International Journal of Interdisciplinary Innovative Research &Development (IJIIRD)
ISSN: 2456-236X
Vol. 08 Special Issue 01 | 2023

A framework to detect the snetoment of twitter using machine learning tchniques

Mr. Rushikesh R Sonawane[1], Mr. Rushikesh K Kahate[2], Mr. Prajwal D Dhandare[3],
Mr. Rushikesh K. Rajure[4]
[1,2,3,4] *Student,CSE Department, Padm. Dr. V. B. Kolte, College of Engineering, Malkapur, Maharashtra, INDIA*

ABSTRACT

The micro blogging site started in 2006 has become great popularity such as Twitter, Weibo, Tumblr, facebook etc. This is resulted in explosion of the amount of short text messages. In this majority of tweets many of the tweets consists of some meaningful information or news. Tweets, in their raw form, while being informative, can also be immense. The searching for a hot topic may yield millions of tweets, spanning weeks. For this there is one solution namely filtering. Even if filtering is allowed, plowing for important contents, through such huge amount of tweets is also very difficult and hard to possible task. This is happen because of enormous amount of noise and redundancy. Another possible solution for information overload problem is Sentiments.

Sentiments is the process of reducing a text document with a computer program for creating a summary that contains the only important points of the original document. The problem of information overload is increases, and because of the quantity of data is increasing, there is a necessity automatic Sentiments. This technology makes use of a coherent summary such as length, style of writing and syntax. Automatic data Sentiments is a very important area within machine learning and data mining. These Sentiments technologies are widely used today, in a large number of micro blogging industries. Here are some examples of search engines in which Sentiments techniques are used such as Twitter, Facebook, and Google etc. Other category includes document Sentiments, image collection Sentiments and video Sentiments.

***Keyword:** - Microblogging, information, news, data, Sentiments*

1. INTRODUCTION

The main idea behind Sentiments is to find a representative and common subset of the data, which represent unique information of the entire set. Document Sentiments, tries to automatically create a representative summary or abstract of the entire document, by finding the most informative sentences. Similarly, in image Sentiments the system finds the most representative and important (or salient) images. For tweet Sentiments mostly document Sentiments technique is used. There are two types of automatic Sentiments approaches: extraction and abstraction. In extraction based Sentiments task, the automatic system extracts objects from the entire collection, without modifying the objects itself. Examples of this include key phrase extraction, where the goal is to select individual words or phrases to "tag" a document, and document Sentiments, where the goal is to select whole sentences (without modifying them) to create a short paragraph summary. Similarly, in image collection Sentiments the system extracts images from the collection without modifying the images themselves. On the other hand, abstraction based Sentiments task, involves paraphrasing sections of the source document. In general, abstraction can condense a text more strongly than extraction, but the programs which can do this are harder to develop as they require the use of natural language generation technology, which itself is a growing field.

Traditional document Sentiments techniques are not effective for big size tweets as well as not suitably applicable for tweets which are arrived fast and continuously. To overcome this problem tweet Sentiments is requires which should have new functionality significantly different from traditional Sentiments. Tweet Sentiments has to take into consideration the temporal feature of the arriving tweets.

Consider example of Apple tweets. A tweet Sentiments system will monitor Apple related tweets which are produced a real-time timeline of the tweet stream. Given a timeline range, the document system may generate a series of current time summaries to highlight points where the topic or subtopics evolved in the stream. Such a system will effectively enable the user to learn major news or discussion related to Apple without having to read through the entire tweet stream.

Stemming is the term used in linguistic morphology and information retrieval to describe the process for reducing inflected (or sometimes derived) words to their word stem, base or root form generally a written word form. The stem need not be identical to the morphological root of the word; it is usually sufficient that related words map to the same stem, even if this stem is not in itself a valid root. Algorithms for stemming have been studied in computer science since the 1960s. Many search engines treat words with the same stem as synonyms as a kind of query expansion, a process called conation.

In computing, stop words are words which are filtered out before or after processing of natural language data (text). Though stop words usually refer to the most common words in a language, there is no single universal list of stop words used by all natural language processing tools, and indeed not all tools even use such a list. Some tools specifically avoid removing these stop words to support phrase search.

Any group of words can be chosen as the stop words for a given purpose. For some search engines, these are some of the most common, short function words, such as the, is, at, which, and on. In this case, stop words can cause problems when searching for phrases that include them, particularly in names such as "The Who", "The The", or "Take That". Other search engines remove some of the most common words including lexical words, such as \want" from a query in order to improve performance.

