# Sarcasm Identification in Social Network Data[1]

**Satheeshkumar Karuppusamy**
Viterbi School of Engineering
University of Southern California
Los Angeles,CA-90089
karuppus@usc.edu

## Abstract

Sarcasm can be defined as the concept of delivering a message which needs more intuitive interpretation for understanding the actual underlying meaning. In recent years, the remarkable growth in Social media has paved the way for the development of large number of Opining mining and Web Sentiment Analysis systems which exploits the user generated data from social media to read the pulse of people towards the participating entities. Certain fraction of the performance of Opining mining systems is degraded by the apathy of these systems towards Negation sentences especially sarcastic data. Classification based approach to detect sarcastic messages in social media data have been proposed and evaluated in this report.

## 1 Introduction

Sarcasm identification is an open area in NLP which has been getting enormous attention due to the importance that has been given to analyzing sentiments and opinion in social media in recent years. Since sarcastic content tends to be more subjective, it is more difficult to find sarcastic utterances, especially in text utterances. Voice message has heuristics like time break and vocal cues which could be exploited to detect sarcasm. But identifying sarcasm in text messages is too tough to solve.

It requires large amount of human effort to annotate the training data required for these classifiers. Human annotators themselves tend to disagree in large cases in annotating sarcastic data. The popularity of social media sites like Twitter and the tagging made by the users with hash tags ('#') provide us a way to automatically generate  large amount of training data for tasks like these. This reduces the time and cost spent in human annotation and almost yields comparatively better annotation than the manually annotated content.

We are approaching sarcasm identification as a classification problem and have deeply analyzed the issues and performance of machine learning classifiers in discriminating sarcastic utterances, from positive and negative utterances. The two classifiers namely Negative vs Sarcastic Classifier and Positive vs Sarcastic Classifier are built and tuned with rich feature sets. Those feature sets are used in classifying sarcastic utterances using Sarcastic vs Non-sarcastic and Positive vs Negative vs Sarcastic Classifier. The similar kind of approach has been followed by (Roberto, 2011) but with less number of features and with smaller number of manually filtered dataset. In our approach, SVM Classification algorithm with both linear and poly Kernel, Naïve Bayes Classification Algorithm, SMO Classification algorithm and Decision Trees Algorithm with 10-Fold cross validation are being employed for this task.

## 2 Related Work

Sarcasm identification in written text has not been given that much importance until the evolution of Social media. Couple of works has been done in Sarcasm Recognition in recent times with quite a remarkable success. The Work done by Dmitry et.al (2010) in recognizing sarcasm from Amazon Product review dataset by modeling the patterns which invokes sarcasm has yielded the F-score of about 0.78. Patterns are defined as a set of words in the sentence with particular proportion of High and low frequency words.(Dmitry et al.,2010) Those patterns which adhere to the above rule which invokes sarcasm are automatically identified by their algorithm. Once the patterns are obtained, the features for patterns in the training set has been chosen as one among four possible values exact match,

[1] ***This paper has been submitted as a final project for the course CSCI-544 Applied Natural Language Processing at University of Southern California. This course was taught by Dr. Zornitsa Kozareva during Spring 2013 semester.***

Sparse match, incomplete match and no match(Dmitry et.al, 2010). The learned patterns from the Amazon dataset are then applied on the twitter data which seems to work good as well, which result in yielding 0.83. They have used Amazon Mechanical Turk for annotating 66K Amazon Product review dataset and about 5.9 million tweets. The issue with this system is that remarkable human cost and time is being spent for annotating the dataset. Social Media like Twitter contains the self-annotated data by the people who meant sarcasm which could be effectively used. Another work by Roberto et.al (2011) deals mainly with this issue. It employs the idea of utilizing the self-annotated data from twitter for training purpose and applying it to recognize the sarcastic tweets. But they have used only very small dataset of 3000 tweets and they have filtered the tweets at the step one which breaks their unsupervised nature of data preparation. Features and algorithms tried are very minimal on the dataset of about 3000 tweets which is considered to be too small in the context of giganticness of the Social media.

