ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

Sarcasm Detection in Plain Text Using Machine Learning

[1] D.V.Nagarjana Devi, [2]Dr.T.V.Rajanikanth, [3] Dr.V.V.S.S.S. Balaram
[1] Assistant Professor, [2] [3] Professor, SNIST
[1] IIIT-RGUKT Nuzvid, [2] [3] Hyderabad,TG.

Abstract: Sarcasm is a sophisticated form of irony widely used in social networks and micro blogging websites. It is usually used to convey implicit information within the message a person transmits. Sarcasm might be used for different purposes, such as criticism or mockery. However, it is hard even for humans to recognize. Therefore, recognizing sarcastic statements can be very useful to improve automatic sentiment analysis of data collected from micro blogging websites or social networks. Sentiment analysis refers to the identification and aggregation of attitudes and opinions expressed by internet users towards a specific topic. For the detection of Sarcasm in plain text we are going to use Machine Learning Classification Methods. By detecting Sarcasm from social media's like twitter we can identify irrelevant and sarcastic opinions of people. These opinions can be used as reviews for the effective business decisions.

Keywords: Sarcasm, Stemming, Unigrams, Classification.

I. INTRODUCTION

Sarcasm is defined as a cutting, often ironic remark intended to express contempt or ridicule. Sarcasm detection is the task of correctly labelling the text as 'sarcastic' or 'non-sarcastic'. It is a challenging task owing to the lack of intonation and facial expressions in text. Nonetheless humans can still spot a sarcastic sentiment in the text and reason about what makes it so. Recognizing sarcasm in text is an important task for Natural Language processing to avoid misinterpretation of sarcastic statements as literal statements. Accuracy and robustness of NLP models are often affected by untruthful sentiments that are often of sarcastic nature. Thus, it is important to filter out noisy data from the training data inputs for various NLP related tasks. For example, a sentence like "So thrilled to be on call for work the entire weekend!" could be naively classified as a sentence with a high positive sentiment. However, it's actually the negative sentiment that is cleverly implied through sarcasm. The use of sarcasm is prevalent across all social media, micro-blogging and e-commerce platforms. Sarcasm detection is imperative for accurate sentiment analysis and opinion mining. It could contribute to enhanced automated feedback systems in the context of customer based sites. Twitter is a micro-blogging platform extensively used by people to express thoughts, reviews, and discussions on current events and convey information in the form of short texts. The relevant context of the tweets are often specified with the use of # (hash-tag). Twitter data provides a diverse corpus for sentences which implicitly contain sarcasm. In academic literary works on sarcasm detection from tweets, sarcastic tweets are mostly sampled by querying the Streaming API using keywords #sarcasm and other sentiment tweets, filtering out non-English tweets and re-tweets. For this task, we used the available resources at [1]. However, it was found that this collection of tweets was indeed slow and did not yield a rich set of sarcastic (perceivably sarcastic) tweets. Thus, we resorted to use an existing dataset. We are using Machine learning. It is a type of artificial intelligence (AI) that provides computers with the ability to learn without being explicitly programmed. Machine learning focuses on the development of computer programs that can change when exposed to new data. In any Machine Learning task features are of central importance. The quality of the classification depends on the features selected. Carefully designed and chosen features play a big role in improving the results both qualitatively and quantitatively. Sarcasm detection is a non-trivial task. Usually sarcasm is cleverly embedded in a sentence which has a positive sentiment. The context also plays a role in determining whether sarcasm is present as a hidden sentiment or not. Hence, it is a linguistically complex task in the domain of Natural Language Processing. Rule-based model for detecting sarcasm would have very limited performance and its application would be specific to the data.

IFERP connecting engineers... developing research

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

II. PROCEDURE FOR SARCASM DETECTION

Sarcasm in plain text is being detected by following some step by step methods as prescribed in following fig1.

