Imperial Journal of Interdisciplinary Research (IJIR)
Vol-3, Issue-5, 2017
ISSN: 2454-1362, http://www.onlinejournal.in

Sarcasm detection using combinational Logic and Naïve Bayes Algorithm

Rathan K[1] & Suchithra R [2]
[1]Student, Department of MSc IT Jain university-SCMS, Bangalore, India.
[2]Head Of Department(HOD), Department of MSc IT Jain university-SCMS, Bangalore, India

Abstract: *One of the challenges in Sentimental Analysis is the presence of Sarcasm. Users have actively participated on internet by sharing their opinions, views about the services, products, blogs, political issues and entertainment. For humans it's very difficult to consistently identify a subjective vocal inflection. Sarcasm detection without vocal cues is very complicated task in hand. The existing state of art solutions in sentimental analysis and sarcasm scope detection has been investigated. Moreover, a corpus of social media data with linguistic negation has been developed and an enhanced framework for sarcasm detection has been developed and assessed. Current works, approach this research issue basically from only linguistic point of view point of view. And the results are not accurate by this means making it a sour algorithm to detect sarcasm sentiment. Hence, we prove that by including a combined approach of hyperbole, emoticons, lexical analysis, contrast we can achieve better accuracy when compared to usage of only linguistic features. The paper also concentrated on Naïve Bayes approach for feature extraction and classification. Some new schemes have been introduced in this paper which is a sequential and combinational logic scheme for sarcasm detection in twitter data. The sequential scheme is called hierarchical approach and combinational logic is called Hybrid approach. For hybrid approach an upper level and lower level features are determined for decision control and classification of data.*

Keywords: *Sarcasm detection, Naïve Bayes Algorithm, Combinational logic, Hyperbole Approach, Hashtag Approach, Contrast Approach, n-grams, bigrams, unigrams.*

1. INTRODUCTION

In recent years, social networking sites have become an important part of peoples life in societies .These sites are the source on entertainment, news and thereby these sites have create large amounts of data. This data need to be analyzed to derive important and valuable information about different topics.

Sentiment analysis (SA), otherwise called assessment mining is the procedure of grouping the emotions passed on by content, for instance as negative, positive and then again neutral. The information made accessible by online networking has contributed to a burst of research work in the domain of Sentiment Analysis. Data picked up from applying Sentiment Analysis to online networking information has numerous potential utilization, for example, to help advertisers assess the accomplishment of a promotion , to distinguish how diverse demographics have received product release , to anticipate user behavior , or to anticipate election results.

Sarcasm research needs both machine learning and natural language processing. For instance, consider this review about a pen: "This pen is justified regardless of the 1000 Rs it costs. It writes more regrettable than an ordinary pen and has none of the elements of a typical pen! It tears the page after every stroke. I'm so excited I purchased it." This is plainly a sarcastic comment of a costly pen. It talks about a costly pen, and despite the fact that the user says positive things in regards to the pen in the first and last sentence; he records just negative components in the middle.

This prompts some intriguing perceptions. These perceptions are the features or the elements that are important to identify sarcasm consequently. One perception is that perusing the first or last sentence in separation does not give any indication of sarcasm. They appear like conventional positive sentences about the item. Obviously, it might sound somewhat odd that a pen could cost $100, however it may be encrusted with diamonds or made out of gold, making the sentence sound sensible. Nonetheless, the center two sentences are plainly negative as it talks about what the pen needs and the shocking impact of utilizing the pen. This movement in sentiment between sentences is demonstrative of sarcasm. While sarcasm discovery is intrinsically perplexing and difficult, the style and nature of substance on Twitter further convolute the procedure. Contrasted with other, more routine sources, for example, news

articles and books, Twitter is more casual in nature with a developing vocabulary of slang words and condensing and has a point of confinement of 140 characters for each tweet which gives less word-level signs accordingly including more equivocalness.