2. OBJECTIVES

- To make a broad overview of promising Sentiments approaches on a Twitter topic. Also an automatic evaluation of Sentiments techniques by surveying recent evaluation methodologies is used to provide the proper result.
- To design a continuous tweet stream Sentiments framework.
- To make an efficient framework with more effective and productive output.
- To compress tweets and maintained in online fashion.

3. LITERATURE REVIEW

A Graph Based Clustering Technique for Tweet Sentiments by Soumi Dutta, Sujata Ghatak, Moumita Roy, Saptarshi Ghosh and Asit Kumar Das.In this paper a graph-based approach for summarizing tweets, where a graph is first constructed considering the similarity among tweets, and community detection techniques are then used on the graph to cluster similar tweets. Finally, a representative tweet is chosen from each cluster to be included into the summary. The similarity among tweets is measured using various features including features based on WordNet synsets which help to capture the semantic similarity among tweets. The proposed approach achieves better performance than Sumbasic, an existing Sentiments technique. Calculating the Sentiments tweets will provide better result in the proposed system and helps to evaluate it accuracy. [1]

Graph Sentiments for Hash tag Recommendation by Mohammed Al-Dhelaan, Hadel Alhawasi, In this paper hash tag recommendation is the problem of finding interesting hash tags for a user, which are not easily found via Twitter search. Searching a hash tag simply shows a list of tweets, each contains the query hash tag string. To find even more relevant hash tags, proposing to use a graph-based approach to find similar hash tags by using the social network graph around hash tags. We start by using a heterogeneous social graph that contains users, tweets, and hash tags, then summarizing the graph to a hash tag graph that shows the similarity between different hash tags. Finally, ranking the vertices in respect to a query hash tag using a random walk with restart and a content similarity measure. Hash tag being one of the important features to recognize the background of the tweets and helps to generate the positive and negative impact on the same. [2]

An Open Access Dataset of Tweets related to Exoskeletons and 100 Research Questions by Nirmalya Thakur and Chia Y. Han1. The Internet of Everything era of today's living, characterized by people spending more time on the Internet than ever before, holds the potential for developing such a dataset by the mining of relevant web behavior data from social media communications, which have increased exponentially in the last few years. Twitter, one such social media platform, is highly popular amongst all age groups, who communicate on diverse topics including but not limited to news, current events, politics, emerging technologies, family, relationships, and career opportunities, via tweets, while sharing their views, opinions, perspectives, and feedback towards the same. To address this research challenge by utilizing the potential of the Internet of Everything style of living, this paper makes multiple scientific contributions to this field. Considering the method of tweeting makes it possible to execute the parameters in proper manner and proceed to work in proper channel.[3]

Multi-criterion real time tweet Sentiments based upon adaptive threshold by Abdel hamid. Real time Sentiments in microblog aims at providing new relevant and non redundant information about an event as soon as it occurs. In this paper, introducing a new tweet Sentiments approach where the decision of selecting an incoming tweet is made immediately when a tweet is available. Unlike existing approaches where thresholds are predefined, the proposed method estimates thresholds for decision making in real time as soon as the new tweet arrives. Tweet selection is based upon three criterion namely informativeness, novelty and relevance with regards of the user's interest which are combined as conjunctive condition. Only tweets having an informativeness and novelty scores above a parametric-free threshold are added to the summary. The evaluation of our approach was carried out on the TREC MB RTF 2015 data set and it was compared with well-known baselines. Different sets of tweets make the Sentiments pattern easy to understand in terms of various categories of tweets posting.[4]

Tweet Segmentation and its Application to Named Entity Recognition by Chenliang Li, Aixin Sun, Jianshu Weng, and Qi. Twitter has attracted millions of users to share and disseminate most up-to-date information, resulting in large volumes of data produced every day. However, many applications in Information Retrieval (IR) and Natural Language Processing (NLP) suffer severely from the noisy and short nature of tweets. In this paper, we propose a novel framework for tweet segmentation in a batch mode, called HybridSeg. By splitting tweets into meaningful segments, the semantic or context information is well preserved and easily extracted by the downstream applications. HybridSeg finds the optimal segmentation of a tweet by maximizing the sum of the stickiness scores of its candidate segments. The stickiness score considers the probability of a segment being a phrase in English (i.e., global context) and the probability of a segment being a phrase within the batch of tweets (i.e., local context). Splitting of tweets is one of the essential parameters to resolve the unwanted characterization of various tweets and helps to interact with proper words present in tweets.[5]