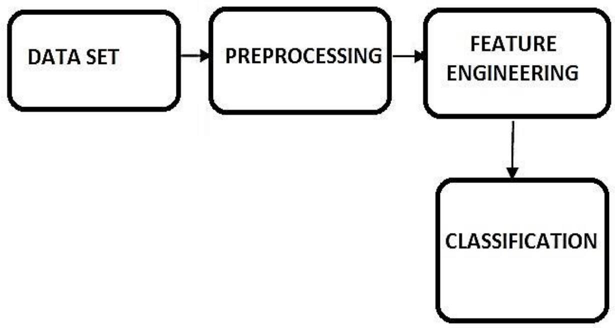

Fig1: Algorithmic architecture of Sarcasm detection.

ALGORITHMIC PROCEDURE

STEP1: Collected Datasets from twitter
STEP2: Pre-processing of data
STEP3: Filtration of html tags, stop words, non-letters
STEP4: Stemming each words
STEP5: Feature engineering the data
STEP6: counting user tags, hash tags, capitalized words
STEP7: unigrams
STEP8: Creating Boolean matrix
STEP9: Classification
STEP10: Applying SVM classification
STEP11: Applying Logistic Regressing classification
STEP12: Applying Naïve Bayes classification
STEP13: Applying Decision Tree classification
STEP14: Observing results with ROC (Receiver Operating Characteristic) curves.

Eventually data set was collected from twitter database 12000 sarcastic and 12000 non-sarcastic tweets were taken to test the sarcasm.

A. PREPROCESSING

In order to prepare our corpora for use, it first had to be sanitized. The pre-processing aims to minimize the vocabulary of terms used in the tweets. This involved a certain amount of pre-processing steps which involved.

i) Tokenizing, stemming, and filtering out non-English tweets. This process is known as cleaning un-useful and meaningless words from each tweet.
ii) Filtering out duplicate html tags and hyperlinks removed because html tags and hyperlinks expresses no meaning.
iii)) Hash tags #sarcasm and #sarcastic were filtered out in order to not influence our models due to their presence. All other hashtags were kept in place.

Each and every tweet of corpora preprocessed by following previous steps. All tweets were tokenized and stemmed. Hyperlinks, non-English tweets were filtered out.

B. FEATURE ENGINEERING

The main features we chose here were:
i) User tags (@UserName). User tags are generally used to mention another user or response to the tweet.
ii) Hashtags (#hashtag). Except #sarcasm and #sarcastic all the other hashtags are counted and taken in consideration to be a feature.
iii) The use of all-caps. This feature was mentioned by Bammas et al (2015)[1], although it was used in the context of both initial caps and all-caps. We chose all-caps as this is one of the known cues expressing emotion in written text[2], and we felt that words with initial capitalization would not be as predictive.
iv) Unigrams: formed Boolean matrix by using unigrams from tweets with max feature 5000.

C. CLASSIFICATION

While looking to classify our test data, we chose four methodologies in which to test our features:
i) Naïve Bayes (NB), where we assume features are independent,
ii) Support vector machine (SVM), where 'support vectors' determine the ideal decision boundary by separating sarcastic tweets from genuine tweets,
iii) Logistic Regression (LR), where our independent variables are the features we've mentioned, and the dependent variable is a binary response whether or not a tweet is sarcastic.
iv) Decision Trees (DT), where feature importance is clear and relations can be viewed easily.

III. EVALUATION

A. Experimental data

The data used in the experiment was gathered manually from twitter. The size of training data is 24000 which consists of 12000 sarcastic and 12000 genuine tweets. The size of the training and testing data is 16000 tweets, which consists 60% as trained data and 40% as testing data.

B. Evaluation results and analysis

Tweets were evaluated by using four classifiers. The first classification is on Support Vector Classification, second one is Logistic Regression, third one is Naïve Bayes Classification and fourth one is Decision Tree Classification. In order to evaluate our results, we chose a fairly standard heuristic, that is the area under the curve (AOC) for a receiver operating characteristic (ROC) curve for all classifications.

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

1) Support Vector Classification
After preprocessing and Feature engineering we got the well-organized data within that we have trained SVC with 60% data and used 40% for testing purpose and we got 89% accuracy.

