NATURAL LANGUAGE PROCESSING ANALYSIS WEBSITE USING PYTHON/DJANGO

A PROJECT REPORT

Submitted by

NANTHAKUMAR J J. [REGISTER NO:211417104159]

PRAVEEN K. [REGISTER NO: 211417104197]

NIRMAL KUMAR N. [REGISTER NO: 211417104166]

in partial fulfillment for the award of the degree

of

BACHELOR OF ENGINEERING

IN

COMPUTER SCIENCE AND ENGINEERING

PANIMALAR ENGINEERING COLLEGE, CHENNAI-600123.

ANNA UNIVERSITY: CHENNAI 600 025

APRIL 2021

BONAFIDE CERTIFICATE

Certified that this project report **"NATURAL LANGUAGE PROCESSING ANALYSIS WEBSITE USING PYTHON/DJANGO"** is the bonafide work of " NANTHAKUMAR J J. (211417104159), PRAVEEN K. (211417104197), NIRMAL KUMAR N. (211417104166) " who carried out the project under my supervision.

SIGNATURE

Dr. S. MURUGAVALLI,M.E.,Ph.D.,
HEAD OF THE DEPARTMENT
PROFESSOR
DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

SIGNATURE

Mr. K.KAJENDRAN, M.C.A., M.E.,
SUPERVISOR
ASSOCIATE PROFESSOR
DEPARTMENT OF CSE,
PANIMALAR ENGINEERING COLLEGE,
NAZARATHPETTAI,
POONAMALLEE,
CHENNAI-600 123.

Certified that the above candidate(s) was/ were examined in the Anna University Project Viva-Voce Examination held on.....

INTERNAL EXAMINER **EXTERNAL EXAMINER**

ACKNOWLEDGEMENT

We express our deep gratitude to our respected Secretary and Correspondent **Dr.P.CHINNADURAI, M.A., Ph.D.,** for his kind words and enthusiastic motivation, which inspired us a lot in completing this project.

We would like to extend our heartfelt and sincere thanks to our Directors **Tmt.C.VIJAYARAJESWARI**, **Thiru.C.SAKTHIKUMAR, M.E.,** and **Tmt. SARANYASREE SAKTHIKUMAR B.E.,M.B.A.,** for providing us with the necessary facilities for completion of this project.

We also express our gratitude to our Principal **Dr.K.Mani, M.E., Ph.D.,** for his timely concern and encouragement provided to us throughout the course.

We thank the HOD of CSE Department, **Dr. S.MURUGAVALLI , M.E.,Ph.D.,** for the support extended throughout the project.

We would like to thank my **Project Guide Mr. K. KAJENDRAN, M.C.A.,M.E.,** and all the faculty members of the Department of CSE for their advice and suggestions for the successful completion of the project.

NANTHAKUMAR J J.

PRAVEEN K.

NIRMAL KUMAR N.

ABSTRACT

Reviews act as a valuable source of information for decision-making. Online e-commerce sites have provided their users to make their opinion about products and services. A huge amount of such opinions are publicly available in the form of reviews. Manufacturers, retailers as well as customers have great interest in customer reviews. A customer has an interest in such reviews to determine which product or a particular brand to buy. Manufacturers can analyze the improvement area in their product from such opinionated reviews. Due to a large number of reviews available on the internet for analysis, it is not cost worthy to read these manually. To optimize this time-consuming task there is a need for an automated system that provides the summarized result of user sentiments. Opinion Mining (OM) in the field of study that analyzes people's sentiments or opinions from reviews or opinionated text. In the sentiment analysis process, machine learning is used to analyze sentiments, emotions and produce a summarized result for decision making. Opinion Mining can be viewed as a natural language processing task, the task is to develop a system that understands the people's language. Opinion Mining is a difficult task due to the ambiguous nature of human languages(like English). In this paper, we present a comprehensive study of sentiment analysis tasks and the challenges for the sentiment analysis system. As aspect-based sentiment analysis is an active research area in SA, we will cover this level of sentiment analysis in detail.

Keywords: opinion mining, polarity, textblob (NLP)

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF ABBREVIATION

S.NO	ABBREVIATION	EXPANSION
1.	NLP	Natural Language Processing
2.	OM	Opinion Mining
3.	API	Application Programming Interface
4.	REST	Representational State Transfer
5.	MTV	Model Template Views
6.	ORM	Object Relational Mapping

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW

Natural Language Processing or NLP is a field of Artificial Intelligence that gives the machines the ability to read, understand and derive meaning from human languages.

It is a discipline that focuses on the interaction between data science and human language, and is scaling to lots of industries. Today NLP is booming thanks to the huge improvements in the access to data and the increase in computational power, which are allowing practitioners to achieve meaningful results in areas like healthcare, media, finance and human resources, among others.

USE CASES OF NLP

In simple terms, NLP represents the automatic handling of natural human language like speech or text, and although the concept itself is fascinating, the real value behind this technology comes from the use cases.

NLP can help you with lots of tasks and the fields of application just seem to increase on a daily basis. Let's mention some examples:

- NLP enables the recognition and **prediction of diseases** based on electronic health records and patient's own speech. This capability is being explored in health conditions that go from cardiovascular diseases to depression and even schizophrenia. For example, Amazon Comprehend Medical is a service that

uses NLP to extract disease conditions, medications and treatment outcomes from patient notes, clinical trial reports and other electronic health records.

- Organizations can determine what customers are saying about a service or product by identifying and extracting information in sources like social media. This **sentiment analysis** can provide a lot of information about customers choices and their decision drivers.

- Companies like Yahoo and Google filter and classify your emails with NLP by analyzing text in emails that flow through their servers and **stopping spam** before they even enter your inbox.

- To help **identifying fake news**, the NLP Group at MIT developed a new system to determine if a source is accurate or politically biased, detecting if a news source can be trusted or not.

- Amazon's Alexa and Apple's Siri are examples of intelligent **voice driven interfaces** that use NLP to respond to vocal prompts and do everything like find a particular shop, tell us the weather forecast, suggest the best route to the office or turn on the lights at home.

NLP is particularly booming in the **healthcare industry**. This technology is improving care delivery, disease diagnosis and bringing costs down while healthcare organizations are going through a growing adoption of electronic health records. The fact that clinical documentation can be improved means that patients can be better understood and benefited through better healthcare. The goal should be to optimize their experience, and several organizations are already working on this.

Basically, they allow developers and businesses to create a software that understands human language. Due to the complicated nature of human language, NLP can be difficult to learn and implement correctly. However, with the knowledge gained from this article, you will be better equipped to use NLP successfully, no matter your use case.

Natural language processing has a wide range of applications in business.

As just one example, brand sentiment analysis is one of the top use cases for NLP in business. Many brands track sentiment on social media and perform social media sentiment analysis. In social media sentiment analysis, brands track conversations online to understand what customers are saying, and glean insight into user behavior.

“One of the most compelling ways NLP offers valuable intelligence is by tracking sentiment — the tone of a written message (tweet, Facebook update, etc.) — and tag that text as positive, negative or neutral,” says Rehling.

Similarly, Facebook uses NLP to track trending topics and popular hashtags.

CHAPTER 2

LITERATURE SURVEY

TITLE SENTIMENT ANALYSIS ON TWITTER DATA

DESCRIPTION

Sentiment analysis is a type of natural language processing for tracking the mood of the public about a particular product or topic. Sentiment analysis, which is also called opinion mining, involves in building a system to collect and examine opinions about the product made in blog posts, comments, reviews or tweets. Sentiment analysis can be useful in several ways. In fact, it has spread from computer science to management sciences and social sciences due to its importance to business and society as a whole. In recent years, industrial activities surrounding sentiment analysis have also thrived. Numerous startups have emerged. Many large corporations have built their own in-house capabilities. Sentiment analysis systems have found their applications in almost every business and social domain. The goal of this report is to give an introduction to this fascinating problem and to present a framework which will perform sentiment analysis on online mobile phone reviews by associating modified K means algorithm with Naïve bayes classification and KNN.

TITLE: Sentiment Analysis of Twitter Data using Machine Learning Approaches

DESCRIPTION:

Twitter is a micro-blogging website that has become increasingly popular with the network community. Users update short messages, also known as Tweets, which are limited to 140 characters. Users update their personal opinions on many subjects, discuss current topics and write about life events through tweets. This platform is favored by many users because it has no political and economic restrictions and is easily available to large number of people. As the amount of users increase, micro-blogging platforms are becoming a place to find strong viewpoints and sentiment. People use twitter to forecast and analyze in a lot of different areas. For example, people have already forecasted the stock market success by using data from Twitter [7]. People use Twitter to forecast popularity and sales revenue of electronic products. From these case studies, we can know that Twitter is really useful for predicting products, services, or markets. It is one important reason why Twitter is taken into consideration to predict how people think about the popularity of day to day products. Another reason is because Twitter serves as a worthy platform for sentiment analysis due to its large user base with people across the world having different perspective. Twitter contains enormous amount of tweets, with millions being added every day. This can be easily collected through its APIs, which makes it easy to build a training set.

TITLE: Twitter Data Analysis

DESCRIPTION: The widespread and different types of information on Twitter make it one of the most appropriate virtual environments for information monitoring and tracking. In this paper, the authors review different information analysis techniques; starting with the analysis of different hashtags, twitter's network-topology, event spread over the network, identification of influence, and finally analysis of sentiment. Future research and development work will be addressed.

TITLE: Sentiment Analysis Techniques Involving

DESCRIPTION:

Activities that take place or are influenced as a result of decisions being made are influenced by opinions at the root level. Analysis of opinions or sentiment analysis plays a vital role in trying to make as close approximation as possible. This is an extremely important aspect given that carefully planned and executed sentiment analyses can yield better and more accurate forecasts in politics as well as in business. At the base level, sentiment analysis stems from opinions shared or expressed by individuals and users. In the Internet space that permeates almost every known sphere and area of human activity on our planet, data in millions of bytes are posted and shared by individuals on social networking platforms, blogs, product review sites, and various other web forums. The potential to harvest such information and analyse the data can yield vital insights into how products, services, political personalities, companies, governments, etc. are perceived and viewed. Sentiment Analysis can engage multiple challenges such as accuracy-related issues, binary classification problem, data sparsity problem and

polarity shift. While there have been various methods that have been postulated and developed for sentiment analysis, there yet remains to be an efficient approach in extracting and producing accurate sentiment analysis on a consistent basis. Although machine learning algorithms have come a long way, with Naïve Bayes, Support Vector Machine and Maximum Entropy being the significant ones to feature prominently in research and mainstream use, sentiment classification by category involving positive and negative sentiments, is a topic of research interest in its own right. This paper presents a survey on prominent Sentiment Analysis approaches and methodologies and seeks to submit a clear evaluation report upon which grounds for further research can be based

CHAPTER 3

SYSTEM ANALYSIS

3.1 EXISTING SYSTEM

Twitter, with 500 million users and million messages per day, has quickly become a valuable asset for organizations to invigilate their reputation and brands by extracting and analyzing the sentiment of the tweets by the public about their products, services market, and even about competitors highlighted that, from the social media generated opinions with the mammoth growth of the world wide web, super volumes of opinion texts in the form of tweets, reviews, blogs or any discussion groups and forums are available for data analysis, thus making the world wide web the very fast, most comprising and easily accessible medium for Opinion Mining.

3.2 PROPOSED SYSTEM

Imagine you just launched a new product feature and notice a sharp increase in mentions on Twitter. Are customers tweeting more because they are delighted with the new feature? Or, are they actually complaining about the feature?

Going through each of these comments manually would take far too much time. You did miss out on valuable feedback that could help you instantly improve a customer's experience with the latest feature (bug issues, user experience).

