TEXTUAL SENTIMENT DISSECTION OF LIVE TWITTER REVIEWS USING MACHINE LEARNING

1. Mr.K.Veera Kishore, Asso professor,CSE,Sri Indu Institute of Engineering&Technology(SIIET), Sheriguda,Ibrahimpatnam,Hydarabad
2.K.Gokul Krishna ,Student,CSE,SIIET,Sheriguda,Ibrahimpatnam,Hydarabad
3.K.Peter Prashanth,Student,CSE,SIIET,Sheriguda,Ibrahimpatnam,Hydarabad
4.M.Sujeeth Reddy,Student,CSE,SIIET,Sheriguda,Ibrahimpatnam,Hydarabad
5. N. Bharathsimha reddy,Student,CSE,SIIET,Sheriguda,Ibrahimpatnam,Hydarabad
6. R. Kranthikiran,Student,CSE,SIIET,Sheriguda,Ibrahimpatnam,Hydarabad

Abstract

Textual dissection can be a very useful aspect for the extraction of useful information from text documents. The ideology of textual dissection is the way people think about a particular text. It is the process where given reviews are classified as positive or negative.A huge amount of data (reviews) is present on the web which can be analyzed to make it useful. It can prove to be useful specifically for marketing, business, polity as it allow us to do easy analysis of the subject under consideration. In today's era of internet, lots and lots of people can connect with each other. Internet has made it possible for us to connect and find out the opinions dissection.Internet has provided a lot of platform through which opinions from different people can be taken through Forums, Blogs, and Social networking sites. This paper proposes the use of Tweepy and TextBlob as a python library to access and classify Tweets using Naïve Bayes, a Machine Learning technique. Our Technique is meant to ease out the process of analysis, summarization and classification.

1. Introduction

Sentiment analysis algorithms actually consists of Natural Language Processing (NLP) like Part of Speech Tagging (POS) by the use of resources like emotion-based lexicons or some dictionaries [18, 19]. This can be done in any text format and on different granularity levels which may range from a word to a sentence or a document. Most of the research on Textual dissection has been carried out on texts such as product reviews [10, 17]. This paper is based on twitter reviews on Live Dataset and can be done on any topics available in twitter. There is a window size which can be adjusted to take that number of tweets and analyze the polarity of the review. The training data is trained using a Naïve based classifier present on TextBlob, Library of Python. This paper proposes a method or an algorithm that uses two libraries of python that is Tweepy to access the Live Tweets and TextBlob to classify, label and calculate the polarity of the tweets.

2. Related Work

There are many research works published in the area of Sentiment Analysis [3,5,7,8,9,12,13,14,15,16]. The part of speech tagging is used in the Penn Treebank Project [1]. Out of 36 tags, some of the tags used in the work mentioned in the table below.

Number	Tag	Description
1.	VB	Verb
2.	JJ	Adjective
3.	NN	Noun, Singular
4.	NNS	Noun, Plural
5.	RB	Adverb
6.	IN	Preposition or subordinating conjunction

Table I: Common POS Tagging

Pankaj Gupta et al. [16], authors has used Naïve Bayes and SVM and created a collection of useful text from different Bag-Of-Words (BOW) and presented the summary. Rui Xia et al. [12], they have presented textual dissection in real sense as: the 9 datasets, 2 antonym dictionaries and 3 classification algorithms; has been inspected and classified using binary classification into two classes (positive and negative). Moreover, the binary classes further extended into ternary classes (positive-negative-neutral) that includes third class as neutral reviews. The benefit is that, it is very operative for parting classification and extending the DSA algorithm.

Md. Ansarul Haque and Tamjid Rahman [14] has proposed analysis of sentiments using Fuzzy Logic where they use emotion analysis with the help of fuzzy logic
that will help the creators or consumers or any concerned person for taking the actual decision according to their
product or service interest. The added value is that it is supportive for anyone in any way to meet up their benefits or what they justify.
Minara P Antony et al. [15], proposed about POS tagging method which is used to recognize the stop words and to distinct the sentiment terms. Unigram method is used to calculate the overall rating. The Stanford collection for organizing the data into negative, positive and neutral words and twofold prediction to properly identify the polarity of the data is being proposed. Rushlene Kaur Bakshi et al. [13], discusses about emotion exploration which is a linguistic independent technology and also applied in the study of sociology, law, psychology etc. Agrawal, Rakesh, and Ramakrishnan Srikant [20] discusses about some new hybrid fast algorithms for association rule mining.
Authenticating and accessing of the Live Tweets is done by a Python Library, Tweepy. As Twitter requires all requests to use OAuth for Authentication and this Tweepy is an API that helps to authenticate a user and allows the user to access Live Twitter Data by just creating a Twitter application and retrieving your API access keys and tokens. These tokens will allow you to authenticate your Python client application with Twitter. TextBlob, a Python Library is used for noun phase extraction, POS Tagging, Classification, Tokenization, Word Inflection, Word and Phrase frequency,
etc. NLTK is slightly different from TextBlob as TextBlob has a MIT License, is built on the top of NLTK and is very easily accessible. TextBlob is used for fast prototyping. On Comparing the Code Quality as calculated and provided by Lumnify, TextBlob is L3 and that of NLTK is L2. They vary from L1 to L5 with “L5” being the highest. TextBlob is used to provide an API for diving into common NL Processing tasks. Finally, the proposed code uses the above libraries to display the polarities of the Live Twitter Reviews as negative or positive and then display some Positive and

Negative Reviews. This is not 100 % accurate but is very useful in consideration of millions of tweets in every minute.