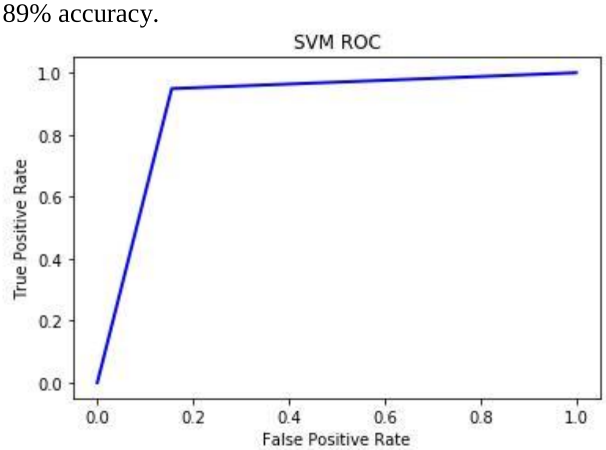

Fig 2: SVM ROC

2) Logistic Regression
When evaluating our methodology, it is clear that Logistic Regression gave us the best results. We hoped for a target of 90% and the Logistic Regression has given us more than that a. We got 94% accuracy with this. Given that our tweet classification was inherently binary sarcastic or non-sarcastic it follows that logistic regression is an appropriate classifier for undertaking this task.

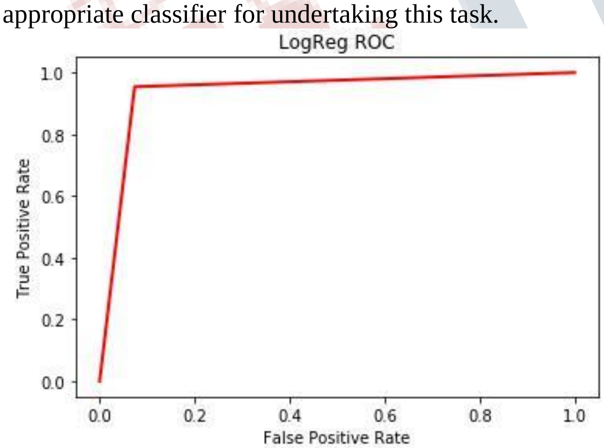

Fig 3: Logistic Regression ROC

3) Naïve Bayes classification
Naïve Bayes is one among the well-known classification algorithms. We used this classification method to test accuracy level and we got 78% accuracy.

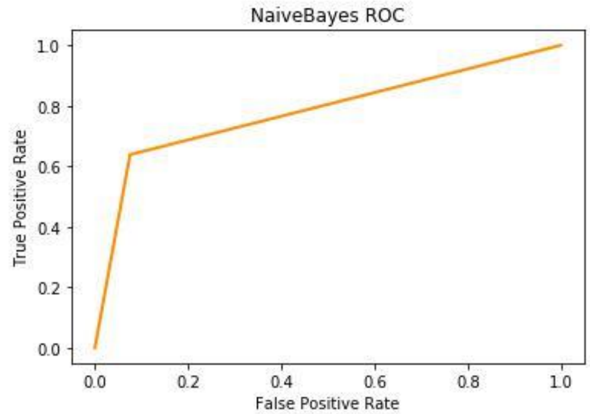

Fig 4: Naïve Bayes ROC

4) Decision Tree classification
As we have seen earlier logistic regression given the best output, and by testing with Decision Tree classification we got as much as Logistic Regression's accuracy that was 94% accuracy.

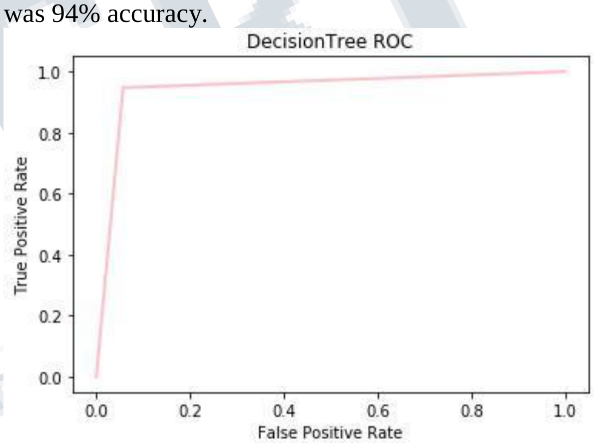

Fig 5: Decision Tree ROC

We have analyzed sarcastic and non-sarcastic tweets and represented by using ROC curve for each classifier for better understanding.