4. SYSTEM ARCHITECTURE

From the literature study about traditional document Sentiments methods, it is concluded that these approaches are not much effective in the case when tweets arrive fast and continuous in a large volume. Due to the above-defined challenges, the tweet Sentiments approach must be capable to provide some advanced functionalities. Twitter is open to all for the uploading of the content so someone may upload any irrelated or bad content in such cases Sentiments process may help and play a significant role.

For better results proposed multi hoped Sentiments approach performs tweets Sentiments as well as segmentation. A tweets segmentation makes it possible to conserve the semantics of tweets. Tweets are generally short in size therefore tweets (short-text messages) are being generated and shared at an unusual rate. Tweets in raw form also may be informative and overwhelming. Here proposing a method for end-users and data analysts, it is very difficult to go through the huge number of tweets and also specifically when they are with an enormous amount of noise. The proposed architecture is capable to overcome the above-defined problems and also capable to generate online Sentiments with a multitopic version. Therefore, tweet Sentiments is an important concept that has to take into consideration for the temporal feature of the arriving tweets.

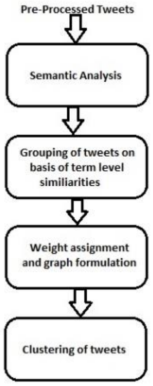
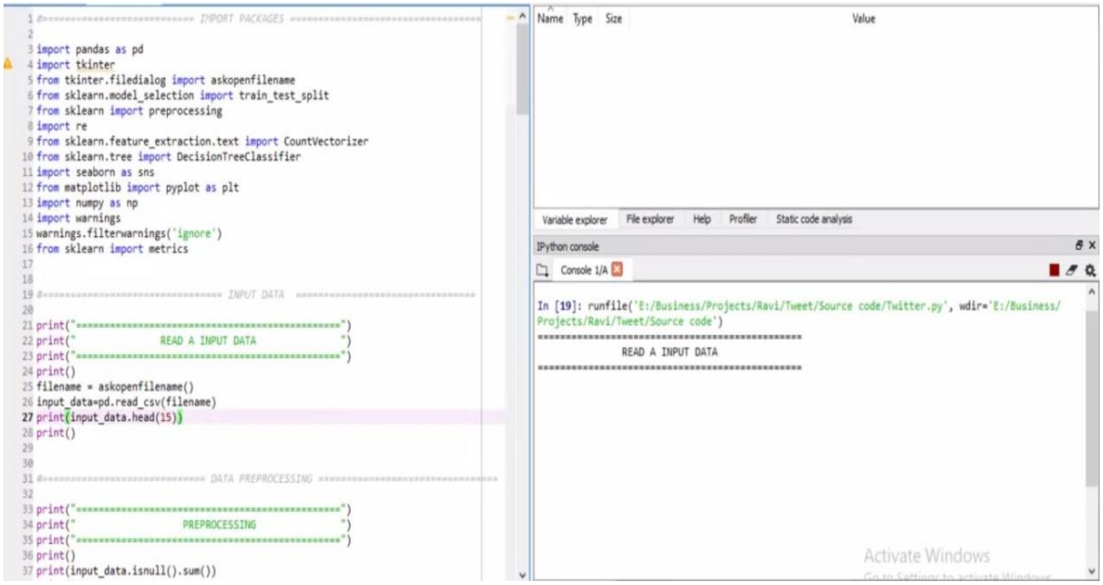

Pre-Processed Tweets
Semantic Analysis
Grouping of tweets on basis of term level similiarities
Weight assignment and graph formulation
Clustering of tweets