3. Textual Dissection through Naïve Bayes

An Overview of the steps and techniques used in Textual dissection process is shown in Figure-1. This work proposes an algorithm with the help of which the user would be able to access the real-time tweets and analyze their sentiments at the same time. For this process, the user would require various tools and using those tools in a proper manner.

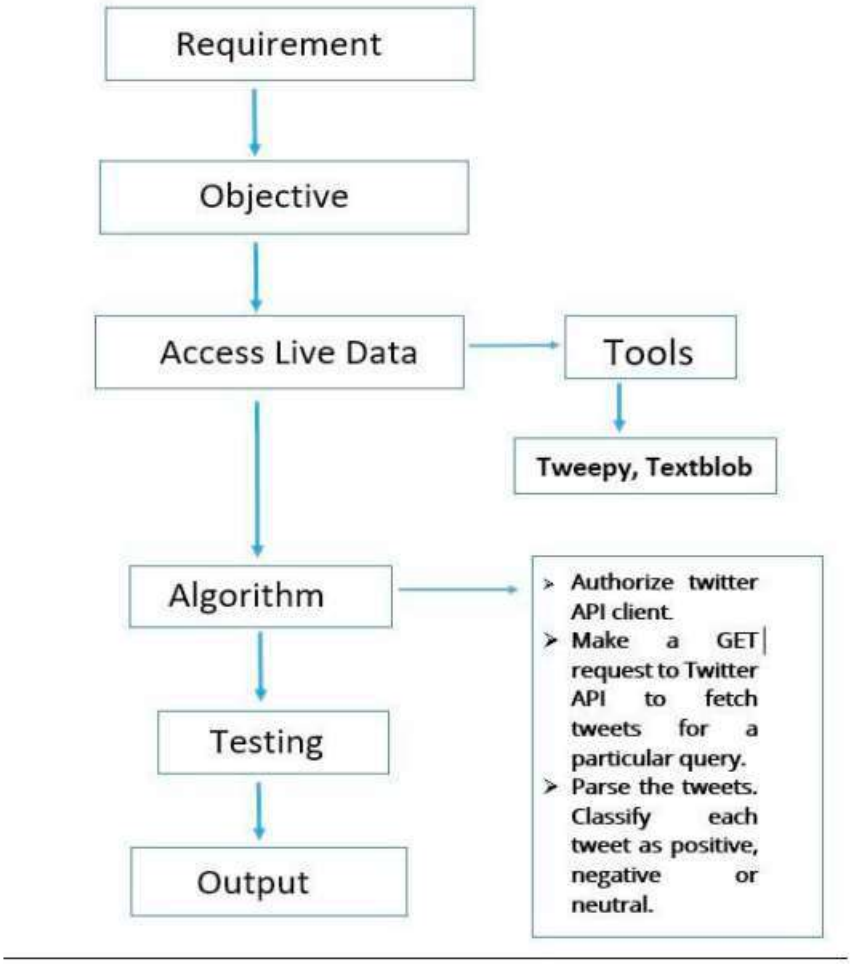

Fig1:- Flow Chart of Sentiment Analysis

Natural Language Toolkit (NLTK), Tweepy and TextBlob are the most important tool the author has worked on. The flow of work is explained in the following modules.

A. Requirements :

As the live Twitter Data is going to be accessed, the user need to import the Tweepy (Python Client for official Twitter API) and TextBlob.

For this, first the user need to install "pip", a package used to install software packages written in python. After that the user needs to install Tweepy and TextBlob, using the following commands on the Terminal window:-

pip install tweepy

pip install textblob

For TextBlob, the user needs to install the NLTK Corpora Tool (contains large amount of data) using:-

python -m textblob.download_corpora

B. Objective

Develop an algorithm that would take the query of the person's name for whom the user want to calculate the
percentage of positive tweets and the percentage of negative tweets. And also the user want to show the five positive
and five negative tweets.

C. Access Live Data and Processing

I. Tweepy: - The users have to register their app on the Twitter website and then get the tweets. This is done
at the Application Management of the Developers Section of Twitter. This is a very important step for the
OAuth Authentication of Tweepy tool.
After creation of the application, the user need to generate private keys. These are required for OAuthHandlertakes
this as the parameters. This is shown in dev.twitter.com. After successful generation of all the keys, we will copy the
keys which will be further used in our Algorithm.

Fig 2 - Dataset in TextBlob

TextBlob: The data in the text format is usually processed using the Textblob Library. TextBlob objects are treated as if they were like Python strings that learned to do the natural language processing.

D. Algorithm Description

The algorithm proposed in our work has mainly 3 major steps.