TESTING METHOD:
In testing tweets we were adopted a algorithmic process as, when we are testing tweets in each classifier if any one of the four classifier declares it as sarcastic tweet then that tweet further considered as sarcastic without considering other classifiers result. With this process the accuracy level was increased.

IV. RELATED WORK

Social media is beginning to become an all-encompassing means to determining people's opinions as sentiments

ISSN (Online) 2394-2320

International Journal of Engineering Research in Computer Science and Engineering (IJERCSE)
Vol 5, Issue 4, April 2018

towards products, events, and people. There have been a number of works which have attempted to solve the issue of sarcasm detection, as it relates to media such as Twitter. Muresan et al (2016)[3] propose a method wherein training data, which has been tagged as sarcastic by its author, is used to compare sarcastic utterances in Twitter which express positive or negative attitudes either with or without sarcasm. A number of features are investigated to determine their effectiveness and the results are compared to human judges, which represent the gold standard. In the broad scheme of things, this work is the similar to ours, our primary motivation and inspiration. Further inspiration was taken from Ghosh et al (2015)[4] where sarcasm detection is framed as a word disambiguation problem. By this, they classified word senses as literal or sarcastic and call this task Literal/Sarcastic Disambiguation (LSSD). In their work, they show that a SVM classifier with a modified kernel using word embeddings shows a 7-10% improvement in classification. Davidov et al (2010)[5] present a semi-supervised model which analyzed sarcastic sentences in Twitter, as well as in Amazon reviews. They go much more in-depth as to whyusing the #sarcasm hashtag is effective in providing a secondary gold standard for determining sarcasm. This methodology presents results that are generally on par with primary gold standard, being human-annotated sentences. Because of this, detecting sarcasm from the Twitter dataset were much more effective than just using a corpus of Amazon reviews alone. Peng et al (2015)[6] worked to leverage the work of Mathieu Cliche of The Sarcasm Detector[1] and improve his results. They utilized three models for classification: NB, one-class SVM trained only on non-sarcastic data, and a Gaussian Kernel. Their results showed that the NB and One-Class SVM models did not perform nearly as well as the Gaussian kernel model.

V. CONCLUSION AND FURTHER WORK

We can say that our classifier is working as intended, we have successfully examined sarcastic and non-sarcastic tweets using the set of features we have outlined above. We got the best results in Logistic Regression and Decision Tree classifications Till ow we just analyzed few best working classification methods for the detection of sarcasm in the plain text with highest accuracy. In future we will discover a better hybrid classification method to detect sarcasm in plain text using machine learning by combining all these classifications and adding additional feature sentiment analysis for the best accuracy levels.

REFERENCES

[1] D. Bammam and N.A. Smith, “ Contextualized Sarcasm Detection on Twitter,” presented at the 2015 Int. AAAI Conf. on Web and Social Media, Oxford, UK, May 25-29, 2015.

[2] K. Denham and A. Lobeck, “ Representing Language: The Written Word,” in Linguistics for Everyone: An Introduction, 2nd ed. Belmont, CA: Wadsworth Publishing, 2012, ch. 13, p.462.

[3] S. Muresan. R.Gonzalez-lbanez, D. Ghosh, and N. Wacholderr, “Identification of Nonliteral Language in Social Media: A Case Study on Sarcasm,” Journal of the Assoc. for info. Science and Tech.”

[4] Debanjan Ghosh, Weiwei Guo, and Smaranda Muresan, “Sarcastic or Not: Word-Embeddings to Predict the Literal or Sarcastic Meaning of Words”, In Proceedings of EMNLP, 2015, Lisbon, Portugal

[5] D. davidov, O. Tsur, and A.Rappoport, “Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon,” presented at the 2010 Conf. of Computational Natural Language Learning, Uppsala, Sweden, July 15-16, 2010.

[6] C.C. Peng, M.Lakis, and J.W. Pan, “Detecting Sarcasm in Text: An Obvious Solution to a Trivial Problem,” unpublished.