By performing analysis with NLP, you can quickly understand the tone and context of social mentions on Twitter.

3.3 FEASIBILITY STUDY

A feasibility study is carried out to select the best system that meets performance requirements. The main aim of the feasibility study

activity is to determine that it would be financially and technically feasible to develop the product.

3.3.1 TECHNICAL FEASIBILITY

This is concerned with specifying the software will successfully satisfy the user requirement. Open source and business-friendly and it is truly cross platform, easily deployed and highly extensible.

3.3.2 ECONOMIC FEASIBILITY

Economic analysis is the most frequently used technique for evaluating the effectiveness of a proposed system. The enhancement of the existing system doesn't incur any kind of drastic increase in the expenses. Python is open source and ready available for all users. Since the project is runned in python and jupyter notebook hence is cost efficient.

3.4 SYSTEM CONFIGURATION

3.4.1 HARDWARE CONFIGURATION

- Processor - Intel Core i3
- RAM - 4 GB
- Hard Disk - 500 GB

3.4.2 SOFTWARE CONFIGURATION

- Operating System - Windows 7/8/10
- Programming Language : Python 3.x
- Django, tweepy, textblob
- Twitter API Credentials
- Chartjs

3.5 SOFTWARE SPECIFICATION

3.5.1 NLP Analysis

Natural Language Processing (NLP) refers to AI method of communicating with an intelligent systems using a natural language such as English.

Processing of Natural Language is required when you want an intelligent system like robot to perform as per your instructions, when you want to hear decision from a dialogue based clinical expert system, etc.

The field of NLP involves making computers to perform useful tasks with the natural languages humans use. The input and output of an NLP system can be −

- Speech
- Written Text

Their application to Natural Language Processing (NLP) was less impressive at first, but has now proven to make significant contributions, yielding state-of-the-art results for some common NLP tasks. Named entity recognition (NER), part of speech (POS) tagging or sentiment analysis are some of the problems where neural network models have outperformed traditional approaches. The progress in machine translation is perhaps the most remarkable among all.

NLP is a tool for computers to analyse, comprehend, and derive meaning from natural language in an intelligent and useful way. This goes way beyond the most recently developed chatbots and smart virtual assistants. In fact, natural language processing algorithms are everywhere from search, online translation, spam filters and spell checking.

3.6 API

API is an abbreviation for Application Programming Interface which is a collection of communication protocols and subroutines used by various programs to communicate between them. A programmer can make use of various API tools to make its program easier and simpler. Also, an API facilitates the programmers with an efficient way to develop their software programs.

Thus in simpler terms, an API helps two programs or applications to communicate with each other by providing them with necessary tools and functions. It takes the request from the user and sends it to the service provider and then again sends the result generated from the service provider to the desired user.

A developer extensively uses API's in his software to implement various features by using an API call without writing the complex codes for the same. We can create an API for an operating system, database systems, hardware system, for a JavaScript file or similar object oriented files. Also, an API is similar to a GUI(Graphical User Interface) with one major difference. Unlike GUI's, an API helps the software developers to access the web tools while a GUI helps to make a program easier to understand by the users.

Real life example of an API:

Suppose, we are searching for a hotel room on an online website. In this case, you have a vast number of options to choose from and this may include the hotel location, the check-in and check-out dates, price, accommodation details and many more factors. So in order to book the room online, you need to interact with the hotel booking's website which in further will let you know if there is a room

available on that particular date or not and at what price.

Now in the above example, the API is the interface that actually communicates in between. It takes the request of the user to the hotel booking's website and in turn returns back the most relevant data from the website to the intended user. Thus, we can see from this example how an API works and it has numerous applications in real life from switching on mobile phones to maintaining a large amount of databases from any corner of the world.

There are various kinds of API's available according to their uses and applications like the Browser API which is created for the web browsers to abstract and to return the data from surroundings or the Third party API's, for which we have to get the codes from other sites on the web(e.g. Facebook, Twitter).

ADVANTAGES OF APIS

- **Efficiency:** API produces efficient, quicker and more reliable results than the outputs produced by human beings in an organization.
- **Flexible delivery of services:** API provides fast and flexible delivery of services according to developers requirements.
- **Integration:** The best feature of API is that it allows movement of data between various sites and thus enhances integrated user experience.
- **Automation:** As API makes use of robotic computers rather than humans, it produces better and automated results.
- **New functionality**: While using API the developers find new tools and functionality for API exchanges.

DISADVANTAGES OF APIS

- **Cost:** Developing and implementing API is costly at times and requires high maintenance and support from developers.
- **Security issues:** Using API adds another layer of surface which is then prone to attacks, and hence the security risk problem is common in API's.

3.6.1 END POINTS

What is an API Endpoint?

Simply put, an endpoint is one end of a communication channel. When an API interacts with another system, the touchpoints of this communication are considered endpoints. For APIs, an endpoint can include a URL of a server or service. Each endpoint is the location from which APIs can access the resources they need to carry out their function.

APIs work using 'requests' and 'responses.' When an API requests information from a web application or web server, it will receive a response. The place that APIs send requests and where the resource lives, is called an endpoint.

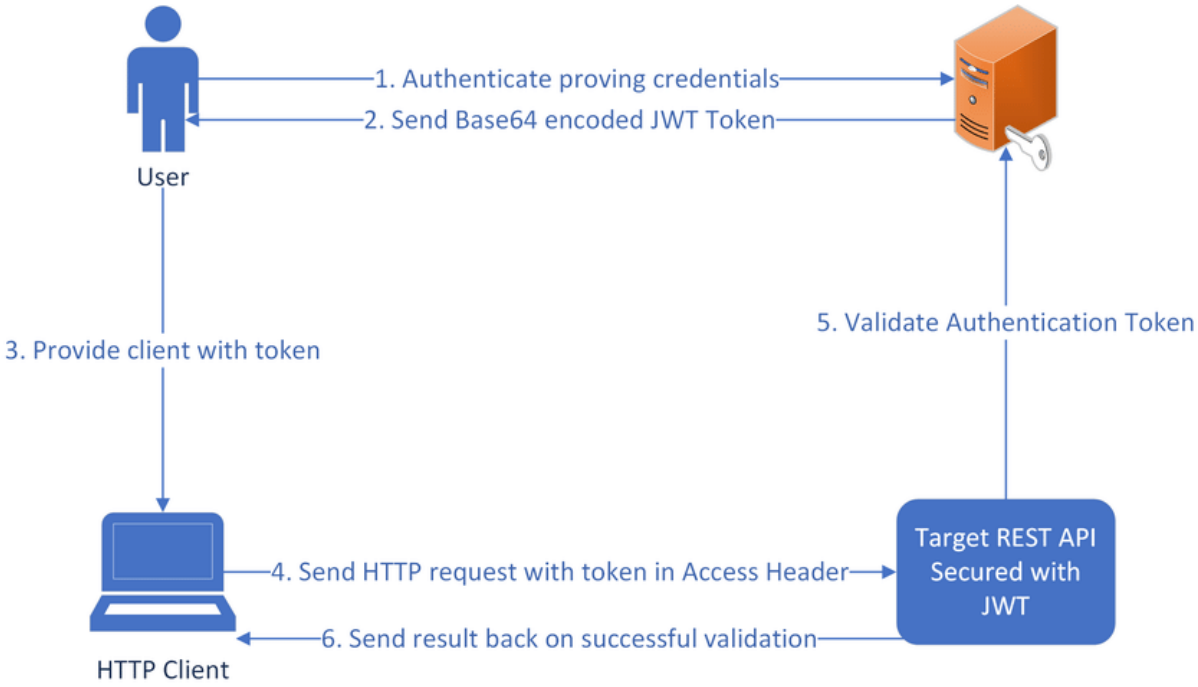
Why Are API Endpoints Important?

All over the world, companies leverage APIs to transfer vital information, processes, transactions, and more. API usage will only increase as time goes on, and making sure that each touchpoint in API communication is intact is vital to the success of each API. Endpoints specify where resources can be accessed by APIs and play a key role in guaranteeing the correct functioning of the software that

interacts with it. In short, API performance relies on its ability to communicate effectively with API Endpoints.

Do I Need to Monitor API Endpoints?

YES. Understanding how each API is performing can drastically change the way you're able to capture the value APIs add to your business. Proactively Monitoring APIs can ensure that you're able to find issues before real users experience them.



REST stands for REpresentational State Transfer and API stands for Application Program Interface. REST is a software architectural style that defines the set of rules to be used for creating web services. Web services which follow the REST architectural style are known as RESTful web services. It allows requesting systems to access and manipulate web resources by using a uniform and predefined set of rules. Interaction in REST based systems happen through Internet's Hypertext Transfer Protocol (HTTP).

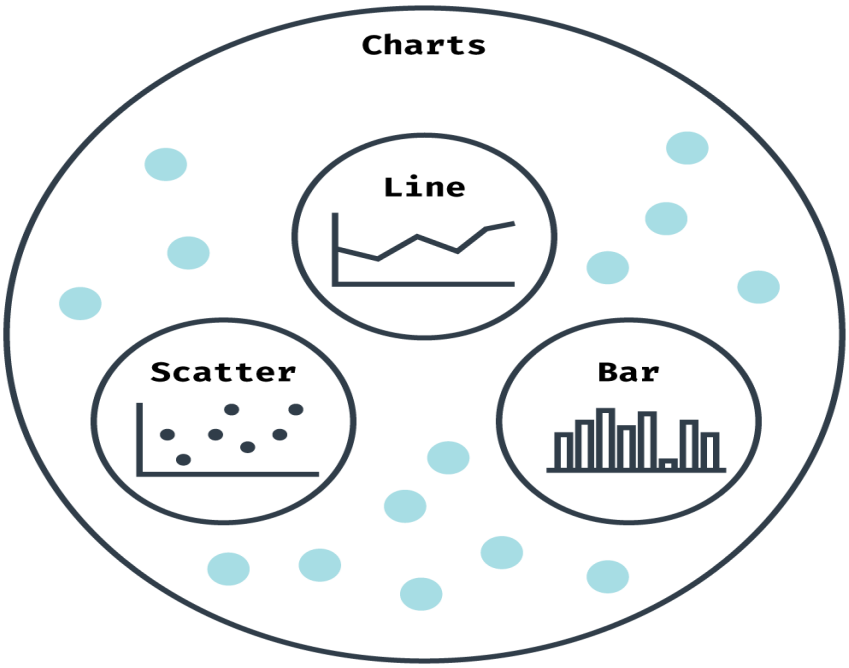
A Restful system consists of a:

- client who requests for the resources.
- server who has the resources.

It is important to create REST API according to industry standards which results in ease of development and increase client adoption.

3.7 VISUALIZATION

Data visualization is a way of exploring **complex patterns** or **large quantities** of data that cannot be easily perceived by looking at a table of numbers or reading paragraphs of text. The goal of data visualization is to communicate information more clearly, and it does so by employing our innate ability to recognize visual patterns in our environment.



Some data visualizations are **exploratory** in that they are created before any analysis is done on the data. Looking at a visual representation of our dataset can give us clues about what to focus on during analysis.

Some data visualizations are **communicative** in that they are created in order to present our analysis findings to an audience. Using visual patterns to represent patterns in data can be an effective way of explaining complex results.

Ultimately, data visualizations can more effectively answer questions, tell stories and put forth arguments than words alone.

In the world of Big Data, data visualization tools and technologies are essential to analyze massive amounts of information and make data-driven decisions.

Our eyes are drawn to colors and patterns. We can quickly identify red from blue, square from circle. Our culture is visual, including everything from art and advertisements to TV and movies.

