SARCASM DETECTION IN TWEETS

Project Report
CS 725 : Machine Learning

Ashwin Bhat : 13D070006 Yash Bhalgat : 13D070014
Kalpesh Patil : 130040019 Navjot Singh : 130110071

May 31, 2017

1 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 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, its 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, 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.

2 Data

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, courtesy of [2].

3 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. 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.

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

- Lexical
- Pragmatic
- Linguistic Incongruity

The features used are described in further detail below :-

3.1 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 sarcastic tweets. Presence of such words can be a strong indicator for sarcasm. We use *unigrams* in order to extract the lexical information contained in the tweets.

Using the training corpus a dictionary is created. Each unique word is mapped onto a particular ID. These ID numbers are used as the feature numbers. The value 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. The dictionary would be large owing to the vocabulary available in the corpus. The tweet would contain only a few words from this large vocabulary. Hence, the feature vector would naturally contain a lot of 0’s corresponding to the words that are not appearing in the tweet but are present in the dictionary. The ID’s with value (frequency) 0 can be discarded since we are looking for presence of words prevalent in sarcastic tweets which can be a potentially important indicator while absence of words conveys no information here.

3.2 Pragmatic Features

These features are associated with the grammatical hints. Grammar plays an important role in any kind of language analysis. These include :-

3.2.1 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 extra impact. Hence this is used as a feature.

3.2.2 Number of emoticons

Emoticons are commonly used across social media platforms to express sentiments. As a feature, they can be captured using UTF-8 encoding. Using the 'codecs' module in Python, files containing emoticons in UTF-8 format can be opened and read. The emoticons present in them can be captured using regular expressions.

3.2.3 Number of slang laughter expressions

Popular slang expressions such as 'lol', 'rofl' and 'lmao' are fairly well used. Various variants of these has also been accounted for. The frequency of occurrence of these expressions is a feature. Since sarcasm is intended to have an element of humor, higher occurrence of these is potentially indicative of sarcasm.

3.2.4 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 a use of exclamation marks is prevalent especially that of '!', '?' and '. . .'. The count of such exclamation marks is hence used as a feature.

3.3 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 a 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.

As an example, consider the tweet : 'My tooth hurts. Yay!'. The negative word 'hurts' is incongruous to the positive situation hinted at by the word 'Yay' followed by an exclamation mark in order to emphasize it as well. So basically, explicit incongruity is an indication of contrasting emotions within the tweets, a hallmark of sarcasm.

The particular features used in this category are :-

3.3.1 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.

3.3.2 Number of words with positive and negative polarity

Using the Senti-strength tool, the polarity of each word is generated. The value generated lies in the range [-5,5]. If the value is positive, it is taken as a word with positive polarity. Similarly, if its negative its taken to be a word with negative polarity. Using these values two features are generated namely the total count of the words with positive and negative polarity.

3.3.3 Largest positive/negative sub-sequence

Length of the longest contiguous sequence of positive/negative features is used as a feature.

3.3.4 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 polarity is more likely to be sarcastic compared to a tweet with overall negative polarity. This is because in general sarcasm tends to be caustic.

Using the senti-strength tool [5], we get the polarity of each word as well as the overall sentence. For the sentence, a polarity score of -1 to -5 is considered to be negative polarity sentence, 0 is taken to be neutral while a score of +1 to +5 is taken to be of positive polarity.

4 Evaluation

4.1 Feature Analysis

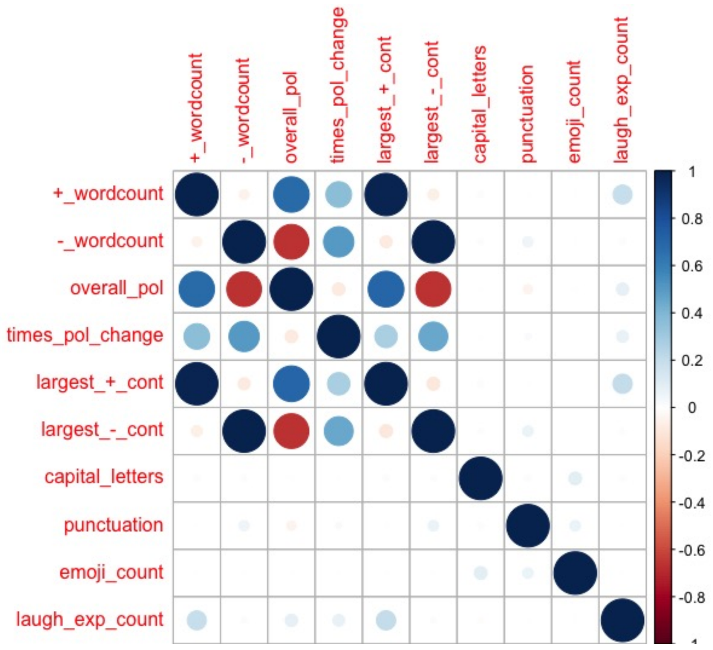


Figure 1: Correlation matrix for the features

The above diagram shows the correlation matrix for the features we have used. Most of the features have very low correlation between each other which is seen from the small size of the dot. Features like total word count and max contiguous positive/negative polarity word count shows some high correlation which is expected. However, both features are still important.