- Authenticate twitter account.
- GET Request is made to the twitter for getting the tweets.
- Select the tweets. Segregate each tweet positive, negative.

The user has to make a twitter api-client. The given class contain the function which allow us to access the tweets.

init function is used for the authentication purpose. The clean tweets is used for cleaning the dataset acquired by the twitter api. For this the user has to import the Regular Expression library which is available in Python.

In **get_tweets** function, the user uses the following piece of code to call the API to get the tweets: -

fetched_tweets=self.api.search(q=query, count=count)

In **get_tweet_sentiment** function the user uses the TextBlob module:-

analysis = TextBlob(self.clean_tweet(tweet))

A classifier function divides the tweets as positive and negative polarity in the range of -1.0 to 1.0

Creation of **Sentiment Classifier**:-

- TextBlob contains a Movie dataset where positive and negative reviews has already been labelled.
- From each positive and negative review, positive and negative features are extracted respectively.
- This data with positive and negative features is now trained on Naive Bayes Classifier.

Then, the user can see a **TextBlob** class, where **sentiment.polarity** method can be used to get the polarity of tweets

between -1 to 1.

Then, we classify polarity as: if analysis.sentiment.polarity is greater than 0 then return 'positive' else return

'negative'.

Finally, parse tweets are returned and then the user will calculate the percentage of positive and negative tweets.

E. Comparative Study

This code is using naïve Bayes for classification while there are several other methods present in these scenario to

classify the Dataset.

PART-1

1. Jin Huang Jingjing Lu Charles X. Ling [6] concludes that both Naïve Bayes and Support Vector Machine

(SVM) experimentally have a very similar accuracy.

2. Kathleen Goeschel's [11] paper has shown that high accuracy may be maintained while reducing false

positives using the proposed model composed of SVMs, decision trees, and Naïve Bayes.

PART-2

The authors used Rapid Miner Tool to compare Decision Tree and Naïve Bayes over a static Dataset "Titanic"

available in Rapid Miner. It was found that Naïve Bayes was 92.58 % accurate and Decision Tree was only 79.04 %

accurate. Table-II and Table-III displays the comparative results. This shows that Naive Bayes was the better option

for classification here.

5. Conclusion & Future Scope

In today's world, there is, violent contents spreading across Twitter and it is usually targeted to a particular community, government, religion, celebrities or politicians. This creates a situation of violent extremism and it needs immediate efforts to control it. The aim of this project is to counter this violent extremism and the spread of extremist contents on Twitter. Lexical resources have been used to enhance the sentiment related nature of the text. It can be concluded that subjective extracts gives a better result when applied on the textual dissection algorithm. In future cognitive angle to Sentiment Analysis can be explored to further extent. This project is able to provide a real-time

sentiment analysis of any community, government, religion, celebrities or politician across the globe anytime.

6. References

[1] Beatrice Santorini, "Part-of-Speech Tagging Guidelines for the Penn Treebank Project," University of Pennsylvania, Department of
Computer and Information Science Technical Report No. MS-CIS-90-47. (2009)
[2] Tiwari, Vivek, et al. "Association rule mining: A graph based approach for mining frequent itemsets." Networking and Information
Technology (ICNIT), 2010 International Conference on. IEEE, 2010.
[3] Smeureanu, Ion, and Cristian Bucur. "Applying supervised opinion mining techniques on online user reviews." Informatica
economica 16.2 (2012): 81.
[4] Tiwari, Vivek, and R. S. Thakur. "Pattern Warehouse: Context Based Modeling and Quality Issues." Proceedings of the National
Academy of Sciences, India Section A: Physical Sciences 86.3 (2016): 417-431.
[5] Rao, Shivani, and Misha Kakkar. "A rating approach based on sentiment analysis." Cloud Computing, Data Science & Engineering
Confluence, 2017 7th International Conference on. IEEE, 2017.
[6] Bahrainian, Seyed-Ali, and Andreas Dengel. "Sentiment analysis using sentiment features." Proceedings of the 2013 IEEE/WIC/ACM
International Joint Conferences on Web Intelligence (WI) and Intelligent Agent Technologies (IAT)-Volume 03. IEEE Computer Society,
2013.
[7] Ahmed, Khaled, Neamat El Tazi, and Ahmad Hany Hossny. "Sentiment Analysis over Social Networks: An Overview." Systems, Man,
and Cybernetics (SMC), 2015 IEEE International Conference on. IEEE, 2015.
[8] Srivastava, Rohan, et al. "Capital market forecasting by using sentimental analysis." Next Generation Computing Technologies
(NGCT), 2016 2nd International Conference on. IEEE, 2016.
[9] Povoda, Lukas, Radim Burget, and Malay Kishore Dutta. "Sentiment analysis based on Support Vector Machine and Big
Data." Telecommunications and Signal Processing (TSP), 2016 39th International Conference on. IEEE, 2016.
[10] Sarlan, Aliza, Chayanit Nadam, and Shuib Basri. "Twitter sentiment analysis." Information Technology and M*ultimedia (ICIMU), 2014*
International Conference on. IEEE, 2014