION

In this paper, a way of improving the existent sarcasm detection algorithms by including better pre-processing and text mining techniques such as emoji detection are presented. For classifying tweets as sarcastic and non-sarcastic there are various techniques used. However, the paper takes up a classification algorithm and suggests various improvements, which directly contribute to the development of accuracy. The project derived analytical views from the input i.e., dataset and also filtered out or reverse analyzed sarcastic inputs to achieve a comprehensive accuracy in the classification of the data that is presented and also give an emoji representing the tone of the user input, as output.
The model would be further implemented for bigger datasets like twitter in the near future. On the basis of the survey, this project will formulate mainly two possible future works - Exploring new feature and new data sets and coverage of various types of sarcasm. The model would be revamped and brought to greater accuracy and unmatched precision.

6. REFERENCES

[1] S.K. Bharti B. Vachha, R.K. Pradhan, K.S. Babu, S.K. Jena "Sarcastic sentiment detection in tweets Streamed in real time: a big dataapproach",Elsevier12July2016.

[2] Aditya Joshi, Vinita Sharma, Pushpak Bhattacharyya "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), pages 757–762, Beijing, China, July 26-31, 2015. C 2015 Association for Computational Linguistic.

[3] Toma Ptacek, Ivan Habernal and Jun Hong "Sarcasm Detection on Czech and English Twitter", Proceedings of COLING 2014, the 25th International Conference on Computational Linguistics: Technical Papers, pages 213– 223, Dublin, Ireland, August 23-292014.

[4] D.Maynard , M.A.Greenwood.2014."Who cares about sarcastic tweets? Investigating the Impact of sarcasm on sentiment analysis", In Proceedings of the LREC 2014 May 26-31

[5] W.Tan, M.B.Blake, I.saleh, S.Dustdar, "Social- network sourced big data analytics", InternetComput.17 (5) (2013)62–69.

[6] E. Riloff, A. Qadir, P. Surve, L. De Silva, N. Gilbert, and R. Huang, "Sarcasm as contrast between a positive Sentiment and negative situation", in Proc. Con Empirical Methods Natural Lang. Process, Oct.2013, pp.704_714.

[7] Erik Forslid and Niklas Wikén., "Automatic irony- and sarcasm detection in Social media", ISSN: 1401-5757, UPTEC F15 045, 2015.

[8] David Bamman and Noah A. Smith., "Contextualized Sarcasm Detection on Twitter", School of Computer Science, Carnegie Mellon University (2016)

[9] Barbieri F., and Saggion, H. 2014. "Automatic Detection of Irony and Humour in Twitter", In Proceedings of the Student Research Workshop at the 14th Conference of the European Chapter of the Association for Computational Linguistics, 56–64. Gothenburg, Sweden: Association for Computational Linguistics.

[10] Tony Veale and Yanfen Hao. 2010. "Detecting Ironic Intent in Creative Comparisons", In ECAI, Vol. 215.765– 770.

[11] A. Rajadesingan, R. Zafarani, and H. Liu, ``Sarcasm detection on Twitter A behavioral modeling Approach", in Proc. 18th ACM Int. Conf. Web Search Data Mining, Feb. 2015,pp.79_106.

[12] M.Bouazizi, T.Ohtsuki ,"Pattern-Based Approach for Sarcasm Detection on Twitter" VOLUME 4, 10.1109/ACCESS.2016.2594194.