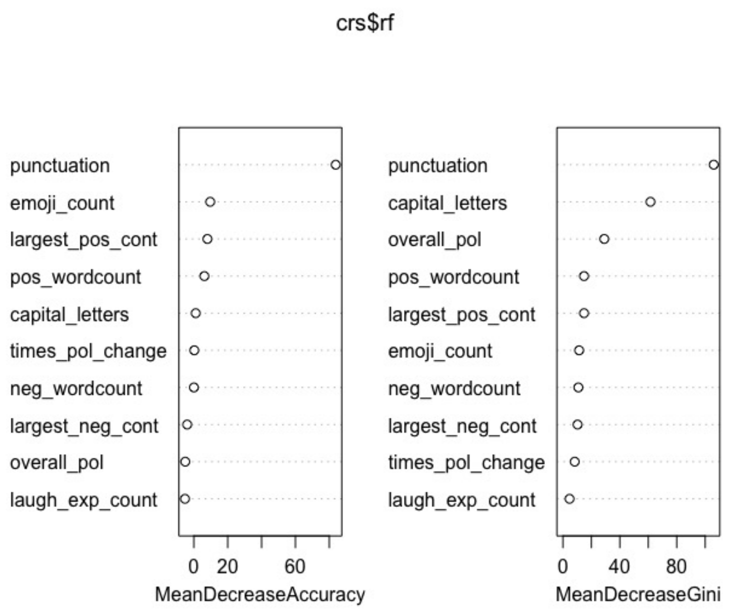


Figure 2: Feature Importance using Random Forest

From the above plot we can see that features like punctuation and capitalization lead to greater reduction in the impurity function value and hence these are important is judging the sarcastic content of a tweet.

4.2 Random Forest Classifier

(Trained using Rattle [4])
RF Parameters : No. of trees = 500 , Variables = 3

Confusion Matrix	Predicted (0)	Predicted (1)
Actual (0)	0.44	0.13
Actual(1)	0.20	0.23

- Accuracy : 0.67
- Precision : 0.62
- Recall : 0.53
- F-measure : 0.59

4.3 Neural Network

NN Parameters : No. of Hidden Layers = 1 , No. of Neurons = 5

Confusion Matrix	Predicted (0)	Predicted (1)
Actual (0)	0.20	0.19
Actual(1)	0.09	0.52

- Accuracy : 0.72
- Precision : 0.73
- Recall : 0.85
- F-measure : 0.78

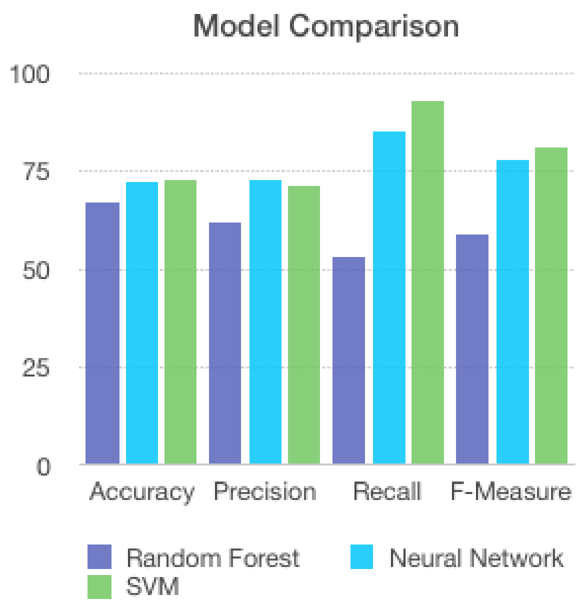
4.4 SVM Classifier

Unigram features have also been included for this model.

Confusion Matrix	Predicted (0)	Predicted (1)
Actual (0)	0.15	0.23
Actual(1)	0.04	0.58

- Accuracy : 0.73
- Precision : 0.71
- Recall : 0.93
- F-measure : 0.81

5 Conclusion



Deciding on the Accuracy, Precision, Recall and F-values, we observe that the SVM classifier (with bag of words) performs better than the other classifiers. Including these unigram features to a Neural Network (using [6]) was not practical due to the high dimensionality of feature vectors involved. This was however practical with the LibSVM library (as in [3])

The Random Forest was trained using the rattle library from R (as in [4])

Further insights from the Feature Importance index (based on decrease in Gini Impurity) reveal that features pertaining to Punctuation (e.g no of '!') , laughter expressions (e.g lol, lmao) and emoticons are important in judging the sarcastic content of a tweet.

Github Repository for codebase:
https://github.com/navisngh11/Sarcasm-Detection-Twitter

References

[1] Stream data using Twitter API
`https://github.com/guyz/twitter-sentiment-dataset`

[2] Sentiment dataset for Sarcasm detection
`https://github.com/dmitryvinn/twitter-sarcasm-measurement`

[3] LibSVM : A Library for Support Vector machines
`http://www.csie.ntu.edu.tw/~cjlin/libsvm/`

[4] Rattle: A Graphical User Interface for Data Mining using R
`http://rattle.togaware.com/`

[5] Senti-Strength Tool
Automatic sentiment analysis of word corpus.
`http://sentistrength.wlv.ac.uk/`

[6] Keras: Deep Learning library for Theano and TensorFlow
`https://keras.io/`

[7] Cliche, M. The sarcasm detector, 2014.
`http://www.thesarcasmdetector.com/about/`

[8] Bharti, Santosh Kumar, Korra Sathya Babu, and Sanjay Kumar Jena. ”Parsing-based sarcasm sentiment recognition in Twitter data.” 2015 IEEE/ACM International Conference on Advances in Social Networks Analysis and Mining (ASONAM). IEEE, 2015.

[9] Joshi, Aditya, Vinita Sharma, and 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. Vol. 2. 2015.
`https://www.aclweb.org/anthology/P/P15/P15-2124.pdf`