Data visualization is another form of visual art that grabs our interest and keeps our eyes on the message. When we see a chart, we quickly see trends and outliers. If we can see something, we internalize it quickly. It's storytelling with a purpose. If you've ever stared at a massive spreadsheet of data and couldn't see a trend, you know how much more effective a visualization can be.

Big Data is here and we need to know what it says

As the "age of Big Data" kicks into high-gear, visualization is an increasingly key tool to make sense of the trillions of rows of data generated every day. Data visualization helps to tell stories by curating data into a form easier to understand,

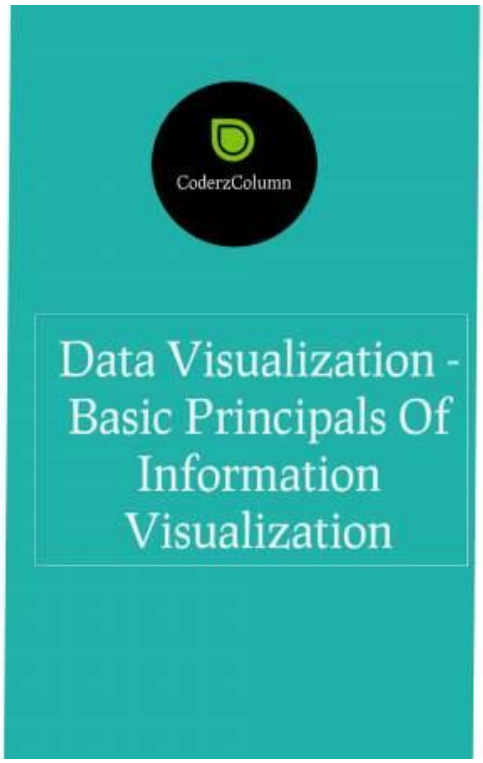
highlighting the trends and outliers. A good visualization tells a story, removing the noise from data and highlighting the useful information.

However, it's not simply as easy as just dressing up a graph to make it look better or slapping on the "info" part of an infographic. Effective data visualization is a delicate balancing act between form and function. The plainest graph could be too boring to catch any notice or it make tell a powerful point; the most stunning visualization could utterly fail at conveying the right message or it could speak volumes. The data and the visuals need to work together, and there's an art to combining great analysis with great storytelling.

Why data visualization is important for any career

It's hard to think of a professional industry that doesn't benefit from making data more understandable. Every STEM field benefits from understanding data—and so do fields in government, finance, marketing, history, consumer goods, service industries, education, sports, and so on.

While we'll always wax poetically about data visualization (you're on the Tableau website, after all) there are practical, real-life applications that are undeniable. And, since visualization is so prolific, it's also one of the most useful professional skills to develop. The better you can convey your points visually, whether in a dashboard or a slide deck, the better you can leverage that information.



The concept of the citizen data scientist is on the rise. Skill sets are changing to accommodate a data-driven world. It is increasingly valuable for professionals to be able to use data to make decisions and use visuals to tell stories of when data informs the who, what, when, where, and how. While traditional education typically draws a distinct line between creative storytelling and technical analysis, the modern professional world also values those who can cross between the two: data visualization sits right in the middle of analysis and visual storytelling.

CHAPTER 4

ARCHITECTURE

4.1 SYSTEM ARCHITECTURE

System architecture is the conceptual model that defines the structure, behavior, and more views of a system. An architecture description is a formal description and representation of a system, organized in a way that supports reasoning about the structures and behaviors of the system.