## 3 Data Preparation

Our training data for sarcastic class is the tweets self-annotated by the tweeters with the hash tags '#sarcastic' and '#sarcasm' fetched from [3]http://www.twitter.com/ using Twitter API. About 20K tweets are collected from twitter for about 6 days using twitter Streaming API. The tweets that end with the hash tags '#sarcastic' or '#sarcasm' are treated as sarcastic tweets and the rest of the tweets which has the hash tags in the middle are ignored. Upon filtering those 20000 tweets, we had ended up with almost less than 10000 tweets. We also have a collection of about 5000 positive and negative tweets each which we have downloaded for Sentiment Analysis project. Positive tweets are those which contain hash tags like '#happy' and '#joy' at the end. Negative tweets are those with hash tags like '#sad' at the end of the tweet. We are using this combination of dataset for our training purpose. Twitter test dataset of 3000 tweets, 1000 of each class has been prepared in the same way as the twitter training dataset in prepared.

Certain pre-processing steps are applied upon the test and the train dataset. The URL content in the tweets is replaced with the unique token 'DEFAULT_URL'. The '@' user mentions in the tweets are replaced with the unique token '@DEFAULT_USER'. The words which refer to the class hash-tags are removed and the words which contain hash-tags are stripped. Along with the twitter dataset Sarcastic statuses from face book are extract from [2] http://www.searchquotes.com/ and some other websites. Since face book is not an open ended system like twitter, it is very difficult to collect face book statuses which evokes certain emotion. It has to be collected manually by carefully looking at the keyword with which the status is tagged and have to annotate with that particular class it belongs. Facebook status test corpus with 368 statuses is formed. It has 218 sarcastic statuses, 50 negative statuses and 100 positive statuses. Those were manually annotated and the annotations are compared with the existing tags. This yielded me the Kappa agreement of 0.61 which is comparably very low. Sarcasm can be identified only by people with high level proficiency in the language and most of the time it is subjective. This makes me to select the sarcastic face book statuses just by based on the keyword with which it is tagged. The main difference in twitter and face book corpus is that because of no limitation in the length of the characters to be posted, face book statuses are too grammatical than what we have in twitter with longer length words and sentence. Most of the sarcastic tweets are Pseudo positive tweets which expresses seems to give positive sense about the context but they actually mean negative sense. Some are in the other category as well, Pseudo Negative tweets.

### 3.1 Description of Other Tools and Data Collections

In addition to the train and the test data, collection of word lists and emoticons are prepared as well. This section will give a brief overview on the data preparation process adopted in those tasks.

### 3.2 NREC Dictionary

The NRC emotion lexicon is a list of words and their associations with eight emotions (anger, fear, anticipation, trust, surprise, sadness, joy, and disgust) and two sentiments (negative and positive). The annotations were manually done through Amazon's Mechanical Turk. This has been prepared as a part of Research work by the National Research Council Canada (Saif et.al, 2011). It has the coverage of about 3787 words in train data and 1712 words in test data which is

[2] http://www.searchquotes.com/

3 http://www.twitter.com/

[4] http://en.wikipedia.org/wiki/List_of_emoticons

little higher than the other data sources like Senti-WordNet.

### 3.3 Interjection List

We believe that interjections form a major part in evoking sarcasm and emotions in twitter data. So we have manually scraped the interjections in the website and formed a dictionary of interjections. Interjection list of about 101 elements are prepared. There were about 1377 tweets in train data and 250 tweets in test dataset which contain one among those interjections in the list.

Example:
Aw, hurray, jeez, oh, yeeaah, ….

### 3.4 Emoticons List

We have manually scraped the http://en.wikipedia.org/wiki/List_of_emoticons[4] and some other pages and formed a list of emoticons. Sometimes winking emoticon evokes positive sense and sometimes it evokes sarcastic sense. So it would not be good if we classify that winking emoticon evokes positive sense always. So we have generally classified the emoticons into 13 different categories. Smiling and Sad emoticons are the commonly occurring emoticons in the tweets.

## 4 Description of Feature sets Used

Different sets of feature are being used in this project. They are primarily classified into 4 different categories namely Lexical Features, Pragmatic Features, Dictionary based polarity features, Topic based features. The idea of dividing the features into Lexical and Pragmatic features, even though our description of those features varies, is adopted from the work of Roberto et.al, 2011.

