ISSN:0975-9646

Tweet Sarcasm: Mechanism of Sarcasm Detection in Twitter

Komalpreet Kaur Bindra [#1], Asst Prof Ankita Gupta [*2]

[#] *Computer Science Department, PEC University of Technology, Chandigarh, India.*

[*] *PEC University of Technology, Chandigarh, India.*

Abstract **– The www is growing at an alarming rate . Users have started participating actively on Internet by giving their opinions on products, services and blogs. Study and analysis of such opinions is known as Opinion Mining. But sometimes users prefer being sarcastic. Sarcasm is a linguistic phenomenon in which people state the opposite of what they actually mean. Sarcasm Detection is a challenging task, even for humans. It is a part of Opinion Mining which is studied so that different sentiments can be analyzed and worked upon. In this paper, the overall mechanism of sarcasm detection is seen, which is rooted on the data from Twitter, a popular micro blogging service.**

Keywords **– hashtags, linguistics, opinion mining, sarcasm detection, tweets**

I. Introduction

Textual data can be divided into two categories, facts and opinions. Facts are objective statements whereas Opinions are subjective statements. Facts state the events that were happened in the world. Opinions indicate the different sentiments, perceptions, observations or views about those events. What others think has always been an important and interesting information for most of us in decision-making process. Opinion Mining in any business or organization can be thought of as-

- When a person wants to buy a phone Looks for comments and reviews
- A person who just bought a phone Comments on it Writes about their experience
- A Phone Manufacturer Gets feedback from customer Improve their products Adjust Marketing Strategies

When it comes to sentiments or emotions no one is concerned about the topic of the text but focuses on its positive or negative expressions. People can easily express their opinions on social media services such as reviews, blogs, social networking sites as they provide a huge amount of valuable information. Now-a-days automated identification of sentiments is done which is beneficial for many NLP systems like review summarization systems (SMO), dialogue systems and public media analysis systems. Majorly the existing sentiment extraction systems are based on polarity identification (e.g., positive vs. negative reviews), but there are many useful and comparatively unexplored sentiment types such as sarcasm, irony or humor. In this paper, the sentiment sarcasm has been explored and its detection has been done on Twitter, as a platform. With the recent trend of tagging posts using HASHTAGS, some social media services like Twitter allow users to add different hashtags to articles/tweets. For this reason blogs are used as a large dataset for sentiment learning and identification. In this paper, different Twitter tags are used as sentiment labels. Different punctuations, words, patterns in the text are observed for detecting sarcasm.

II. Related Work

Sarcasm and irony are well-studied and emerging concepts in linguistics, psychology and cognitive science[1]. But in the opinion mining literature, among all these concepts, automation of sarcasm detection is examined as a difficult problem and has been approached in only a few studies[2]. Sentiment analysis tasks consists of two major steps- (1) Looking for different expressions, and (2) determining the polarity (negative, positive or neutral) of the expressed sentiment. These steps are generally performed to check whether a sentence conveys positive meaning or negative. But in this paper, sarcastic and non-sarcastic tweets are distinguished to find the polarity of a sentence.

It has been proposed that sentiment words or phrases might have different senses thus word sense disambiguation can improve analysis of sentiments[3]. All mentioned work identifies evaluative sentiment expressions and their polarity. However, it has been noted that in many cases , simply a sentence cannot be judged as sarcastic or non sarcastic without the surrounding text or content. For example, the sentence "*Where am I?*" can be assumed as sarcastic only if it is known that it is stated in a review of a GPS device. Also, in some cases, only analyzing few sentences together can reveal the presence of sarcasm.

III. Data

To form an algorithm for detecting sarcasm, first we need to train that algorithm and we require data for that. Classification is a directed learning job, which means, for the classifier to know the difference between different sentences, some sentences labeled as sarcastic and others labeled as non-sarcastic are needed. It can be done by using an online corpus which contains various sarcastic sentences, for example reviews, comments, posts etc and labelling is done. But this is very monotonous exercise in case of large data set. Another option is to make use of the Twitter API to club tweets with the label #sarcasm or #sarcastic, these will be the sarcastic tweets, and others that don't have such label, will become non-sarcastic tweets.