1.1 Sentimental Analysis

Sentiment Analysis is a Natural Language Processing and Information Extraction task that expects to acquire author's sentiments communicated in positive or negative remarks, questions and demands, by investigating expansive quantities of documents. As a rule, sentiment analysis expects to decide the perception of a speaker or an author as for some point or the general tonality of a report. As of late, the exponential increment in the Internet utilization and trade of popular assessment is the main thrust behind Sentiment Analysis today.In conclusion, entity and aspect level analysis endeavors better grain analysis. It considers the sentiment of the content.

It expect that a conclusion comprises of a sentiment (positive or negative) and a target (i.e., the item which the content was composed for) as shown in Fig. 3.0. A case that Liu gives is: "In spite of the fact that the administration is not that extraordinary, despite everything I adore this eatery." There are two elements or parts of the sentence. The administration perspective is given a negative sentiment, while the eatery is given a positive sentiment.

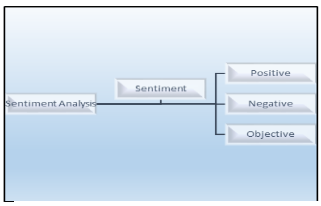


Figure 3.0 Sentiment Analysis Process

1.2 Application of Sentimental Analysis

Sentiment analysis is highly efficient and in demand. Since hundreds of thousands of files are being processed every day for sentiments and many other features included named topics, entities and themes. It helps in building better reports for understanding the customer's opinion and aspects of the company. All these processing can be done in seconds while compared to hours of work manual processed by people. Hence saves lots of time and complexity is reduced over time. Sentiment analysis is being adopted by many businesses' which also includes text analysis for better understanding of their customers and their requirements. There are three major advantages of sentiment analysis in the field of business:

1. Based on sentiment analysis reports if the given sentiment of a particulars is having a negative sentiment, the information can be evaluated and decisions can be taken to improve on the properties. These helps build a good relations with customers.
2. Evaluation of users likes and dislikes of two or more competitor businesses' which highlights the pros and cons of the business and can be taken as a means to improve on the business.
3. Finally using customer surveys for sentiment analysis and classification of data sets.

Other advantages:

1. The ability to identify sentiments in text to improve natural language processing systems.
2. It provides an insight of how various types of linguistic and extra linguistic information are used for text processing.
3. Some of its advantages are review summarization, brand monitoring and personalization of content recommendation.

1.3 Research Challenges in Sentimental Analysis

There are many open research challenges in sentimental analysis that need to be addressed and researched upon One of the challenges is automatic entity resolution which is not yet solved completely. Each product has many names that refer to it even within the same document and clearly across documents. Current techniques classify them as two different properties. Another related major hurdle is handling of anaphora resolution in an accurate way. This is a problem for aspect extraction too, that is, how to group aspects, the problem of resolving what a pronoun, or a noun phrase refers to. "We watched the movie and went to dinner; it was awful." What does "It" refer to? When a document discusses several entities, it is crucial to identify the text relevant to each entity. Current accuracy in identifying the relevant text is far from satisfactory. This is called named entity recognition which is another challenge in sentimental analysis. Another open challenge is to deal with Noisy texts (those with spelling/grammatical mistakes, missing/problematic punctuation and slang) are still a big challenge to most sentiment analysis system. Noisy texts are especially relevant to Twitter tweets, as tweets are commonly entered quickly resulting in typos, short hand notations, and slang. These noisy texts make it difficult for sentiment analysis systems to correctly identify the sentence structure. Context is an issue that requires the usage of semantics, and current systems overlook factual statements although they may contain sentiment.