Fig. 1 Flow Diagram of the Incoming Tweets and their Processing Process

5. RESULT AND IMPLEMENTATION

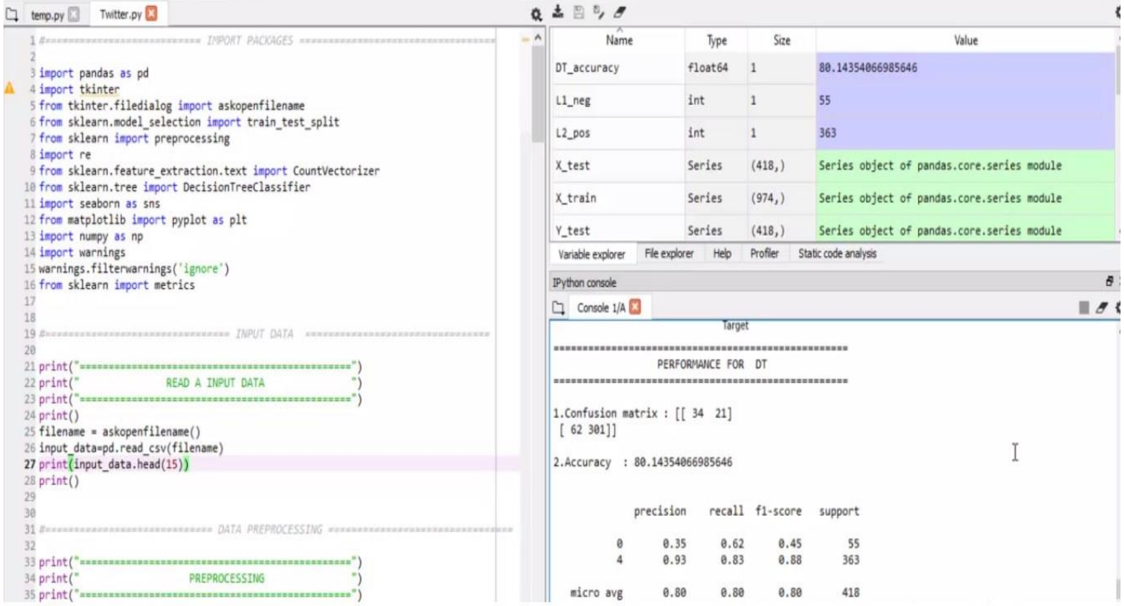


```
#============================ IMPORT PACKAGES ================================

import pandas as pd
import tkinter
from tkinter.filedialog import askopenfilename
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
import re
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.tree import DecisionTreeClassifier
import seaborn as sns
from matplotlib import pyplot as plt
import numpy as np
import warnings
warnings.filterwarnings('ignore')
from sklearn import metrics


#================================ INPUT DATA  ================================

print("==============================================")
print("              READ A INPUT DATA               ")
print("==============================================")
print()
filename = askopenfilename()
input_data=pd.read_csv(filename)
print(input_data.head(15))
print()


#============================ DATA PREPROCESSING ================================

print("==============================================")
print("                 PREPROCESSING                ")
print("==============================================")
print()
print(input_data.isnull().sum())
```

```
In [19]: runfile('E:/Business/Projects/Ravi/Tweet/Source code/Twitter.py', wdir='E:/Business/
Projects/Ravi/Tweet/Source code')
==============================================
              READ A INPUT DATA
==============================================
```

Fig. 2 Editor Window for Execution

Visual Studio Code is a code editor. Like many other code editors, VS Code adopts a common user interface and layout of an explorer on the left, showing all of the files and folders you have access to, and an editor on the right, showing the content of the files you have opened.



```
#============================ IMPORT PACKAGES ================================

import pandas as pd
import tkinter
from tkinter.filedialog import askopenfilename
from sklearn.model_selection import train_test_split
from sklearn import preprocessing
import re
from sklearn.feature_extraction.text import CountVectorizer
from sklearn.tree import DecisionTreeClassifier
import seaborn as sns
from matplotlib import pyplot as plt
import numpy as np
import warnings
warnings.filterwarnings('ignore')
from sklearn import metrics


#================================ INPUT DATA  ================================

print("==============================================")
print("              READ A INPUT DATA               ")
print("==============================================")
print()
filename = askopenfilename()
input_data=pd.read_csv(filename)
print(input_data.head(15))
print()


#============================ DATA PREPROCESSING ================================

print("==============================================")
print("                 PREPROCESSING                ")
print("==============================================")
```

Name	Type	Size	Value
DT_accuracy	float64	1	80.14354066985646
L1_neg	int	1	55
L2_pos	int	1	363
X_test	Series	(418,)	Series object of pandas.core.series module
X_train	Series	(974,)	Series object of pandas.core.series module
Y_test	Series	(418,)	Series object of pandas.core.series module

```
                    Target
==================================================
              PERFORMANCE FOR  DT
==================================================

1.Confusion matrix : [[ 34  21]
 [ 62 301]]

2.Accuracy  : 80.14354066985646


             precision    recall  f1-score   support

          0       0.35      0.62      0.45        55
          4       0.93      0.83      0.88       363

  micro avg       0.80      0.80      0.80       418
```

Fig. 3 Accuracy Measurement

Measurement accuracy is defined as the closeness of agreement between a measured quantity value and a true quantity value of a measured (i.e., the quantity intended to be measured) (ISO-JCGM 200, 2008), and is often limited by calibration errors.

