Sarcasm Detection based on Twitter Data

Kasat, Darshan, M.S.
Nima Zahadat, Ph.D.
George Washington University,
darshankasat22@gwu.edu

Abstract –

Sarcasm is a nuanced form of communication where the individual states the exact opposite of what is implied. Identifying sarcasm is an important task in order to build robust and accurate models for NLP related tasks. I believe that the underlying features in the text can be useful in detecting sarcasm. For example, the use of emoticons can change the polarity of the sentence and make it sarcastic. This paper presents a machine learning based approach to detect sarcasm in tweets. I generate pragmatic and sentiment based features from the tweets and build Logistic Regression and Support Vector Machine model to classify the tweet as sarcastic or non-sarcastic.

Index Terms – logistic regression, sarcasm, SVM, twitter

I. Introduction

Sarcasm is defined as a cutting, often ironic remark intended to express contempt or ridicule. Sarcasm detection is the task of correctly labeling 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 mis-interpretation 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 "Sp thrilled to be on call for work the entire weekend!" could be naively classified as a sentence with high positive sentiment. However, it is 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. Twitter is a micro-blogging platform extensively used by people to express thoughts, reviews, discussions on current events and convey information in the form of short texts. Twitter data provides a diverse corpus for sentences which implicitly contains sarcasm.

II. Related work

There has been a significant research in studying the nature of a sarcasm, the properties that attribute to a text being sarcastic. Kreuz et al. [2] study lexical factors to determine their relative importance on the perception of sarcasm. They found out that the lexical features such as punctuation and interjection are effective in determining the sarcastic nature of text. Joshi et al. [1] present a computational system that uses context incongruity as a basis for detecting sarcasm. They achieved a precision score of 0.77 that beat the two state-of-the-art models for sarcasm detection at that time. These papers gave a fair idea of the features that attribute towards building a good classifier for sarcasm detection.

More recently, Lukin et al. [3] extended the idea of using pragmatic and incongruity based features by using n-gram based features and lexicon-syntactic patterns. Ellen et al. [4] considered 'Sarcasm as a contrast of positive sentiment and negative situation'. They generated corpuses for negative and positive phrases using a novel bootstrapping algorithm. They trained the machine learning classifiers using the tweets containing '#sarcasm'. Alec et al. [5] enormously focused on preprocessing the data inorder to make it coherent with the desired machine learning method. After extracting the features, they trained a SVM classifier for sarcasm detection.

Studies have shown that the lexical, pragmatic, and incongruity based features are effective in classifying the text as sarcastic or non-sarcastic. The work of Joshi et al. [] relates a lot to my idea of combining all those features together for training machine learning based classifier. This paper brings in the exact motivation of what I intend to work.

III. Dataset

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, an attempt was made to collect the data using the Streaming API. However, it was found that this collection of tweets was indeed slow and did not yield a rich set of sarcastic tweets. Thus, I resorted to use an existing dataset of [6].

IV. Feature Engineering

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. In order to determine whether sarcasm is present as a hidden sentiment, context must also be taken into consideration. Hence, sarcasm detection is a linguistically complex task in the domain of Natural Language Processing.

The features in my model can be divided into three categories namely :-

- Lexical
- Pragmatic
- Linguistic Incongruity

These features are further described in detail below :-

1. Lexical Features

N-grams are a commonly used feature set for NLP related tasks in machine learning. Certain words or phrases like “Yeah right!” may be prevalent in tweets that are sarcastic. We use *unigrams* inorder to extract the lexical information contained in the tweets.

Using training corpus, a dictionary is created. Each unique word is mapped onto a particular ID. These ID numbers are used a feature numbers. The values corresponding to each such feature number is the frequency of occurrence of that particular word in the tweet for which we are generating the feature values. Hence, the feature vector would be naturally containing a lot of 0’s corresponding to the words that are present in the dictionary but not in the tweet. Such ID’s with values can be discarded as we are looking for the presence of words prevalent in sarcastic tweets.

2. Pragmatic Features

These features are associated with the grammatical hints. These include:

(a) Number of Capital Letters

In order to lay extra emphasis on the emotion to be conveyed, people employ capitalization. In a similar way, sarcasm might be highlighted by the author to create an impact. Hence this is used as a feature.

(b) Number of Emoticons

Emoticons are commonly used across social media platforms to express sentiments. I used the ‘codecs’ module in python which allowed me to open files containing emoticons in UTF-8 format and read. And then using the regular expressions, the emoticons present can be captured.