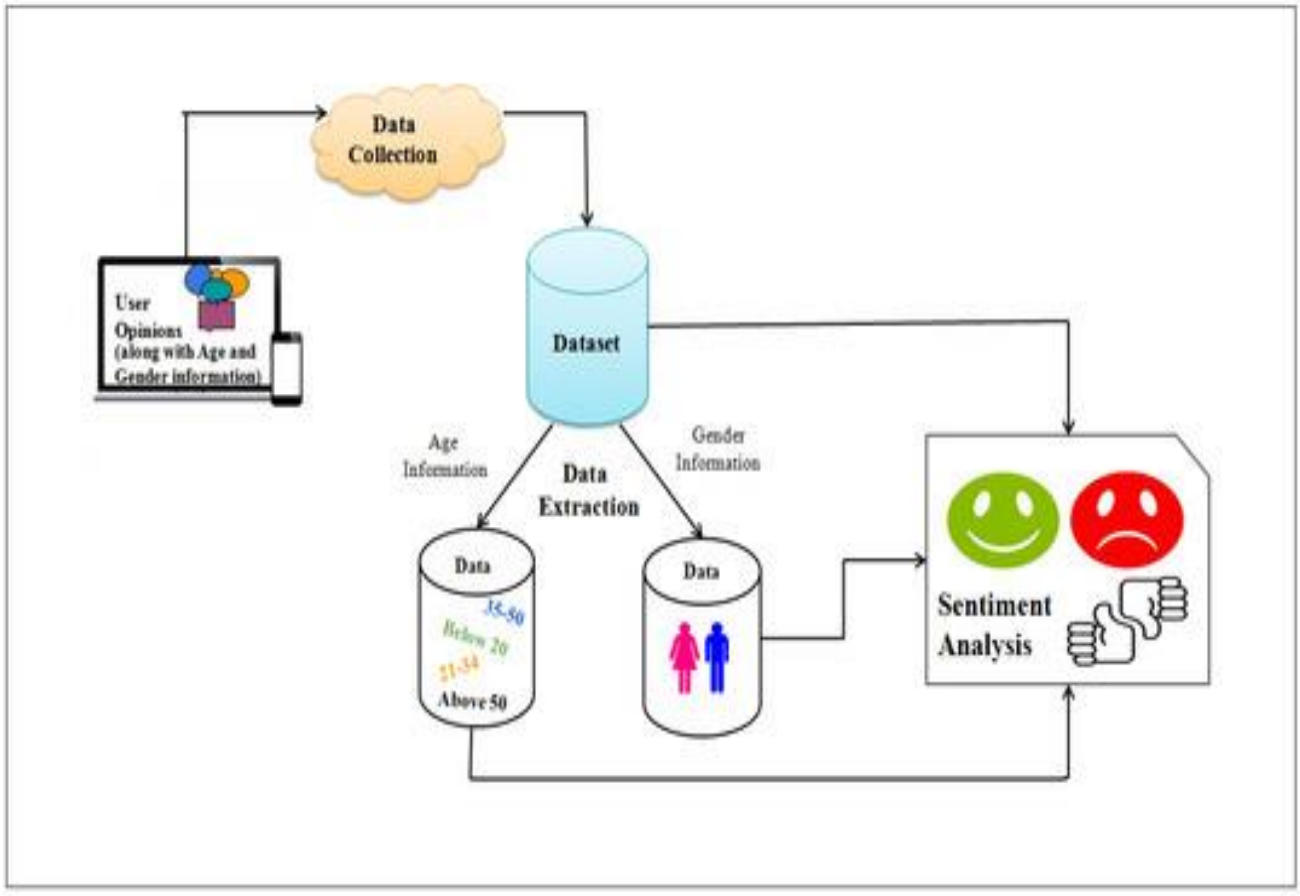


Fig 4.1.1 Architecture Diagram

4.2 UML DIAGRAMS

4.2.1 USE CASE DIAGRAM

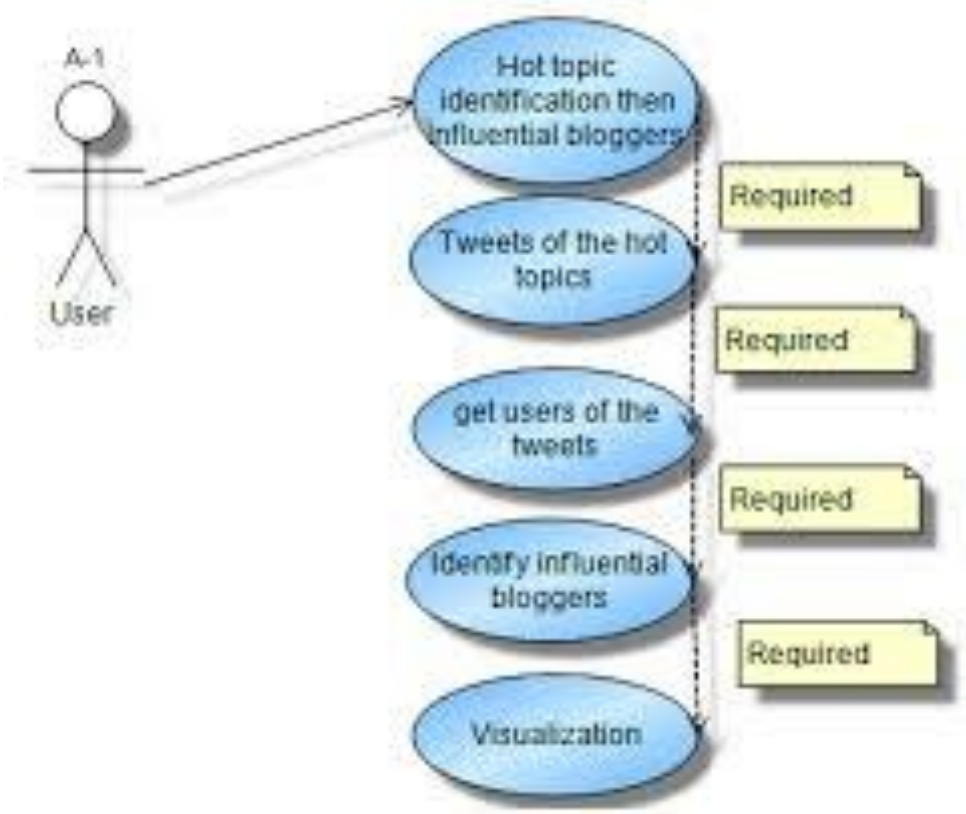


Fig 4.2.1 Use Case Diagram

4.2.2 ACTIVITY DIAGRAM

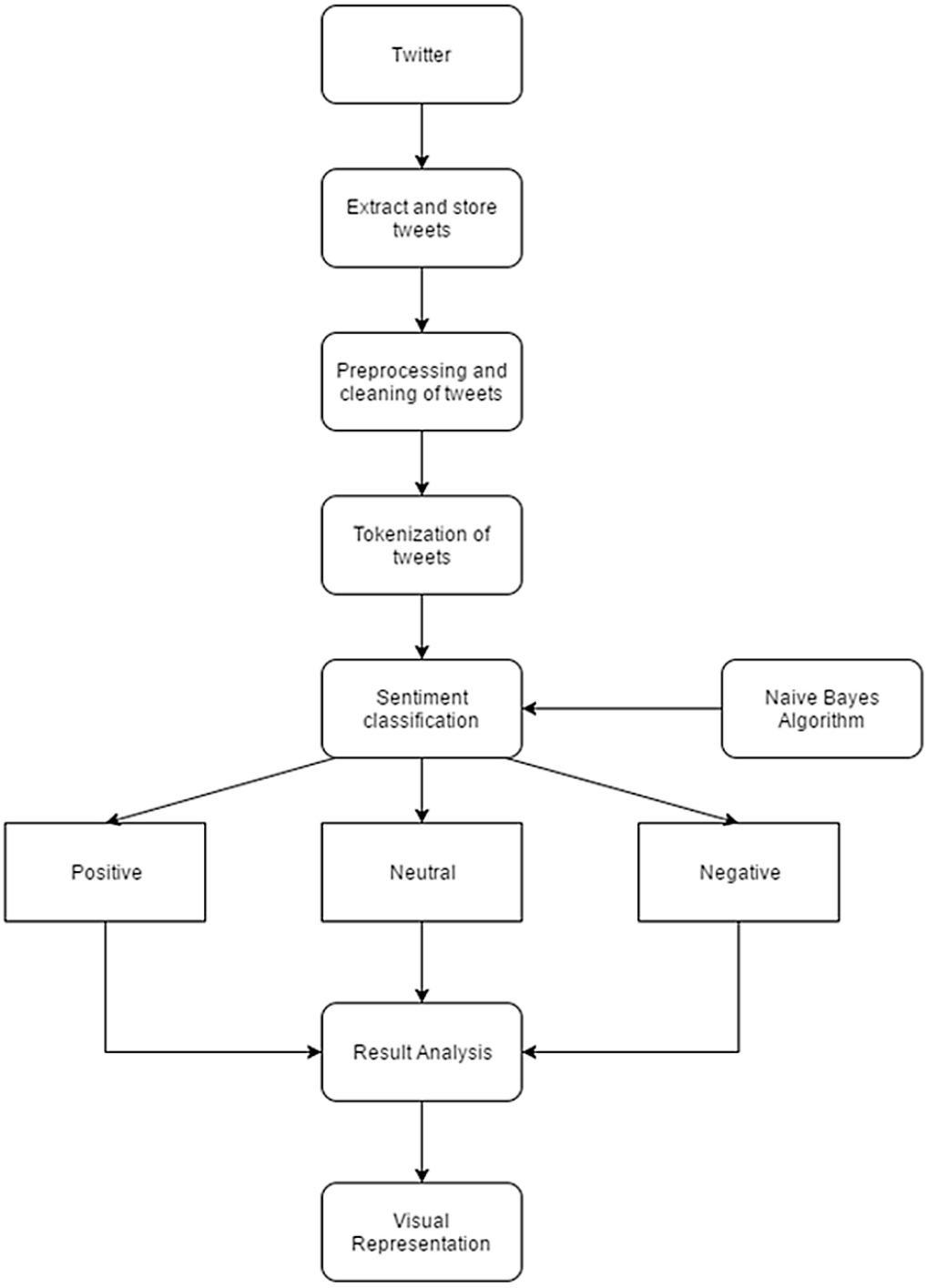


Fig 4.2.2 Activity Diagram

4.2.3 SEQUENCE DIAGRAM

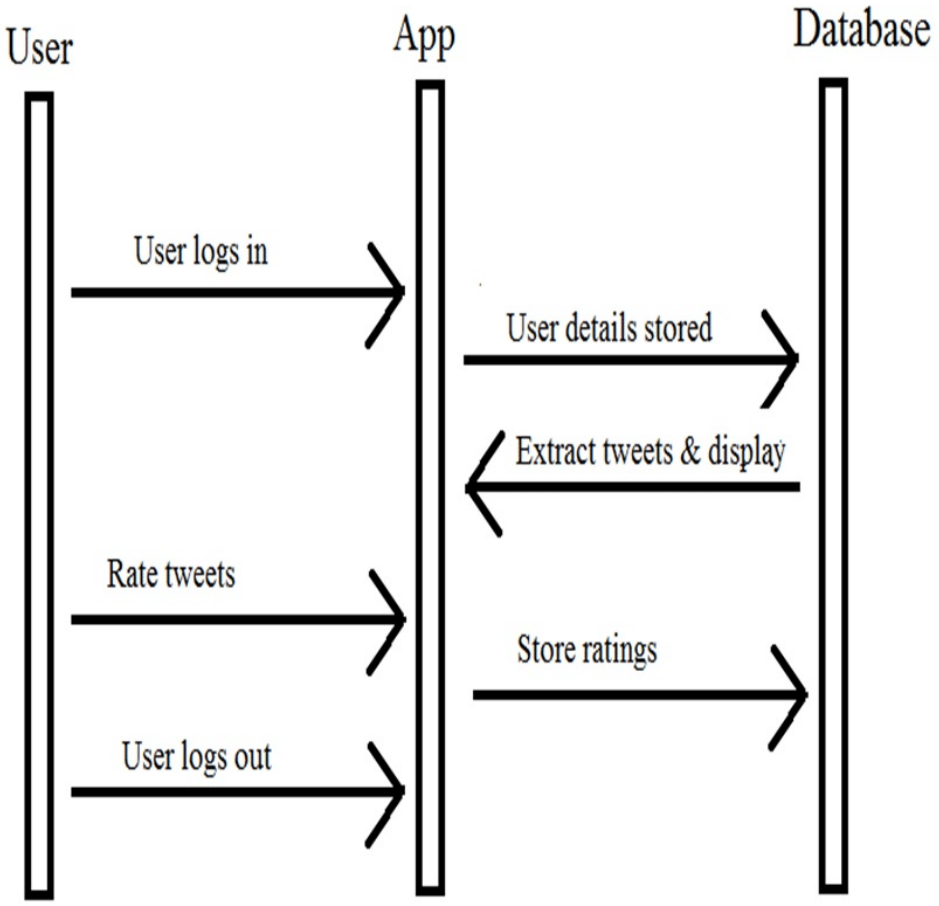


Fig 4.2.3 Sequence Diagram

4.2.3 STATE CHART DIAGRAM

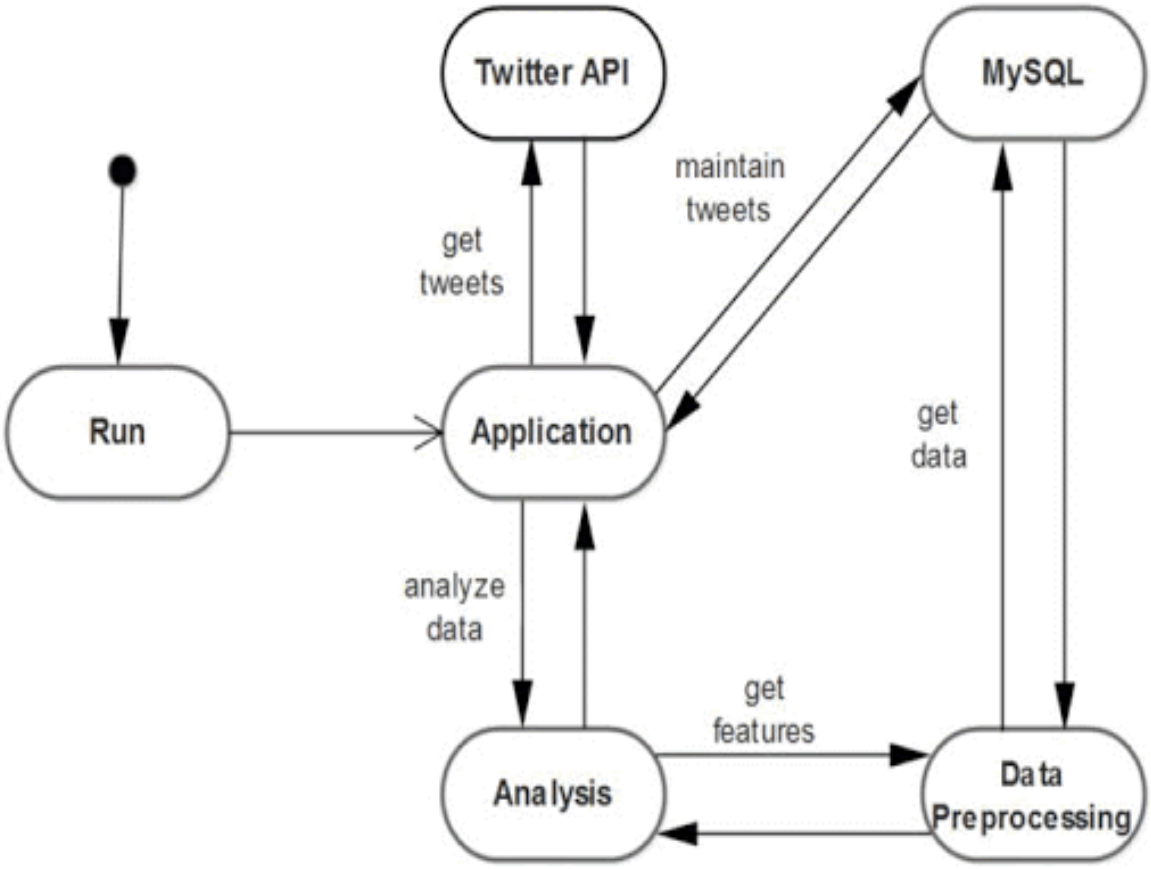


Fig 4.2.3 Statechart Diagram

CHAPTER 5

SYSTEM MODULE

5.1 MODULE

- Twitter Data Collection
- Data Preprocessing
- Text Blob

5.2 MODULE DESCRIPTION

5.2.1 TWITTER DATA COLLECTION

Twitter is a gold mine of data. Unlike other social platforms, almost every user's tweets are completely public and pullable. This is a huge plus if you're trying to get a large amount of data to run analytics on. Twitter data is also pretty specific. Twitter's API allows you to do complex queries like pulling every tweet about a certain topic within the last twenty minutes, or pull a certain user's non-retweeted tweets.

A simple application of this could be analyzing how your company is received in the general public. You could collect the last 2,000 tweets that mention your company (or any term you like), and run a sentiment analysis algorithm over it.

Twitter's Developer Policy is generally interpreted as allowing sharing of tweets locally, i.e., within an academic institution. For example, we share the datasets we have collected at GW Libraries with members of the GW research community (but when sharing outside the GW community, we only share the tweet ids). However, only a small number of institutions proactively collect Twitter data – your library is a good place to inquire.

Another option for acquiring an existing Twitter dataset is TweetSets, a web application that I've developed. TweetSets allows you to create your own dataset by querying and limiting an existing dataset. For example, you can create a dataset that only contains original tweets with the term "trump" from the Women's March dataset. If you are local, TweetSets will allow you to download the complete tweet; otherwise, just the tweet ids can be downloaded. Currently, TweetSets includes nearly a half billion tweets.

We can also target users that specifically live in a certain location, which is known as spatial data. Another application of this could be to map the areas on the globe where your company has been mentioned the most.

As you can see, Twitter data can be a large door into the insights of the general public, and how they receive a topic. That, combined with the openness and the generous rate limiting of Twitter's API, can produce powerful results.

5.2.2 DATA PREPROCESSING

Data preprocessing is an important tool for Data Mining (DM) algorithm. Twitter data is an unstructured data set it is a collection of information from people entered his/her feelings, opinion, attitudes, products review, emotions, etc. This type of information is growing day by day in the internet. May companies want to analyze customers opinions which like the product and the services. The Proposed work to analyses the twitter trending information and collect various different information form the users. It improves the accuracy of Twitter data. This work easy to identify the people reaction or opinion. Additionally, improve the better performance for data preprocessing tool.

5.2.3 TEXTBLOB

TextBlob is a Python (2 and 3) library for processing textual data. It provides a simple API for diving into common natural language processing (NLP) tasks such as part-of-speech tagging, noun phrase extraction, sentiment analysis, classification, translation, and more.

Install TextBlob using the following commands in terminal:

pip install -U textblob

This will install TextBlob and download the necessary NLTK corpora. The above installation will take quite some time due to the massive amount of tokenizers, chunkers, other algorithms, and all of the corpora to be downloaded.