1.4 N-gram Features

N-grams are capable of capturing context to some extent and are widely used in Natural Language processing tasks. Whether higher order n-grams are useful is a matter of debate. Pang *et al.* (2002) reported that unigrams outperform bigrams when classifying movie reviews by sentiment polarity, but Dave *et al.* (2003) found that in some settings, bigrams and trigrams perform better. Using N-gram

Patterns can be created by concatenating adjacent tokens into various unigram, bigrams etc... An example of how the n-grams are constructed is explained with the example sentence;
Hello my name is Raghu and I live to Eat.
Unigram = "hello", "my", "name", "is"….. "Eat"
Bigram = ["hello", "my"], ["my", "name"]…….
Models that assign probabilities to sequences of words are called language models. A new model was introduced where a simplest of models was able to assign language model probabilities to sentences and sequences of words, called the N-gram. An N-gram is a sequence of N-gram words: a 2-gram (or bigram) is a two-word sequence of words like "hello", "my", "name", "is"….. "Eat", and a 3-gram (or trigram) is a three-word sequence of words like "hello my name", or "hey what's up". The process in N-gram model performs the task of computing P (w | D), the probability of a word w given some dataset D. Suppose the Dataset is "many many happy returns of the" and model tries to find the probability that the next word as "day" and it's represented as:
P (day | many many happy returns of the)
Relative frequency counts is one of the way is to estimate this probability:
By taking a very large corpus, count the number of times the corpus contains the dataset sentence in them, and count the number of times this is followed by "day". This solves the problems in hand, whether day was followed by the dataset sentence or not.
P (day | many many happy returns of the) =
C (many many happy returns of the day)
C (many many happy returns of the)

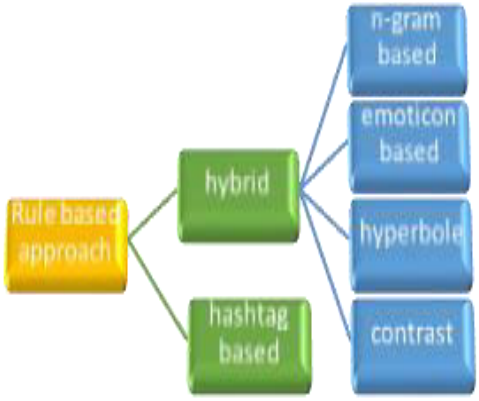
1.4 Naïve Bayes Algorithm

The Naive Bayes classifier is a popular machine learning classifier, because of its simplicity and is often used as baseline for text categorization. The classifier makes the "naive" assumption that independence occurs between all the features. Naive Bayes is often used in spam filtering and document classification. Naïve Bayes simply is a prediction algorithm, which predicts the probability of sentiment in text. It builds a probabilistic classifiers for machine learning based on Bayes theorem. It's a model that assign class labels to problem instances, represented as vectors of feature values. Bayes Algorithm can be efficiently built in supervised learning techniques. Even though it is used for positive negative sentiment analysis we used the features to develop a classifier to detect sarcasm:
Consider a document x, where C is the given class.
Consider email spam filtering there are two classes C = S (spam) and C = x (ham). We classify x as the class which has the highest posterior probability P (C | x), which can be re-expressed using Bayes' Theorem:

$$P(C_k \mid x) = P(C_k)\, P(x \mid C_k) / P(x)$$

2. METHODOLOGY

In the course of the most recent couple of decades, Social networking and micro blogging websites such as twitter has allowed people to encounter in utilizing online assets. A great deal of improvements have been observed in the field of sentimental analysis of twitter data. This project focuses mainly on sarcasm detection which is a major part of sentiment analysis of twitter data which is helpful to analyze the sarcasm in the tweets where views are miscellaneous and highly unstructured, or may be positive, negative, sarcastic, ironic or neutral in some cases. This research work borrows the ideas of utilizing different semi-supervised algorithms like Lexical Analysis with N-grams approach, Naïve Bayes, Knowledge extraction, contrast approach, emoticon based approach and hyperbole approach to propose a new rule based Hybrid and Hierarchical approach for sarcasm detection.