(c) Number of slang laughter expressions

The slang expressions such as ‘lol’, ‘rofl’, ‘lmao’ and ‘haha’ are popular and fairly well used on the social media platforms. The frequency of occurrence of these expressions is a feature. Higher occurrence of these expressions is potentially indicative of sarcasm.

(d) Number of Punctuation marks

Exclamation mark is often used to lay extra emphasis on the underlying emotion like surprise, shock or dismay. Even in sarcastic tweets, such use of exclamation marks is prevalent especially that of ‘!’, ‘?’ and ‘...’. The count of such punctuation is hence used as a feature.

(e) Number of User Mentions

Studies have shown that the number of user mentions is prevalent in detection of sarcasm. Hence, the count of user mentions is also used as a feature.

(f) Number of hashtags

The relevant context of the tweets are often specified with the use of hashtags. Thus the count of hashtags can be used as a feature.

3. Explicit Incongruity

The linguistic theory of *Context Incongruity* suggests that the common form of sarcasm expression consists of a positive sentiment which is contrasted with the negative situation. Well defined and extensively used statistical models can benefit from the use of features generated on the basis of well-established linguistic theories. The particular features used in this category are:

(a) Number of sentiment incongruities

A single numeric feature value which gives the count of the number of times a positive word is followed by a negative word and vice versa.

(b) Number of positive, negative and neutral words

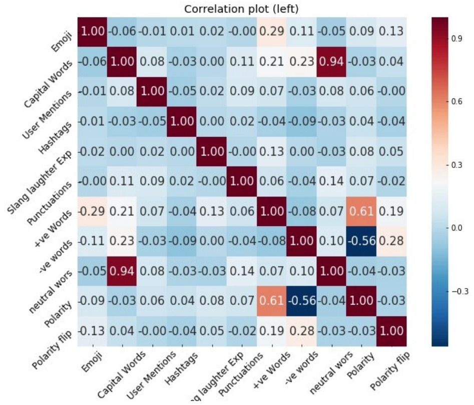
Using VADER (Valence Aware Dictionary and Sentiment Reasoner) [6] module in python, the polarity of each word is generated. The value generated are in the range of [-1,1]. If the value is positive, it is taken as a word with positive polarity. Similarly, if it is negative, it is taken as a word with negative polarity. Using the count of such words can be used as a feature.

(c) Lexical Polarity

This is the overall polarity of the entire sentence. Owing to the theory of lexical incongruity, it can be observed that a tweet which has an overall strong positive sentiment is more likely to be sarcastic compared to the tweet with overall negative polarity.

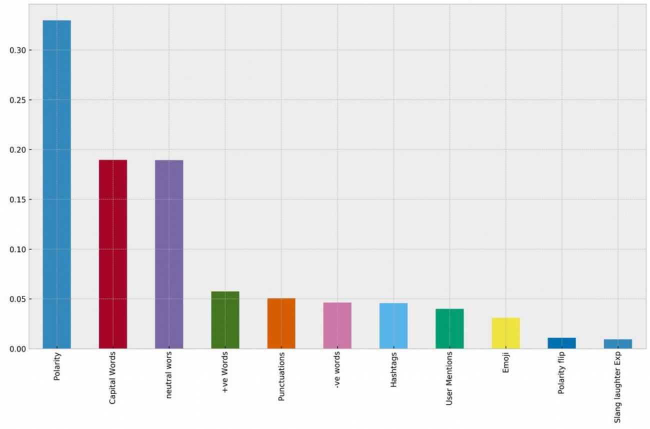
Using the VADER module, the overall polarity of the tweet is calculated.