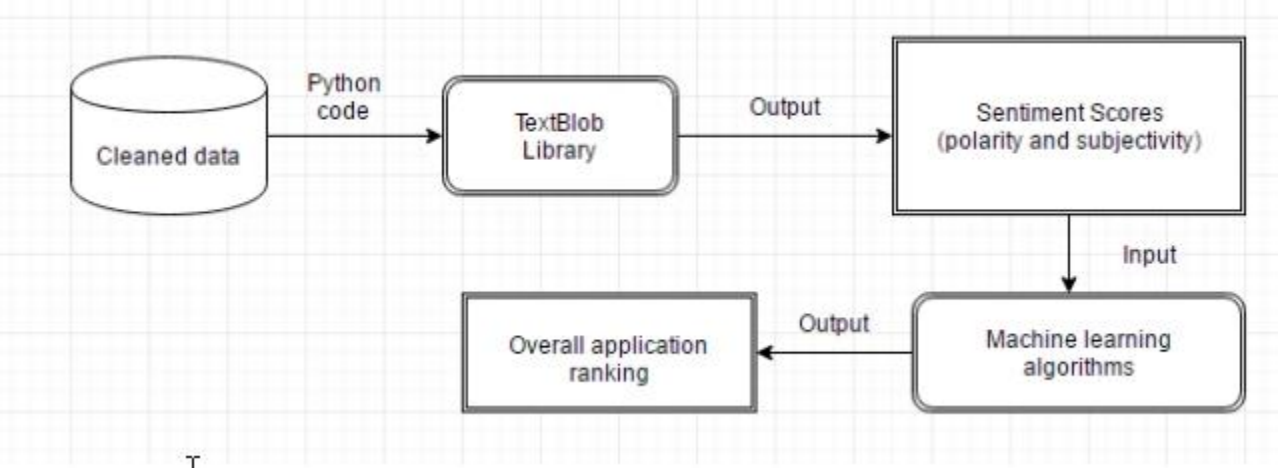
Some terms that will be frequently used are :

- **Corpus** – Body of text, singular. Corpora is the plural of this.
- **Lexicon** – Words and their meanings.

- **Token** – Each "entity" that is a part of whatever was split up based on rules. For examples, each word is a token when a sentence is "tokenized" into words. Each sentence can also be a token

So basically tokenizing involves splitting sentences and words from the body of the text.

The approach that the TextBlob package applies to sentiment analysis differs in that it's rule-based and therefore requires a pre-defined set of categorized words. These words can, for example, be uploaded from the NLTK database. Moreover, sentiments are defined based on semantic relations and the frequency of each word in an input sentence that allows getting a more precise output as a result.



TextBlob's output for a **polarity** task is a float within the range [-1.0, 1.0] where -1.0 is a negative polarity and 1.0 is positive. This score can also be equal to 0, which stands for a neutral evaluation of a statement as it doesn't contain any words from the training set.

Whereas, a **subjectivity/objectivity** identification task reports a float within the range [0.0, 1.0] where 0.0 is a very objective sentence and 1.0 is very subjective.

CHAPTER 6

SYSTEM DESIGN

6.1 TWITTER API CODING

```
import re

import tweepy

from tweepy import OAuthHandler

from textblob import TextBlob


class TwitterClient(object):
    '''
    Generic Twitter Class for sentiment analysis.
    '''


    def __init__(self):
        '''
        Class constructor or initialization method.
        '''
        # keys and tokens from the Twitter Dev Console
        consumer_key = 'JwEbbM7QzpPhjB9BxnhhLM9aJ'
```

```
        consumer_secret =
'4MeImeRZlM0TdRkyOMJbSZ8x5UowD6UKNwZP4d8mBRAkerzoPl'
        access_token = '942033965182984193-
LxNxjkm9FTdgt5AUZTjTV0OpUpoyIr5'
        access_token_secret =
'D5F4C8JfcPCvMbGAuEhb74S8gO5TPNYioIIkVXWdbxcGP'

        # attempt authentication
        try:
            # create OAuthHandler object
            self.auth = OAuthHandler(consumer_key, consumer_secret)
            # set access token and secret
            self.auth.set_access_token(access_token, access_token_secret)
            # create tweepy API object to fetch tweets
            self.api = tweepy.API(self.auth)
        except:
            print("Error: Authentication Failed")

    def clean_tweet(self, tweet):
        '''
```

```
        Utility function to clean tweet text by removing links, special characters
        using simple regex statements.
        '''
        return  ' '.join(re.sub("(@[A-Za-z0-9]+)|([^0-9A-Za-z  \t])|(\w+:\/\/\S+)",  "  ", tweet).split())

    def get_tweet_sentiment(self, tweet):
        '''
        Utility function to classify sentiment of passed tweet
        using textblob's sentiment method
        '''
        # create TextBlob object of passed tweet text
        analysis = TextBlob(self.clean_tweet(tweet))
        # set sentiment
        if analysis.sentiment.polarity > 0:
            return 'positive'
        elif analysis.sentiment.polarity == 0:
            return 'neutral'
        else:
```

```
        return 'negative'


def get_tweets(self, query):
    '''
    Main function to fetch tweets and parse them.
    '''
    # empty list to store parsed tweets
    tweets = []
    try:
        # call twitter api to fetch tweets
        fetched_tweets = self.api.search_full_archive(
            environment_name='project', query=query, maxResults=100)
        print(type(fetched_tweets))
        for tweet in fetched_tweets:
            data= bool(tweet.retweeted)
        # print(json.dumps(data['retweeted_status'],indent=4))
        # parsing tweets one by one
        for tweet in fetched_tweets:
            # empty dictionary to store required params of a tweet
```

```
            parsed_tweet = {}


            # saving text of tweet
            parsed_tweet['text'] = tweet.text
            # saving sentiment of tweet
            parsed_tweet['sentiment'] = self.get_tweet_sentiment(
                tweet.text)
            # appending parsed tweet to tweets list
            if tweet.retweet_count > 0:
                # if tweet has retweets, ensure that it is appended only once
                if parsed_tweet not in tweets:
                    tweets.append(parsed_tweet)
            else:
                tweets.append(parsed_tweet)
        # return parsed tweets
        return tweets
    except tweepy.TweepError as e:
        # print error (if any)
        print("Error : " + str(e))
```

6.2 FRONTEND CODING

Login.html

```
{% extends 'base.html' %}

{% load static %}

{% block title %}Login{% endblock %}

{% block content %}

<div class="bg-image"></div>

<!--form area start-->

<div class="form">

    <!--login form start-->

    <form class="login-form" action="" method="POST">

        {% csrf_token %}

        <h2>Login</h2>

        <div class="icons">

            <a href="#"><i class="fab fa-facebook"></i></a>

            <a href="#"><i class="fab fa-google"></i></a>

        </div>

        <input type="text" name="username" value="" placeholder="Username" required>
```

```
            <input type="password" name="password" value="" placeholder="Password" required>
            <button type="submit" name="button">Login</button>
            <p class="options">Don't have an Account? <a href="{% url 'register' %}"> Register </a></p>
        </form>
        {% for message in messages %}
        <h2 style="color: red;">{{message}}</h2>
        {% endfor %}
        <!--login form end-->
</div>
<!--form area end-->
{% endblock content %}
{% block javascript %}
<script>
        var request=require('request');
</script>

{% endblock javascript %}
```

Register.html

```
{% extends 'base.html' %}


{% load static %}
{% block title %}Register{% endblock %}
{% block content %}


<body>
    <div class="bg-image"></div>
    <div class="form">
        <form class="login-form" method="POST">
            {% csrf_token %}
            <h2>Register</h2>
            <input type="text" name="uname" placeholder="Username" required>
            <input type="email" name="email" placeholder="email" required>
            <input type="password" name="password1" placeholder="Password" required>
            <input type="password" name="password2" placeholder="Confirm Password" required>
            <button>Register</button>
```

```
        </form>
        {% for message in messages %}
        <h2 style="color: red;">{{message}}</h2>
        {% endfor %}
        <h3 class="options">Already Have an Account ? <a href="{% url 'login' %}"> Login </a></h3>
    </div>
{% endblock content %}
```

6.3 VISUALIZATIONCODING

```
function drawPieGraph(data, id) {
        var labels = data.labels;
        var chartLabel = data.chartLabel;
        var chartdata = data.chartdata;
        var canvas = document.getElementById(id)
        var ctx = canvas.getContext('2d');
        var myChart = new Chart(ctx, {
            type: 'pie',
            data: {
                labels: labels,
                datasets: [{
                    label: chartLabel,
                    data: chartdata,
                    backgroundColor: [
                        'rgba(255, 99, 132, 0.2)',
                        'rgba(54, 162, 235, 0.2)',
                        'rgba(255, 206, 86, 0.2)',
```

```
            ],
            borderColor: [
                'rgba(255, 99, 132, 1)',
                'rgba(54, 162, 235, 1)',
                'rgba(255, 206, 86, 1)',
            ],
            borderWidth: 1
        }]
    },


});
}
function drawBarGraph(data, id) {
    var labels = data.labels;
    var chartLabel = data.chartLabel;
    var chartdata = data.chartdata;
    var canvas = document.getElementById(id)
    var ctx = canvas.getContext('2d');
    var myChart = new Chart(ctx, {
```

```
type: 'bar',
data: {
    labels: labels,
    datasets: [{
        label: chartLabel,
        data: chartdata,
        backgroundColor: [
            'rgba(255, 99, 132, 0.2)',
            'rgba(54, 162, 235, 0.2)',
            'rgba(255, 206, 86, 0.2)',
        ],
        borderColor: [
            'rgba(255, 99, 132, 1)',
            'rgba(54, 162, 235, 1)',
            'rgba(255, 206, 86, 1)',
        ],
        borderWidth: 1
    }]
},
```

```
        options: {
            scales: {
                yAxes: [{
                    ticks: {
                        beginAtZero: true
                    }
                }]
            }
        }
    });
}
$('#twitter').on('submit', (e) => {
    let hashtag = e.target[0].value
    if (!hashtag.includes('#')) {
        hashtag = '#' + hashtag;
    }
    console.log(hashtag)
    e.preventDefault();
    $.ajax({
```

```
        url: `{% url 'analyze' %}`,
        data: 'hashtag=' + hashtag,
        type: 'POST',
        success: function (result) {
            console.log(result)
            drawBarGraph(result, 'myChartBar')
            drawPieGraph(result, 'myPie')
            $('#myChartBar').show();
            $('#myPie').show();
        }
    });
})
```

CHAPTER 7

TESTING

7.1 INTRODUCTION

Testing is a process of executing a program with the intent of finding an error. A good test case is one that has a high probability of finding an as-yet – undiscovered error. A successful test is one that uncovers an as-yet- undiscovered error. System testing is the stage of implementation, which is aimed at ensuring that the system works accurately and efficiently as expected before live operation commences. It verifies that the whole set of programs hang together. System testing requires a test consists of several key activities and steps for run program, string, system and is important in adopting a successful new system. This is the last chance to detect and correct errors before the system is installed for user acceptance testing

7.2 TYPES OF TESTS

7.2.1 UNIT TESTING

Unit testing involves the design of test cases that validate that the internal program logic is functioning properly, and that program inputs produce valid outputs. All decision branches and internal code flow should be validated. It is the testing of individual software units of the application .it is done after the completion of an individual unit before integration. This is a structural testing, that relies on knowledge of its construction and is invasive. Unit tests perform basic

tests at component level and test a specific business process, application, and/or system configuration. Unit tests ensure that each unique path of a business process performs accurately to the documented specifications and contains clearly defined inputs and expected results.

7.2.2 INTEGRATION TESTING

Integration tests are designed to test integrated software components to determine if they actually run as one program. Testing is event driven and is more concerned with the basic outcome of screens or fields. Integration tests demonstrate that although the components were individually satisfaction, as shown by successfully unit testing, the combination of components is correct and consistent. Integration testing is specifically aimed at exposing the problems that arise from the combination of components.

7.2.3 FUNCTIONAL TEST

Functional tests provide systematic demonstrations that functions tested are available as specified by the business and technical requirements, system documentation, and user manuals.