``Figure: 2.1.1 Rule based approach

2.1 *Rule1: Hash tag based Appraoch*

We have created a unique dataset dictionary for hashtags and represented them as unigram set of words. The most widely used unigram hashtags for sarcastic comments are #Not, #sarcasm, #sarcastic etc. A program logic is developed to identify these unique tags and make a comparison test to benefit our research for accurate results and classification of tweet data as sarcastic or not.
Input: Processed 'Tweets'
Logic:

From *tweetdict* import *tweets:*
For each *tweet* in *tweets:*
For *data* in *tweets*:
If *data == #Sarcasm-Data:*
Append *'Sarcastic'* to [Output-list]
Else:
Append 'Non-Sarcastic' to [Output-list]
For *tweet* in *tweets:*
For *listdata* in *Output-list:*
Newlst=Join (*tweet* with *list-data)*

Output: [*Newlist*] **of tweets**

2.2 *Contrast Approach*

Previous works have proved that parts of speech information is amongst the most informative and productive approach for this task. We have applied the POS tagger and respective valence value from Warner et al. (2013) Dataset which includes features based on the absolute count and ratio of each tweet, along with the "lexical density" of the tweet, which models the ratio of nouns, verbs, adjectives and adverbs to all words a baseline capacitor (General Dictionary) is developed and implemented for productive analysis and classified data.

Input: Processed 'Tweets'

Logic:

From *tweet-dict* import *tweet*
From *Pos-dictionary* import *All-pos*
For each *tweet* in *tweets:*
For *data* in *tweet:*
If *data* is found in *Positive-verb-box* and *negative-situation:*
Append *'Sarcastic'* to [*Output list]:*
Else:
Append *'Non-Sarcastic'* to [*Output list*]
For *tweet* in *tweets*:
For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*

Output*: [Newlist]* **of tweets**

2.3 *Rule3: N-grams Approach*

Patterns can be created by concatenating adjacent tokens into n-grams where n is the max value on which the all possible combinations can be made for a single pattern. In other words here n is equal to one hence called as unigram. An example of how the n-grams are constructed is explained with the example sentence: "jack likes his new backpack", by using n-grams it may be possible to capture how a word N-grams sequence:

Algorithm:

Input: Processed 'Tweets'

Logic:

From *tweet-dict* import *tweet*
From *Pos-dictionary* import *All-Pos*
From *Bi-gram* import *ngram-dict*
For each *tweet* in *tweets:*
For *data* in *tweet:*
If *Bigram-data* is found in *bigram-dict*:
Append *'Sarcastic'* to [*Output-list*]:
Else:
Append *'Non-Sarcastic'* to [*Output-list*]:
For *tweet* in *tweets*:
For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*

Output*: [Newlist]* **of tweets**

2.4 *Rule4: Hyperbole Approach*

Kreuz and Roberts 1995 theoretical work has stressed the importance of hyperbole for sarcasm, we have implemented indicator logic for whether the tweet contains a word in a list of intensifiers (so, too, very, really), stretched words (righttttt), capitalized words ([HELLO]), punctuations and so on. Hence giving us a more stable classifier hence covering most of the uncovered area for text classification and analysis.

Algorithm:

Input: Processed 'Tweets'

Logic:

From *tweets* import *tweet:*
From *unigram* import *ngram_dict:*
For each *tweet* in *tweets*:
for *data* in *tweets:*
If *unigram-data* is found in {*unigram-dict*}:
Increment count1 by 1
If *unigram-data* is found in {*punctuation-dict*}
Increment count2 by 1
If *count1, count2* [Greater than] *fixedvalue*:
Append *'Sarcastic'* to [*Output-list*]
Else:
Append *'Non-Sarcastic'* to [*Output-list*]:
For *tweet* in *tweets*:
For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*

Output*: [Newlist]* **of tweets**

2.5 *Rule 5: Emoticon Based*

We have developed a dictionary containing the Unicode values for all available emoticons and we use the positive-verb dictionary from "posdict" and assuming that a tweet having a positive sentiment followed by a negative sentiment emoticon is said to be sarcastic based on this theory we are mark and divide the tweets as sarcastic and non-sarcastic.The most commonly used emoticon to comment a sarcastic message in twitter or any social media is upside-down emoticon and it's been logically proved.