6. Conclusion

Sentiment Analysis is a challenging task where traditional text summarization methods do not work well. In the last decade, various research works introduced different approaches for automatic Twitter topic summarization. The main aim of this work is to make a broad overview of promising summarization approaches on a Twitter topic. Also, an automatic evaluation of summarization techniques by surveying recent evaluation methodologies is used to provide the proper result. It also helps to continuous tweet stream summarization, online and historical summarization. The framework that will be designed must be efficient and effective and effect on compressed tweets and be maintained in an online fashion.

7. Future Work

In this way the methods in which comparative analysis was done and promising results were achieved. Computational time was also reduced which is helpful when deploying a model. It was also found out that the dataset should be normalized; otherwise, the training model gets over fitted sometimes and the accuracy achieved is not sufficient when a model is evaluated for real-world data problems which can vary drastically to the dataset on which the model was trained. It was also found out that the statistical analysis is also important when a dataset is analysed and it should have a Gaussian distribution, and then the outlier's detection is also important and a technique known as Isolation Forest is used for handling this. The difficulty which came here is that the sample size of the dataset is not large. If a large dataset is present, the results can increase very much in deep learning and ML as well. Algorithm applied by us in ANN architecture increased the accuracy which compared with the different researchers. dataset size can be increased and then deep learning with various other optimizations can be used and more promising results can be achieved. Machine learning and various other optimization techniques can also be used so that the evaluation results can again be increased. More different ways of normalizing the data can be used and the results can be compared.

8. REFERENCES

[1] Soumi Dutta, Sujata Ghatak, Moumita Roy , Saptarshi Ghosh and Asit Kumar Das "A Graph Based Clustering Technique for Tweet Summarization" 978-1-4673-7231©2015 IEEE.

[2] Mohammed Al-Dhelaan Department of Computer Science King Saud University,Riyadh,Saudi Arbia "Graph Summarization for Hash tag Recommendation" 3rd International Conference on Future Internet of Things and Cloud 2015.

[3] Nirmalya Thakur and Chia Y. Han "An Open Access Dataset of Tweets related to Exoskeletons and 100 Research Questions"

[4] Abdel hamid "Multi-criterion real time tweet summarization based upon adaptive threshold", IEEE/WIC/ACM International Conference on Web Intelligence (WI 2016), 13 October 2016 - 16 October 2016 (Omaha, Nebraska, United States).

[5] Chenliang Li, Ajxin Sun, Jianshu Weng and Qi He "Tweet Segmentation and its Application to Named Entity Recognition" IEEE Transaction On Knowledge and Data Engineering, Submission 2013.

[6] Asif Hossain Khan, Danushka Bollegala,Kaoru Sezaki entitled "Multi-Tweet Summarization of Real-Time Events" in DOI: 10.1109/SocialCom.2013.26

[7] C. Shen, F. Liu, F. Weng, and T. Li, "A participant-based approach for event summarization using twitter streams", in Proc. Human Lang. Technol. Annu. Conf. North Amer. Chapter Assoc. Compute. Linguistics, 2019, pp. 11521162.

[8] L. Vanderwende, H. Suzuki, C. Brockett, and A. Nenkova, "Beyond SumBasic: Task focused summarization with sentence simplification and lexical expansion", Information Processing Management, vol. 43, no. 6, pp. 16061618, 2009.

[9] David Inouye, Jugal K. Kalita, "Comparing Twitter Summarization Algorithms for Multiple Post Summaries", IEEE Trans. Knowledge Data Eng., 23(8):12001214, 2017.

[10] C. C. Aggarwal, J. Han, J. Wang, and P. S. Yu, "A framework for clustering evolving data streams" in Proc. 29th Int. Conf. Very Large Data Bases, 2018, pp. 8192.