Functional testing is centered on the following items:

Valid Input : identified classes of valid input must be accepted.

Invalid Input : identified classes of invalid input must be rejected.

Functions : identified functions must be exercised.

Output : identified classes of application outputs must be exercised

Systems/Procedures: interfacing systems or procedures must be invoked.

Organization and preparation of functional tests is focused on requirements, key functions, or special test cases. In addition, systematic coverage pertaining to identify Business process flows; data fields, predefined processes, and successive processes must be considered for testing. Before functional testing is complete, additional tests are identified and the effective value of current tests is determined.

7.2.4 SYSTEM TESTING

System testing ensures that the entire integrated software system meets requirements. It tests a configuration to ensure known and predictable results. An example of system testing is the configuration oriented system integration test. System testing is based on process descriptions and flows, emphasizing pre-driven process links and integration points.

WHITE BOX TESTING

This testing is also called as Glass box testing. In this testing, by knowing the specific functions that a product has been design to perform test can be conducted that demonstrate each function is fully operational at the same time searching for errors in each function. It is a test case design method that uses the

control structure of the procedural design to derive test cases. Basis path testing is a white box testing.

Basis path testing:

- Flow graph notation
- Kilometric complexity
- Deriving test cases
- Graph matrices Control

BLACK BOX TESTING

Black Box Testing is testing the software without any knowledge of the inner workings, structure or language of the module being tested. Black box tests, as most other kinds of tests, must be written from a definitive source document, such as specification or requirements document, such as specification or requirements document. It is a testing in which the software under test is treated, as a black box .you cannot “see” into it. The test provides inputs and responds to outputs without considering how the software works.

7.2.5 INTEGRATION TESTING

Software integration testing is the incremental integration testing of two or more integrated software components on a single platform to produce failures caused by interface defects.

The task of the integration test is to check that components or software applications, e.g. components in a software system or – one step up – software applications at the company level – interact without error.

Test Results All the test cases mentioned above passed successfully. No defects encountered.

7.2.6 ACCEPTANCE TESTING

User Acceptance Testing is a critical phase of any project and requires significant participation by the end user. It also ensures that the system meets the functional requirements.

Test Results All the test cases mentioned above passed successfully. No defects encountered.

CHAPTER 8

CONCLUSION

8.1 APPLICATIONS

1. Social Media Monitoring

There are more than 3.5 billion active social media users; that's 45% of the world's population. Every minute users send over 500,000 Tweets and post 510,000 Facebook comments, and a large amount of these messages contain valuable business insights about how customers feel towards products, brands and services.

Sentiment analysis allows businesses to mine this data and extract the feelings that underlie social media conversations, to understand how people are talking about a given product or topic, and why.

2. Brand Monitoring

Besides social media, online conversations can take place in blogs, review websites, news websites and forum discussions. Product reviews, for instance, have become a crucial step in the buyer's journey. Consumers read at least 10 reviews before buying, and 57% only trust a business if it has a star-rating of 4 or more.

If you need to get detailed insights on different features related to your product, you should try aspect-based sentiment analysis. This will allow you to see what specific aspects of your product are being praised or criticized by customers.

3. Customer Support Analysis

Providing outstanding customer service experiences should be a priority. After all, 96% of consumers say great customer service is a key factor to choose and stay loyal to a brand.

Fortunately, sentiment analysis can help you make your customer support interactions faster and more effective.

If you run sentiment analysis on all your incoming tickets, you can easily detect the most dissatisfied customers or the most urgent issues and prioritize them above the rest.Plus, you could route tickets to the appropriate person or team in charge of dealing with them.

You can also use sentiment analysis to assess the results of your customer support strategy. Let's take this Tweet complaining about Airbnb customer support

4. Customer Feedback Analysis

Net Promoter Score (NPS) surveys are one of the most popular ways to ask for customers feedback about a product or service. Sentiment analysis of NPS surveys allows you to go beyond the numerical scores and groups (Detractors, Promoters, Passives), as well as speed up the process and obtain more consistent results than if you were tagging these results manually.

By running aspect-based sentiment analysis on a set of open-ended NPS responses, you'll gauge sentiments regarding specific features of your product. That way, you'll find out what customers appreciate and dislike most about your product.

Once your sentiment analysis process is up and running, you'll also be able to compare results with previous NPS surveys and see how sentiments toward aspects of your product have improved over time.

5. Market Research

Want to collect insights on customer feelings, experiences, and needs relating to a marketing campaign for a new product release? Sentiment analysis can help monitor online conversations about a specific marketing campaign, so you can see how it's performing.

You can also find out how customers feel about a new product. Is it already getting positive or negative feedback ahead of its release. Perhaps customers are unhappy with the pricing or would have liked to see an additional feature.

8.2 FUTURE ENHANCEMENT:

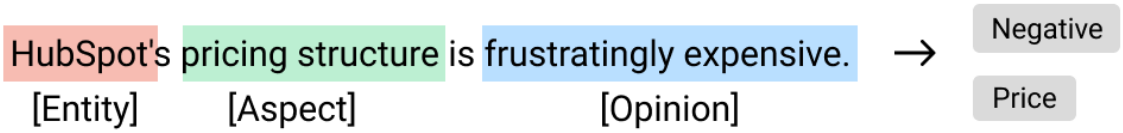
1) Emotion Detection

This sentiment analysis model detects the emotions that underlie a text. It makes associations between words and emotions like anger, happiness, frustration, etc.
For example,

- 'Hubspot makes my day a lot easier :)' → Happiness
- 'Your customer service is a nightmare! Totally useless!!' → Anger

2) Aspect-based Sentiment Analysis

This type of sentiment analysis focuses on understanding the aspects or features that are being discussed in a given opinion. Product reviews, for example, are often composed of different opinions about different characteristics of a product, like Price, UX-UI, Integrations, Mobile Version, etc. Let's see some examples:



3) Intent Detection

This type of sentiment analysis tries to find an action behind a given opinion, something that the user wants to do. Identifying user intents allows you to detect valuable opportunities to help customers, such as solving an issue, making improvements on a product or deriving complaints to the correspondent areas:

- "Very frustrated right now. Instagram keeps closing when I log in. Can you help?" → Request for Assistance

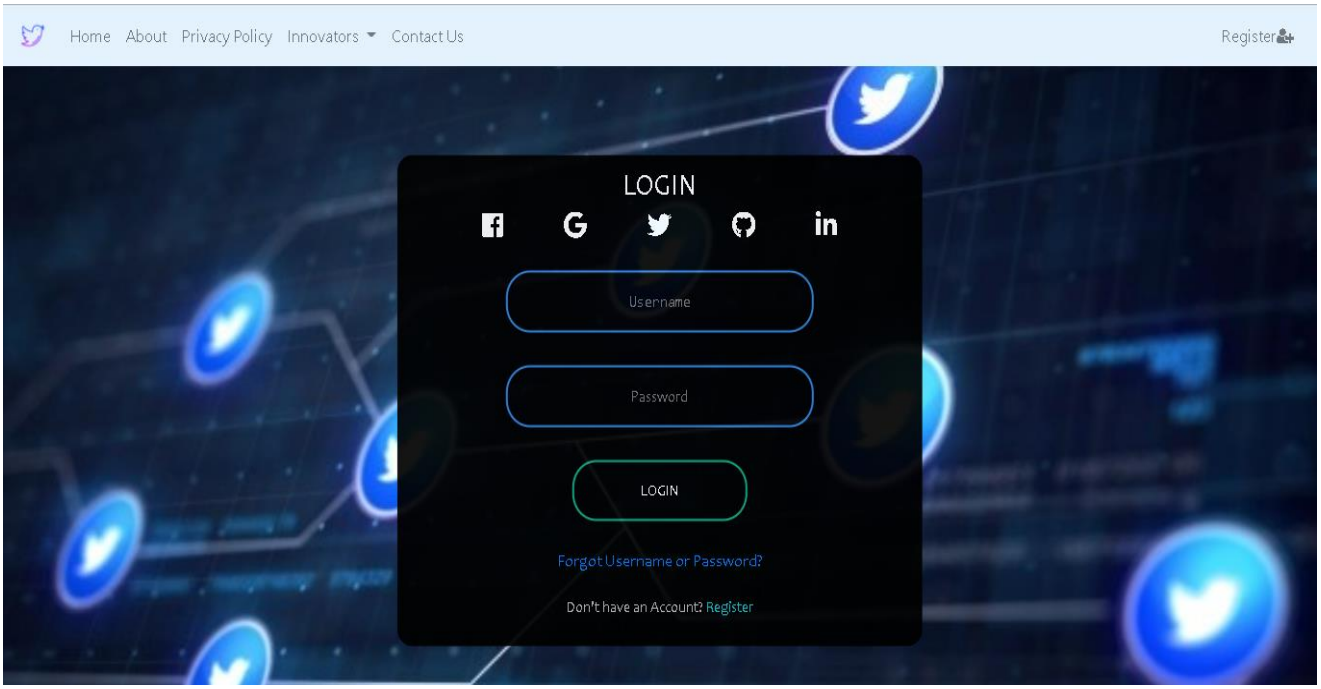
Customers experiencing issues can be easily spotted thanks to sentiment analysis.

8.3 CONCLUSION

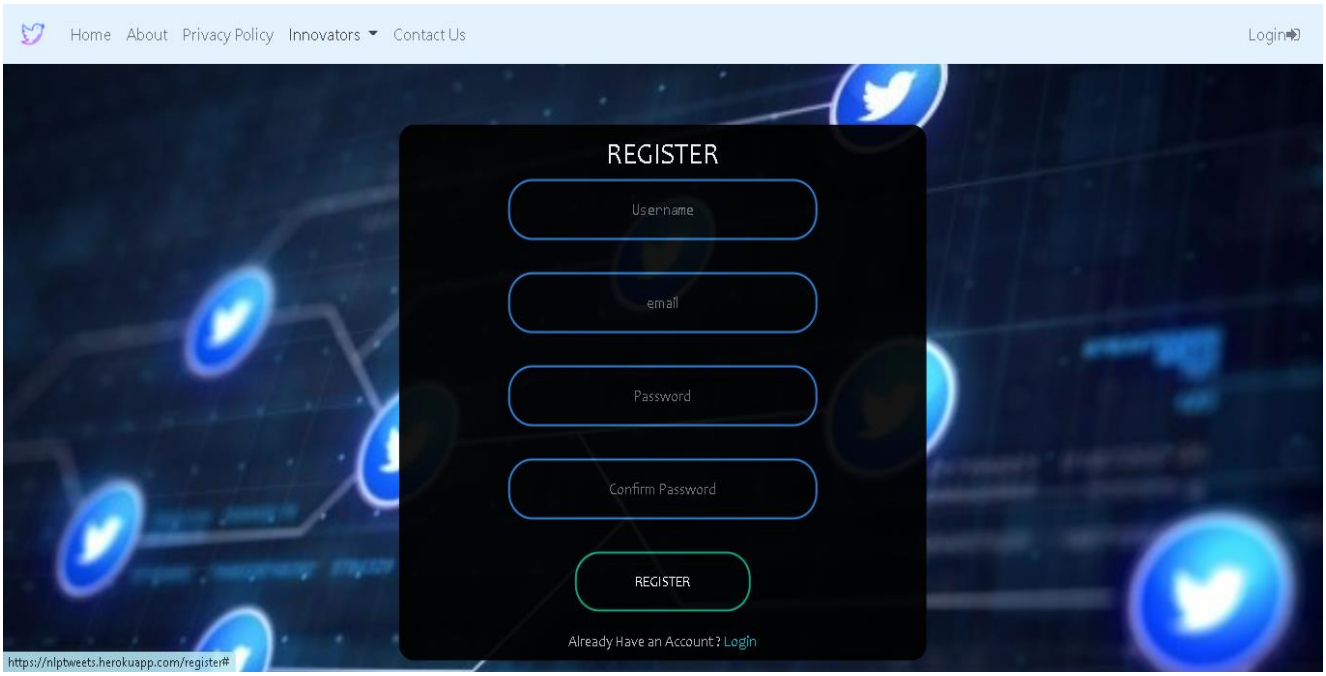
In this project we have successfully designed the web application for analysing the tweets of a particular hashtag using Opinion Mining(OM). To take advantage of both natural language and social networks relationships, a novel research branch is developing.