Algorithm:

Input: Processed 'Tweets'

Logic:

From *tweets* import *tweet*
From *posdata* import *all_dict*
From *emoji* import *emoji_dict*
For each *twee*t in *tweets:*
For *data* in *tweets:*
If *unigram-data* is found in all_dict:
Increment count1 by 1
If unigram-data is found in emoticon-dict
Increment count2 by 1
If count1 =positive and count2 = negative
Append *'Sarcastic'* to [*Output-list*]:

Else:
Append *'Non-Sarcastic'* to [*Output-list*]:
For *tweet* in *tweets*:
For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*
Output*: [Newlist]* **of tweets**

2.6 *Rule 6: Naïve Bayes Algorithm*

We performed a Naïve Bayes approach on a given test dataset using bayes theorem using python nltk. The values are initially trained under a large corpus of dataset of 20000 sarcastic and non-sarcastic tweets. And finally the tweets are trained for feature extraction and then classified as sarcastic and non-sarcastic tweets. For better results the data set is classified in three fold validation and classification using SVM (support vector machine) Classifier, binomial classifier and linear classifier.

Algorithm:
Input: Processed 'Tweets'
Logic:
Import *tweets*
From *tweets* import *tweet*
For each *twee*t in *tweets:*
For *data* in *tweets:*
If *NaïvebaeysClassifier(data)== 'sarcastic'*:
Append *'Sarcastic'* to [*Output-list*]:
Else:
Append *'NaivebayesClassifier* to [*Output-list*]:
For *tweet* in *tweets*:
For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*

2.7 *Rule 7: Hierarchical Approach*

The sixth rule in the rule based approach is hierarchical approach. It's a sequential logic of all the above approaches The tweets are marked as sarcastic if it's found true in anyone of the rule except for in hashtag rule (Since #tag rule tags all tweets as sarcastic) the implementation is simplified using hierarchical directory format.

Algorithm:
Input: Processed 'Tweets'
Logic:
For *every-rule twee*t in *Allrules:*
Import *tweets*
From *tweets* import *tweet*
For each *twee*t in *tweets:*
For *data* in *tweets:*
If *everyrule (data) == 'sarcastic'*:
Append *'Sarcastic'* to [*Output-list*]:
Else:
Append *'Non-Sarcastic'* to [*Output-list*]:
For *tweet* in *tweets*:
For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*
Output*: [Newlist]* **of tweets**

2.8 *Rule 8: Hybrid approach*

The final rule in the rule based approach is hybrid approach. It's a combinational logic of all the above approaches The tweets are marked as sarcastic if it's found true in anyone of the rule except for in hashtag rule (Since #tag rule tags all tweets as sarcastic) the implementation is simplified into hybrid directory format.

Logic:
For *every-rule twee*t in *Allrules:*
Import *tweets*
From *tweets* import *tweet*
For each *twee*t in *tweets:*
For *data* in *tweets:*
If *tworules (data) == 'sarcastic'*:
Append *'Sarcastic'* to [*Output-list*]:
Else:
Append *'Non-Sarcas'* to [*Output-list*]:
For *tweet* in *tweets*:
For *listdata* in *list:*
Newlist Join (*tweet* with *Output-List)*
Output*: [Newlist]* **of tweets**

3. RESULTS

The output of combinational approach, is a list of tweets marked either sarcastic or non-sarcastic for every rule in a rule engine and each results were saved by their rules name respectively as ".csv" files and charts were produced automatically by python and plotly. A comparative analysis has been shown for the Rule based approach. And proving that hybrid approach yields a better result compare to other approaches. The Rule based approach was able to detect sarcasm in text over 80% at precision rate which proved that it was giving 3% more accurate results compare to the existing algorithm in few test runs.

Datasets	**200 sarcastic tweets**	**100 non-sarcastic tweets**
Dataset 1	167	95
Dataset 2	175	89
Dataset 3	187	85

4. FUTURE WORK

In the future work we will be concentrating on un-structured data and will be working on un-supervised learning and classifying.