A. Hashtags

In Twitter, a message can be of about 140 characters . Except the normal text, a twitter message can contain references to other users (@<user>), hashtags (#hashtag) and URLs . For example, : "*@personA Check out @personB for amazing ideas :) http://xxxxxx.com #happy #hour*"[4]. So for building the corpus of sarcastic (S), negative (N) and positive(P) tweets, the annotations that tweeters assign to their tweets using hashtags are used. Twitter API is used to collect tweets that include hashtags of sarcasm (#sarcastic, #sarcasm), direct positive sentiment (e.g., #happy, #joy, #lucky, #amazing, #exiting), and direct negative sentiment (e.g., #sadness, #angry, #frustrated, #bad, #fail) [5]. Also, automatic filtering is applied to remove quotes, spam, duplicates , tweets written in languages other than English. The advantage of using Twitter API is that we can have enough samples to fulfill our requirement. Every day people write tweets, use sarcasm, that can be easily collected, clubbed and stored in a database. But there's a drawback in collecting data from Twitter, that is, the data is little noisy! People also use the #sarcasm hashtag to show that the tweet is sarcastic, but a Human cannot simply guess or assume that the tweet is sarcastic without the label #sarcasm . So for this we need to pre-process the data i.e cleaning up the data.

For doing this, all the tweets which contain Non-ASCII characters, link to other tweets and non sarcastic behaviour, are removed. After that all the hashtags and all occurences of the word sarcasm or sarcastic are removed from the remaining tweets. And still if the tweet is atleast 3 words long, it is added to the dataset [6]. The above is done to remove all the noise.

B. Feature Engineering

Different steps were adopted for doing feature extractions.

1) n-grams: It can be bigrams and unigrams. These are group of single word (example: *seriously*, *great*, *amazing*, etc.) and double words (example: *really bad*, *super amazing*, *very good*, etc.). To extract them from the remaining text, each tweet is passed through tokenization, stemming , uncapitalization and then each and every n-gram is added to a binary feature dictionary[7]. Tokenization is a task of cutting through the character sequence into bits and pieces, called tokens. Eg- "That is the old file" ,$w(s_a)$ = {"that", "is", "the", "old", "file"}[8]. Stemming is used in linguistic morphology and retrieving information to describe the process of reducing inflected words to their word stem, base form or root. Eg: words fishing, fisher, fished has the same root word 'Fish'[9].

2) Sentiments: Sarcastic tweets are observed to be more negative than non-sarcastic tweets. Also, there is a large variety of sentiments in tweets that are sarcastic. It starts with a quite positive sentiment and ends with a negative sentiment (example: *I enjoy getting slapped #sarcasm*)[10]. So for this a sentence is broken into parts and sentiment analyzers are used on each part separately. Many research work has been done on these analyzers. There's an analyzer made called, SentiWordNet dictionary. It gives a positive and a negative sentiment value to every single word of the English language. By looking for the words in the dictionary, sentiment value can be given to every single part of the tweet. Another deployment of this sentiment analysis can use the python library tool TextBlob which contains a built-in sentiment ranking function.

3) Pattern Extraction : For automatic extraction of patterns, the definitions about patterns provided by Davidov and Rappoport, are used. Words are categorized into content words (CWs) and high-frequency words (HFWs). A word having more(less) corpus frequency or occurence than *FH* (*FC*) is said to be a HFW (CW)[11]. All single punctuation characters or their consecutive sequences are considered as HFWs. A pattern is said to be an directed sequence of high frequency words and some slots for (CWs) content words.