### 4.1 Lexical Features

We are using three kinds of features which describe the lexical information of the content in the tweets such as Unigrams, Bigrams and trigrams. Since phrases form an important part in evoking sarcasm than individual word based features we gave more importance to forming phrase based features.

### 4.2 Pragmatic Features

In Roberto et. al (2011), it has been proved that '@' Reply tags forms a major part in evoking and detecting sarcasm in twitter data. We have used this as a numeric features which represents the count of '@' reply tags that is being used in the current tweet. In addition to that the presence of '?', '!' are also treated as separate binary features. The presence of the words which are in the list of interjections and the emoticons are also treated as other binary features.

### 4.3 Dictionary and Part of Speech based Polarity Features

Sarcastic tweets often tend to have words of both positive and negative polarity. Finding the words with opposite polarity is itself gives us an idea that the tweet may be sarcastic. This technique has been employed by Apoorv et.al (2011) in analyzing sentiments of the tweets. Number of words representing each emotion is labeled and counted for each tweet and represented as a feature. Count of Positive and Negative polarity words across each tweet, total emotion in a tweet are used as separate features in our system.

### 4.4 Topic Based Features

Sarcasm would also be based on topics. The possibility of Sarcasm being evoked when a speaker talks about some topic is relatively more. This idea is being employed in incorporating this feature. Topic probabilities for each tweet are identified and the feature values of top 'n' topics are set to 1 and the rest are set to 0. Features equivalent to number of topics - binary valued (top 'n' possible topics gets 1 and others get 0)

## 5 Our Approach

Since sarcastic sentence implicitly invokes positive or negative emotion, the sarcasm identification system can be improved only by improving the accuracy of classifying positive and negative tweets from sarcastic tweets. We have designed two separate classifiers one for observing the behavior of sarcastic tweets along with positive tweets and the other for observing the behavior of sarcastic tweets with negative tweets. The feature sets that better distinguish sarcastic tweets from positive and negative tweets are found and the combination of those feature sets is then applied to our third set of classifiers to classify sarcastic tweets. Two different types of classifiers one to treat this problem as a binary classification problem, sarcastic and non- sarcastic tweets

[2] http://www.searchquotes.com/
3 http://www.twitter.com/
[4] http://en.wikipedia.org/wiki/List_of_emoticons

and the other is to treat this as a multi-class classification problem with positive, negative and sarcasm class.

## 6 Baseline System

The Baseline system we have considered is a multiclass Positive vs Negative vs Sarcasm classification system with only unigram features. Linear SVM algorithm is applied on the input tweet corpus and their accuracies are as given below:

| System | Unigrams Alone | | |
|---|---|---|---|
| | Precision | Recall | F-Score |
| Positive | 0.31 | 0.051 | 0.0875 |
| Negative | 0.32 | 0.21 | 0.253 |
| Sarcasm | 0.32 | 0.759 | 0.475 |

Table 1: Baseline System

## 7 Description of Our Approach

Two classifiers namely Positive vs Sarcasm, Negative vs Sarcasm are designed to analyze the impact of Positive and Negative sentences in classifying Sarcastic sentences. Classifying the sarcastic sentences from negative sentences is not that much efficient and needs much innovative features. Finally two types of classifiers namely Sarcasm vs Non-Sarcasm and Sarcasm vs Positive vs Negative classifiers are designed and implemented to detect sarcasm in the given input corpus of tweets. The Machine Learning tools like SVM Multiclass and Weka are being used for this project. Linear SVM model is generated using SVM Light and SVM multiclass tools. Poly SVM, Naïve Bayes, Decision Trees and SMO algorithms are applied with feature selection using Weka.