A.1 SAMPLE SCREENS

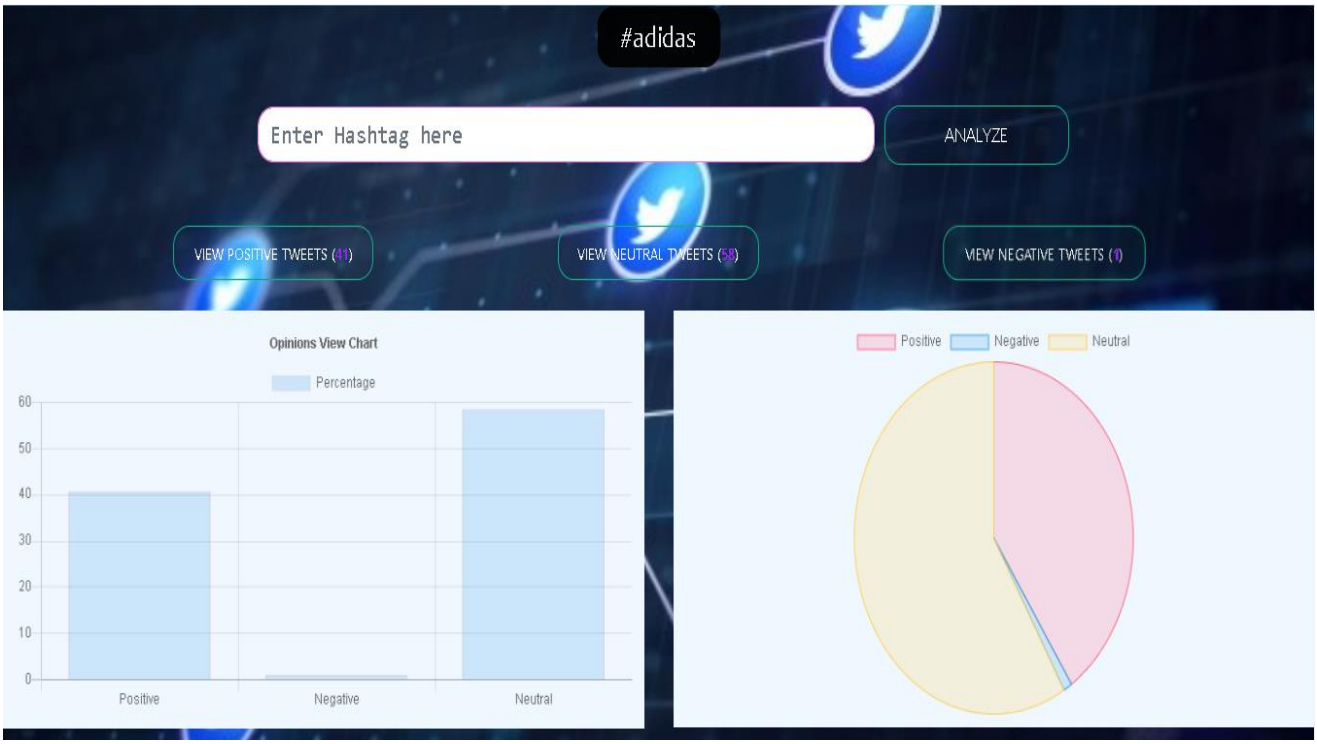
USER LOGIN SCREEN

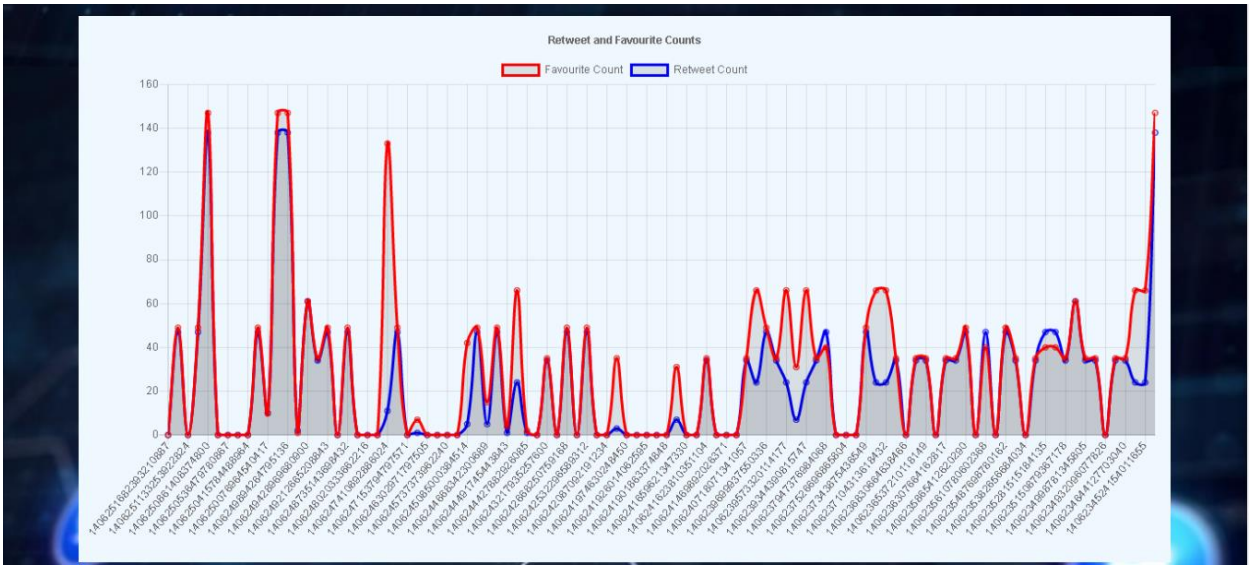


USER REGISTRATION SCREEN



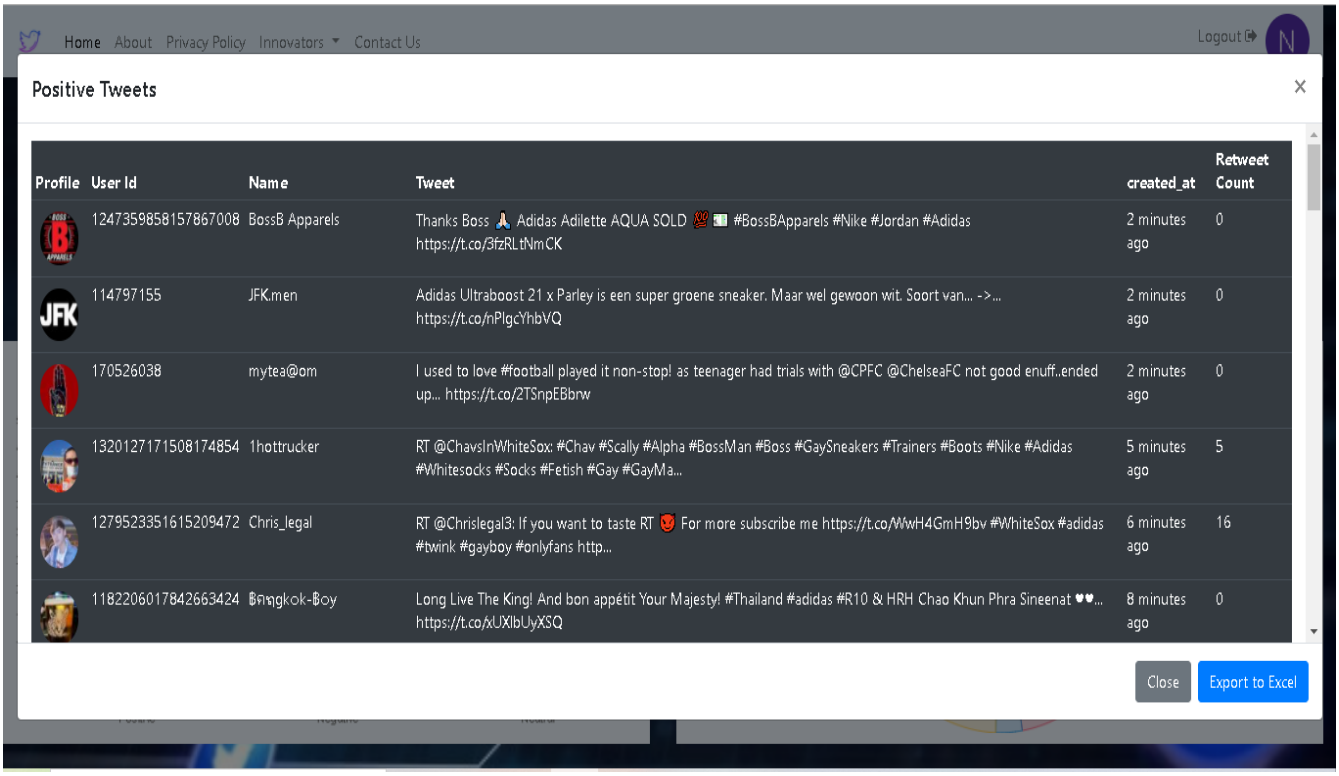
DATA VISUALIZATION SCREENS




Retweet and Favourite Counts
Favourite Count
Retweet Count
160
140
120
100
80
60
40
20
0


Opinions View by Date
Tweets Count
18
16
14
12
10
8
6
4
2
0
6/19/2021
6/18/2021
6/17/2021
6/15/2021
6/14/2021
6/9/2021
6/8/2021
6/6/2021
6/5/2021
6/4/2021
6/3/2021
6/2/2021
6/1/2021
5/31/2021
5/29/2021
5/28/2021
5/26/2021
5/25/2021
5/24/2021
5/22/2021
5/20/2021

TWEETS VIEW SCREEN



A.2 PUBLICATIONS

Onam Bharti *et al*, International Journal of Computer Science and Mobile Computing, Vol.5 Issue.6, June- 2016, pg. 601-609

Available Online at www.ijcsmc.com

International Journal of Computer Science and Mobile Computing

A Monthly Journal of Computer Science and Information Technology

ISSN 2320–088X
IMPACT FACTOR: 5.258

IJCSMC, Vol. 5, Issue. 6, June 2016, pg.601 – 609

SENTIMENT ANALYSIS ON TWITTER DATA

Onam Bharti
M.Tech (CE)
Bhartionam100@gmail.com

Mrs. Monika Malhotra
Assistant Professor
mrsmalhotra16@gmail.com

World College of Technology and Management, Farukh Nagar Gurgaon Haryana 122506

ABSTRACT: Sentiment analysis is a type of natural language processing for tracking the mood of the public about a particular product or topic. Sentiment analysis, which is also called opinion mining, involves in building a system to collect and examine opinions about the product made in blog posts, comments, reviews or tweets. Sentiment analysis can be useful in several ways. In fact, it has spread from computer science to management sciences and social sciences due to its importance to business and society as a whole. In recent years, industrial activities surrounding sentiment analysis have also thrived. Numerous startups have emerged. Many large corporations have built their own in-house capabilities. Sentiment analysis systems have found their applications in almost every business and social domain. The goal of this report is to give an introduction to this fascinating problem and to present a framework which will perform sentiment analysis on online mobile phone reviews by associating modified K means algorithm with Naïve bayes classification and KNN.