C. Classification Algorithm

The investigation on the applicabality of pragmatic and lexical features in machine learning is done to classify different positive, negative and sarcastic Tweets . Two standard classifiers that are generally used in sentiment classification are: logistic regression (LogR) and support vector machine with sequential minimal optimization (SMO). In machine learning**,** support vector machines are supervised learning models having associated learning algorithms (SMO) which can analyze data and recognize patterns[12]. Whereas, **Logistic Regression** is a regression in which *binary response variable* is related to a **set of explanatory variables**, that can be discrete or continuous[13].

IV. EVALUATION

The main purpose of evaluating is to learn how well the framework helps in identification and differentiation of sentiments defined by tags and to test if the framework can be successfully used to identify sentiments in new untagged sentences[14]. It can be done using cross-validation technique. **Cross-validation**, is a technique for assessing the results of a statistical analysis. It is mainly used in settings where the goal is prediction, and one wants to estimate how accurately a predictive model will perform in practice[15]. The metric used to explain the cross-validation is the F-score. It is a useful metric when there are more samples from one particular category than from the other . For example, if we have 10 times more non-sarcastic messages or tweets than sarcastic then this type of metric is considered.

Since precision is needed so some other metric is required. Precision defines the number of rightly identified sarcastic tweets upon the total number of tweets that are sarcastic, whereas recall gives the number of sarcastic tweets rightly identified upon the total number of sarcastic tweets in the cross validation set. Both recall and precision are better scores or ranking methods to quantify the quality of a sarcasm classifier. The F-score provides the harmonic mean of precision and recall[16].

To gain an insight into what the algorithm has acquired, different feature coefficients in the trained SVM are noticed to see the most important ones. According to a survey, for n-grams, the features that are important to classify them into sarcastic tweets are *how, what*, *perfect place*, *a blast*, *shocking* , *just eat*, etc. The most relevant n-grams for the classification of non-sarcastic tweets are *feeling great*, *spend more*, *praying*, *too funny*, *be smart*, *goodnight*, etc[17]. Finally,it was observed by the classifier that sarcastic tweets are more about expressing emotions and feelings, either positive or negative, than non-sarcastic ones.

V. CONCLUSION

Sarcasm detection is a really fascinating subject. It evaluates diverse feature types for sentiment extraction including sentiments, words, patterns and n-grams, confirming that each feature type contributes to the sentiment classification framework. As we have seen that it is feasible to do sarcasm detection using NLP tools, one quick and easy way to improve this detector is to use a spell corrector along with, for the tweets. This would help in minimizing the order of dimensions of the dictionary for the n-gram features and will improve the sentiment analysis operation as well. In the future, these methods can be applied for automated clustering of sentiment types and sentiment dependency rules and can be expanded to detect some other non-literal form of sentiments like humor.

REFERENCES

[1] Gibbs, 1986; Gibbs & Colston 2007; Kreuz and Gluckberg, 1989; Utsumi, 2002

[2] Nigam & Hurst, 2006 and Pang and Lee, 2008 for an overview

[3] Esuli and Sebastiani, 2006; Andreevskaia and Bergler, 2006; Wiebe and Mihalcea, 2006

[4],[5] Identifying Sarcasm in Twitter: Roberto González-Ibáñez Smaranda Muresan Nina Wacholder School of Communication & Information Rutgers

[6],[7] http://www.thesarcasmdetector.com/about

[8] http://nlp.stanford.edu/IRbook/html/ htmledition/tokenization-1.html

[9] http://nlp.stanford.edu/IRbook/html /htmledition/stemming-and-lemmatization-1.html

[10] http://www.thesarcasmdetector.com

[11] Semi-Supervised Recognition of Sarcasm in Twitter and Amazon, Dmitry Davidov, Oren Tsur, Ari Rappoport

[12] https://en.wikipedia.org/wiki/Support_vector_machine

[13] https://en.wikipedia.org/wiki/Logistic_regression

[14] p241-davidov.pdf

[15] https://en.wikipedia.org/wiki/Cross-validation_(statistics)

[16],[17] http://www.thesarcasmdetector.com