### 7.1 Feature Selection

Initially we had about 40,000 N gram features, 100 topic based features, around 20 to 30 pragmatic features which include count of emoticons, reply features, count of interjections and so on along with few dictionary based features. The feature space was too large that it was too time consuming so we could not apply any machine learning classification algorithms directly on this. Dimensionality Reduction is done by applying Principal Component Analysis feature Selection Algorithm. Given below is an example of best performing feature combination after applying PCA:

-0.504interjectionFeature=0-0.483reply_feature
=0+0.472interjectionFeature
=1+0.441reply_feature=
1+0.159exclamationFeature…

## 8 Experiments and Results

The baseline system was implemented using SVM Multiclass which does not support Dimensionality techniques. Even though the input matrix is a sparse representation of actual feature-set, the system is not that much efficient since it uses Linear Kernel. The recall rate of the sarcastic class is very low while applying SVM Linear Kernel. After applying PCA to the input matrix, Naïve Bayes, SVM Poly Kernel, SMO Classifier and Decision Tree classifier algorithms are applied and evaluated with the test set. SVM Poly Kernel, SMO Classifier and Decision Tree algorithms perform equally better. The results of top two systems for Positive vs Sarcasm Classifiers are as given below:

| SVM Poly Kernel | | | | |
|---|---|---|---|---|
| | Precision | Recall | F-Score | Accuracy |
| Negative | 0.579 | 0.756 | 0.656 | 0.603 |
| Sarcasm | 0.648 | 0.45 | 0.531 | |
| Total | 0.614 | 0.603 | 0.593 | |
| SMO Classifier | | | | |
| Negative | 0.551 | 0.888 | 0.68 | 0.5815 |
| Sarcasm | 0.711 | 0.275 | 0.397 | |
| Total | 0.631 | 0.582 | 0.538 | |

Table 2: Negative vs Sarcasm Classifier

The same algorithms are applied for studying the performance of classifying positive and sarcastic tweets and their results are given as below:

| SVM Poly Kernel | | | | |
|---|---|---|---|---|
| | Precision | Recall | F-Score | Accuracy |
| Positive | 0.599 | 0.816 | 0.691 | 0.6345 |
| Sarcasm | 0.711 | 0.453 | 0.553 | |
| Total | 0.655 | 0.635 | 0.622 | |
| SMO Classifier | | | | |
| Positive | 0.593 | 0.766 | 0.668 | 0.62 |
| Sarcasm | 0.669 | 0.474 | 0.555 | |
| Total | 0.631 | 0.62 | 0.612 | |

Table 2: Positive vs Sarcasm Classifier

[2] http://www.searchquotes.com/

3 http://www.twitter.com/

[4] http://en.wikipedia.org/wiki/List_of_emoticons

The feature sets and algorithms used in previous publications stated that it is more difficult to discriminate negative sentences from sarcastic sentences than discriminating positive sentences. But from our above results it could be little clear that discriminating positive and negative sentences from sarcastic sentences are equally hard.

The main objective of our project is to discriminate the non-sarcastic sentences from the sarcastic sentences which have been achieved by developing two kinds of Classifiers whose results are given as below:

| **SVM Poly Kernel** | | | | |
|---|---|---|---|---|
| | **Precision** | **Recall** | **F-Score** | **Accuracy** |
| Non-Sarcasm | 0.71 | 0.88 | 0.78 | 0.68 |
| Sarcasm | 0.54 | 0.28 | 0.37 | |
| Total | 0.65 | 0.68 | 0.64 | |
| **SMO Classifier** | | | | |
| Non-Sarcasm | 0.70 | 0.89 | 0.78 | 0.68 |
| Sarcasm | 0.52 | 0.23 | 0.32 | |
| Total | 0.64 | 0.67 | 0.63 | |

Table 3: Non-Sarcasm vs Sarcasm Classifier

The result of three way classification for sarcasm detection is as given below:

| **SVM Poly Kernel** | | | | |
|---|---|---|---|---|
| | **Precision** | **Recall** | **F-Score** | **Accuracy** |
| Positive | 0.63 | 0.35 | 0.45 | 0.47 |
| Negative | 0.53 | 0.41 | 0.75 | |
| Sarcasm | 0.52 | 0.33 | 0.40 | |
| Total | 0.52 | 0.47 | 0.46 | |
| **SMO Classifier** | | | | |
| Positive | 0.63 | 0.32 | 0.42 | 0.46 |
| Negative | 0.41 | 0.84 | 0.55 | |
| Sarcasm | 0.52 | 0.23 | 0.32 | |
| Total | 0.52 | 0.46 | 0.437 | |

Table 4: Positive vs Negative vs Sarcasm Classifier

In our case two way classification algorithms performs better in terms of Accuracy in classifying each tweet into Sarcastic or Non-Sarcastic tweets. But in terms of F-score, three way classification systems are better than the two way classification system. Using Facebook statuses as test data in the above settings results in very low recall and precision since there is large variation in the vocabulary of communication and the data we have collected doesn't have emoticons in them or reply user tag which were the primary features in sarcasm identification.