5. CONCLUSION

Sarcasm detection on twitter tweets is more complicated has it provides very less detailed results, and developing a dictionary for these kind of text documents takes more time and resources. Social media posts are hard to analyze on the phrase or sentence level because of theirs unique structure

and grammar. Since twitter allows user to enter 140 characters processing time also increases. The sarcasm detection was ignored for different languages (except English), repeated tweets and empty or a single letter/word tweets. Finally by using different types of features and their combinational logic we were able to detect sarcasm in twitter training data set. Preprocessing being the very important part of our project it was successfully completed. And the results were clean preprocessed and tagged tweets. In this phase we were able to remove anomalies or the noises such as hyperlinks, emailed, links and mention. We got 100% accuracy in detection and deletion of these noises. In POS tagging most of the important words were successfully detected whereas the undetected ones were due to misspelled words or the words which may be missing from dictionary or it may have been prepositions (in, of, as, a, the) which we are not considering overall we got 75% and over detection which gave us a better POS tagging dataset for feature extraction. The algorithm also detected emoticons and renamed them. Finally the result data set is moved to post processing Out of 300 tweets we considered 100 #sarcastic tweets, 100 non-sarcastic tweets and 100 sarcastic tweets with no # tags. The results were found to be extraordinary. The algorithm was able to classify accurately over this combined dataset. We found 68% sarcastic in one dataset and 71.35% in another dataset. The future work will be focused on backtracking of tweets (analyzed based on user's past replies and comments) and multilingual language support.

.

References

1. Whiting, A. and D. Williams, "Why people use social media: a uses and gratifications approach"
2. B. Liu, Sentiment Analysis and Opinion Mining. Morgan and Claypool Publishers, 2012.
3. B. Pang and L. Lee, "Opinion mining and sentiment analysis," Foundations and Trends in Information Retrieval, 2008.
4. "Oxford English dictionary online." http://www.oed.com, 2013
5. Hiroshi Shimodaira "Text Classification using Naive Bayes"
6. Go, A., Bhyani, R., and Huang, L. "Twitter sentiment classification using distant supervision." CS224N Project Report, Stanford, 2009.
7. Ms. Harvinder Jeet Kaur, Mr. Rajiv Kumar, "Sentiment Analysis from Social Media in Crisis Situations" ICCCA, 2015

8 Barbosa, L., and Feng, J. "Robust sentiment detection on twitter from biased and noisy data." In Proceedings of COLING, pp. 36–44, 2010.

8. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R. "Sentiment analysis of twitter data." In Proceedings of the ACL 2011 Workshop on Languages in Social Media, pp. 30–38, 2011.
9. Abdellatif Rahmoun and Zakaria Elberrichi, "Experimenting N-Grams in text Categorization", October 2007
10. Subhabrata Mukherjee and Akshat Malu Balamurali A.R and Pushpak Bhattacharyya. "Sarcasm detection in twitter data"
11. Dadvar, M. and Hauff, C. and de Jong, F.M.G. (2011) "Scope of negation detection in sentiment analysis".
12. Ashwin Rajadesingan and Reza Zafarani Arizona and Huan Liu Arizona "Sarcasm Detection on Twitter: A Behavioral Modeling Approach"
13. Dmitry Davidov and Oren Tsur and Ari Rappoport "Enhanced sentiment learning using Twitter hashtags and smileys"
14. N. Kourtellis, J. Finnis, P. Anderson, J. Blackburn, C. Borcea, and A. Iamnitchi, "Prometheus: user-controlled p2p social data management for socially-aware applications,"
15. J. Ratkiewicz, M. Conover, M. Meiss, B. Gonalves, S. Patil, A. Flammini, and F. Menczer, "Truthy: mapping the spread of AstroTurf in microblog streams,"
16. Haruna Isah, Paul Trundle, Daniel Neagu "Social Media Analysis for Product Safety using Text Mining and Sentiment Analysis"
17. N. Jakob and I. Gurevych, "Extracting opinion targets in a single- and cross-domain setting with conditional random fields," in Proceedings of the 2010 Conference on Empirical Methods in Natural Language Processing. Stroudsburg, PA, USA: Association for Computational Linguistics, 2010, pp. 1035–1045.