V. FEATURE ANALYSIS



	Emoji	Capital Words	User Mentions	Hashtags	g laughter Exp	Punctuations	+ve Words	-ve words	neutral wors	Polarity	Polarity flip
Emoji	1.00	-0.06	-0.01	0.01	0.02	-0.00	0.29	0.11	-0.05	0.09	0.13
Capital Words	-0.06	1.00	0.08	-0.03	0.00	0.11	0.21	0.23	0.94	-0.03	0.04
User Mentions	-0.01	0.08	1.00	-0.05	0.02	0.09	0.07	-0.03	0.08	0.06	-0.00
Hashtags	0.01	-0.03	-0.05	1.00	0.00	0.02	-0.04	-0.09	-0.03	0.04	-0.04
Slang laughter Exp	0.02	0.00	0.02	0.00	1.00	-0.00	0.13	0.00	-0.03	0.08	0.05
Punctuations	-0.00	0.11	0.09	0.02	-0.00	1.00	0.06	-0.04	0.14	0.07	-0.02
+ve Words	0.29	0.21	0.07	-0.04	0.13	0.06	1.00	-0.08	0.07	0.61	0.19
-ve words	0.11	0.23	-0.03	-0.09	0.00	-0.04	-0.08	1.00	0.10	-0.56	0.28
neutral wors	-0.05	0.94	0.08	-0.03	-0.03	0.14	0.07	0.10	1.00	-0.04	-0.03
Polarity	0.09	-0.03	0.06	0.04	0.08	0.07	0.61	-0.56	-0.04	1.00	-0.03
Polarity flip	0.13	0.04	-0.00	-0.04	0.05	-0.02	0.19	0.28	-0.03	-0.03	1.00

The above diagram shows the correlation matrix of the features I have used. Mos of the features have a very low correlation between each other. Features like capital letters count and neutral words shows some high correlation which was unexpected. However, both the features are still important. This can be seen from the plot below.



The above plot shows the feature importance that were generated using random forest algorithm. It can be seen that the features such as polarity, capitalization, word polarity can lead to a greater reduction in the impurity function and hence these are important in judging the sarcastic content of the tweet.

VI. RESULTS

1. Random Forest Classifier

RF Parameters: No. of trees = 350

Confusion Matrix	Predicted (0)	Predicted (1)
Actual (0)	0.32	0.21
Actual (1)	0.19	0.28

- Accuracy : 0.60
- Precision: 0.58
- Recall: 0.59
- Fscore: 0.59

2. Support Vector Machine

Unigram features have also been included in this model.

Confusion Matrix	Predicted (0)	Predicted (1)
Actual (0)	0.30	0.22
Actual (1)	0.22	0.32

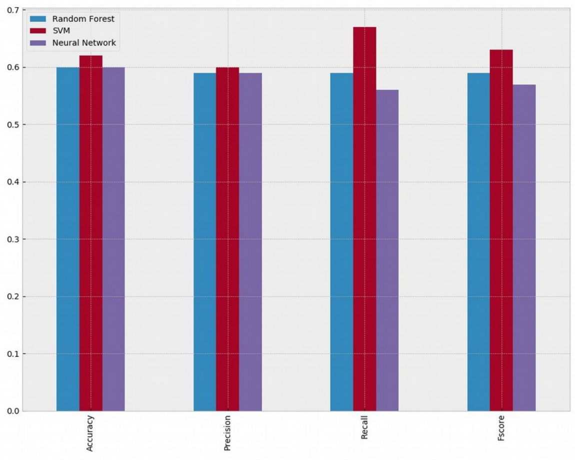
- Accuracy : 0.63
- Precision : 0.59
- Recall : 0.68
- Fscore : 0.63

3. Neural Network

Network Parameters : No. of Hidden layers = 1, No. of Neurons = 8

Confusion Matrix	Predicted (0)	Predicted (1)
Actual(0)	0.34	0.18
Actual (1)	0.22	0.26

VII. CONCLUSION



Deciding on the Accuracy, Precision, Recall and F-score, we observe that the SVM classifier performs better than the other classifiers.

Further insights from the Feature Importance Index reveal that features pertaining to polarity, emoticons, capitalization are important in judging the sarcastic nature of a tweet.

VIII.References

[1] Aditya Joshi, Vinita Sharma, Pushpak Bhattacharyya, “Harnessing Context Incongruity for Sarcasm Detection”, 7th International Joint Conference on Natural Language Processing. Vol. 2. 2015.

[2] Roger J. Kreuz, Gina M. Caucci, “Lexical Influences on the Perception of Sarcasm”, In proceddings to the workshop on computational approaches to figurative languages. Pages 1-4.

[3] Stephanie Lukin, Marilyn Walker, “Really? Well. Apparently Bootstrap Improves the performance of Sarcasm and Nastiness classifiers for online Dialogue”, Workshop on Natural Language Analysis in Social Media, pages 30-40.

[4] Riloff Ellen "Sarcasm as Contrast between a Positive Sentiment and Negative Situation." EMNLP. Vol. 13. 2013.

[5] Go, Alec, Richa Bhayani, and Lei Huang. "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford 1.12 (2009).

[6] https://github.com/AniSkywalker/SarcasmDetection/