## 9 Performance comparison with Existing Systems

The classification accuracy is used as a metric in [2] and their best performance is around 70% whereas the accuracy of our best system is around 68%. It should be noted that they have manually filtered small amount of tweets to avoid noise in twitter data. And also Accuracy alone could not be a best evaluation metric in systems like these. Even a system with good accuracy can have low recall for sarcastic class. So we have used F-score metric along with Accuracy metric. From Table 3 and Table 4 we could understand that recall for sarcastic tweets are always less when compared to non-sarcastic tweets. Our system is better in terms of F-score as well. Best system with SVM Poly Kernel Classifier fetches the F-score of about 64%. When considering the recall of the sarcastic tweets as a metric, our three way classifier with SVM Poly kernel achieves about 40% recall rate.

## 10 Conclusion

In this project, we have analyzed, studied and implemented a sarcasm identification system which takes content from manually annotated Social media stream and identifies sarcastic content from them. Rich Feature sets and variety of Machine Learning algorithms are applied for this problem and the performance of the system towards discriminating Sarcastic tweets from Positive and Negative sentences are observed. The best system in our project gets around 68% of accuracy which is in par with the current best performing systems. We can exploit the manual annotation in Social data for tasks like Sarcasm identification more effectively since there is always some subjectivity in sarcasm which reduces the accuracy of human annotations. Using Rich set of features and Machine Learning algorithms will get us efficient and affordable results. Cross domain applicability of this Sarcasm identification technique is limited because of large difference in vocabulary of the language used in different social medium.

[2] http://www.searchquotes.com/

3 http://www.twitter.com/

[4] http://en.wikipedia.org/wiki/List_of_emoticons

## References

Roberto González-Ibáñez, Smaranda Muresan, Nina Wacholder, 2011. Identifying Sarcasm in Twitter: A Closer Look, in the Proceedings of the 49th Annual Meeting of the Association for Computational Linguistics

Dmitry Davidov, Oren Tsur, Ari Rappoport, 2010. Semi-Supervised Recognition of Sarcastic Sentences in Twitter and Amazon, in the Proceedings of Fourteenth Conference on Computational Natural Language Learning

Saif M. Mohammad, Peter D. Turney, 2010. Crowdsourcing a word-emotion association lexicon

Marilyn A. Walker, Pranav Anand, Robert Abbott and Ricky Grant, Stance Classification using Dialogic Properties of Persuasion

Approv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, Rebecca Passonneau, 2011, In the proceedings of Workshop on Languages in Social Media, Sentiment Analysis of Twitter Data

Mark Hall, Eibe Frank, Geoffrey Holmes, Bernhard Pfahringer, Peter Reutemann, Ian H. Witten (2009); The WEKA Data Mining Software: An Update; SIGKDD Explorations, Volume 11, Issue 1.

McCallum, Andrew Kachites. "MALLET: A Machine Learning for Language Toolkit." http://mallet.cs.umass.edu. 2002.

Kristina Toutanova, Dan Klein, Christopher Manning, and Yoram Singer. 2003. Feature-Rich Part-of-Speech Tagging with a Cyclic Dependency Network. In Proceedings of HLT-NAACL 2003, pp. 252-259.

Olutobi Owoputi, Brendan O'Connor, Chris Dyer, Kevin Gimpel, Nathan Schneider and Noah A. Smith. Improved Part-of-Speech Tagging for Online Conversational Text with Word Clusters. In *Proceedings of NAACL 2013*.

[2] http://www.searchquotes.com/
3 http://www.twitter.com/
[4] http://en.wikipedia.org/wiki/List_of_emoticons