I. INTRODUCTION:

Natural Language Processing (NLP) deals with actual text element processing. The text element is transformed into machine format by NLP. Artificial Intelligence (AI) uses information provided by the NLP and applies a lot of maths to determine whether something is positive or negative. Several methods exist to determine an author's view on a topic from natural language textual information. Some form of machine learning approach is employed and which has varying degree of effectiveness. One of the types of natural language processing is opinion mining which deals with tracking the mood

of the people regarding a particular product or topic. This software provides automatic extraction of opinions, emotions and sentiments in text and also tracks attitudes and feelings on the web. People express their views by writing blog posts, comments, reviews and tweets about all sorts of different topics. Tracking products and brands and then determining whether they are viewed positively or negatively can be done using web. The opinion mining has slightly different tasks and many names, e.g. sentiment analysis, opinion extraction, sentiment mining, subjectivity analysis, affect analysis, emotion analysis, review mining, etc.

Natural language processing (NLP) is a field of computer science, artificial intelligence, and computational linguistics concerned with the interactions between computers and human (natural) languages. As such, NLP is related to the area of human–computer interaction. Many challenges in NLP involve: natural language understanding, enabling computers to derive meaning from human or natural language input; and others involve natural language generation.

Modern NLP algorithms are based on machine learning, especially statistical machine learning. The paradigm of machine learning different from that of most prior attempts at language processing. Prior implementations of language-processing tasks typically involved the direct hand coding of large sets of rules. The machine-learning paradigm calls instead for using general learning algorithms — often, although not always, grounded in statistical inference — to automatically learn such rules through the analysis of large corpora of typical real-world examples. A corpus (plural, "corpora") is a set of documents (or sometimes, individual sentences) that have been hand-annotated with the correct values to be learned. Many different classes of machine learning algorithms have been applied to NLP tasks. These algorithms take as input a large set of "features" that are generated from the input data. Some of the earliest-used algorithms, such as decision trees, produced systems of hard if-then rules similar to the systems of hand-written rules that were then common. Increasingly, however, research has focused on statistical models, which make soft, probabilistic decisions based on attaching real-valued weights to each input feature. Such models have the advantage that they can express the relative certainty of many different possible answers rather than only one, producing more reliable results when such a model is included as a component of a larger system.

Systems based on machine-learning algorithms have many advantages over hand-produced rules: The learning procedures used during machine learning automatically focus on the most common cases, whereas when writing rules by hand it is often not at all obvious where the effort should be directed. Automatic learning procedures can make use of statistical inference algorithms to produce models that are robust to unfamiliar input (e.g. containing words or structures that have not been seen before) and to erroneous input (e.g. with misspelled words or words accidentally omitted). Generally, handling such input gracefully with hand-written rules — or more generally, creating systems of hand-written rules that make soft decisions — is extremely difficult, error-prone and time-consuming. Systems based on automatically learning the rules can be made more accurate simply by supplying more input data. However, systems based on hand-written rules can only be made more accurate by increasing the complexity of the rules, which is a much more difficult task. In particular, there is a limit to the complexity of systems based on hand-crafted rules, beyond which the systems become more and more unmanageable. However, creating more data to input to machine-learning systems simply requires a corresponding increase in the number of man-hours worked, generally without significant increases in the complexity of the annotation process. The subfield of NLP devoted to learning approaches is known as Natural Language Learning (NLL) and its conference CoNLL and peak body SIGNLL are sponsored by ACL, recognizing also their links with Computational Linguistics and Language Acquisition. When the aim of computational language learning research is to understand more about

human language acquisition, or psycholinguistics, NLL overlaps into the related field of Computational Psycholinguistics.

Major tasks in NLP

The following is a list of some of the most commonly researched tasks in NLP. Note that some of these tasks have direct real-world applications, while others more commonly serve as subtasks that are used to aid in solving larger tasks. What distinguishes these tasks from other potential and actual NLP tasks is not only the volume of research devoted to them but the fact that for each one there is typically a well-defined problem setting, a standard metric for evaluating the task, standard corpora on which the task can be evaluated, and competitions devoted to the specific task.

Automatic summarization

Produce a readable summary of a chunk of text. Often used to provide summaries of text of a known type, such as articles in the financial section of a newspaper.

Coreference resolution

Given a sentence or larger chunk of text, determine which words ("mentions") refer to the same objects ("entities"). Anaphora resolution is a specific example of this task, and is specifically concerned with matching up pronouns with the nouns or names that they refer to. The more general task of coreference resolution also includes identifying so-called "bridging relationships" involving referring expressions. For example, in a sentence such as "He entered John's house through the front door", "the front door" is a referring expression and the bridging relationship to be identified is the fact that the door being referred to is the front door of John's house (rather than of some other structure that might also be referred to).

Discourse analysis

This rubric includes a number of related tasks. One task is identifying the discourse structure of connected text, i.e. the nature of the discourse relationships between sentences (e.g. elaboration, explanation, contrast). Another possible task is recognizing and classifying the speech acts in a chunk of text (e.g. yes-no question, content question, statement, assertion, etc.).

Machine translation

Automatically translate text from one human language to another. This is one of the most difficult problems, and is a member of a class of problems colloquially termed "AI-complete", i.e. requiring all of the different types of knowledge that humans possess (grammar, semantics, facts about the real world, etc.) in order to solve properly.

Morphological segmentation

Separate words into individual morphemes and identify the class of the morphemes. The difficulty of this task depends greatly on the complexity of the morphology (i.e. the considered. English has fairly simple morphology structure of words) of the language being, especially inflectional morphology, and thus it is often possible to ignore this task entirely and simply model all possible forms of a word (e.g. "open, opens, opened, opening") as separate words. In languages such as Turkish or Manipuri,[4] a highly agglutinated Indian language, however, such an approach is not possible, as each dictionary entry has thousands of possible word forms.

II. RELATED WORK:

[1] Jalaj S. Modha, Prof & Head Gayatri S. Pandi Sandip J. Modha, **Automatic Sentiment Analysis for Unstructured Data**, International Journal of Advanced Research in Computer Science and Software Engineering , Volume 3, Issue 12, December 201,

In this thesis they discussed about exiting methods, approaches to do sentimental analysis for unstructured data which reside on web. Currently, Sentiment Analysis concentrates for subjective statements or on subjectivity and overlook objective statements which carry sentiment(s). So, they proposed new approach classify and handle subjective as well as objective statements for sentimental analysis.

Proposed Approach:

In Sentiment Analysis, numbers of sentences or sentences of documents. All these documents or sentences may convey opinion or maybe not. Formally, there is document set D= {d1, d2, ..., dN}, sentence set S= {S1, S2, ..., Sn} and all these documents and sentences belong to some specific entity e where e is a product, service, topic, issue, person, organization, or event

They followed four steps of classification.

1.) First step: First classify sentences or sentences of documents into two categories Opinionated and No- Opinionated, regardless whether it is subjective or objective.

2.) Second Step: In this step we have opinionated sentences so now they are classified as subjective sentences and Objective sentences.

3.) Third Step: The third step is classifying subjective sentences into positive, negative or neutral category. For complex type of sentences we may need to attach context or semantic orientation

4.) Fourth Step: The fourth step is classifying objective sentences into positive, negative or neutral category. Here also we have to provide context or sentiment orientation as and when needed.

[2] R M. Chandrasekaran, G.Vinodhini, **Sentiment Analysis and Opinion Mining: A Survey** International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 6, June 2012,

Sentiment Analysis for objective sentences is very trending research topic now-a-days because there are so many data sources which have objective sentences that carry sentiment but because of lake of proper algorithms and contexts we can't get the fruitful result from the objective sentences. According to recent article published by Ronen Feldman express that objective sentences that carry sentiment should be analyzed for getting efficient sentiment analysis and this is one of the challenging task in sentiment analysis.

Source of objective sentences are including news articles, blogs, social media etc. where we get good amount of objective sentences.

We consider following examples which are objective sentences but still carry sentiment.

- "Firefox keeps crashing." defined sentences carry negative sentiment about Firefox web browser.
- "The earphone broke in two days." defined sentence carry negative sentiment about the earphones.
- "I get relaxed time after today's session." define positive sentiment about person's routine.

In this particular area just challenges are proposed but still researchers are trying to find out efficient solution to get analyzed these kinds of implicit opinions in the objective sentences. Available sentiment dictionaries don"t have enough vocabulary to get analyzed objective sentences and categorized them efficiently into positive, negative or neutral. Provide proper context or semantic orientation is also very important part of sentiment analysis of objective Sentences.

[3] Bing Liu. **Sentiment Analysis and Opinion Mining,** Morgan & Claypool Publishers, May 2012,

Opinions and its related concepts such as sentiments, evaluations, attitudes, and emotions are the subjects of study of sentiment analysis and opinion mining. The inception and rapid growth of the field coincide with those of the social media on the Web, e.g., reviews, forum discussions, blogs, microblogs,

Twitter, and social networks, because for the first time in human history, we have a huge volume of opinionated data recorded in digital forms. Since early 2000, sentiment analysis has grown to be one of the most active research areas in natural language processing. It is also widely studied

in data mining, Web mining, and text mining. In fact, it has spread from computer science to management sciences and social sciences due to its importance to business and society as a whole. In recent years, industrial activities surrounding sentiment analysis have also thrived. Numerous startups have emerged. Many large corporations have built their own in-house capabilities. Sentiment analysis systems have found their applications in almost every business and social domain.

The goal of this book is to give an in-depth introduction to this fascinating problem and to present a comprehensive survey of all important research topics and the latest developments in the field. As evidence of that, this book covers more than 400 references from all major conferences and journals. Although the field deals with the natural language text, which is often

Considered the unstructured data, this book takes a structured approach in introducing the problem with the aim of bridging the unstructured and structured worlds and facilitating qualitative and quantitative analysis of opinions. This is crucial for practical applications. In this book, defined the problem in order to provide an abstraction or structure to the problem.

[4] Arti Buche, Dr. M. B. Chandak, Akshay Zadgaonkar, **OPINION MINING AND ANALYSIS: A SURVEY,** International Journal on Natural Language Computing (IJNLC) Vol. 2, No.3, June 2013

The current research is focusing on the area of Opinion Mining also called as sentiment analysis due to sheer volume of opinion rich web resources such as discussion forums, review sites and blogs are available in digital form. One important problem in sentiment analysis of product reviews is to produce summary of opinions based on product features. We have surveyed and analyzed in this thesis, various techniques that have been developed for the key tasks of opinion mining. They have provided an overall picture of what is involved in developing a software system for opinion mining on the basis of our survey and analysis.
Classifying entire documents according to the opinions towards certain objects is called as sentiment classification. One form of opinion mining in product reviews is also to produce feature-based summary. To produce a summary on the features, product features are first identified, and positive and negative opinions on them are aggregated. Features are product attributes, components and other aspects of the product. The effective opinion summary, grouping feature expressions which are domain synonyms is critical. It is very time consuming and tedious for human users to group typically hundreds of feature expressions that can be discovered from text for an opinion mining application into feature categories. Some automated assistance is needed. Opinion summarization does not

summarize the reviews by selecting a subset or rewrite some of the original sentences from the reviews to capture the main points as the classic text summarization.

[5] Fred Popowich**, Using Text Mining and Natural Language Processing for Health Care Claims Processing**, SIGKDD Explorations. Volume 7, Issue 1 - Page 59

The application makes use of a natural language processing (NLP) engine, together with application-specific knowledge, written in a concept specification language. Using NLP techniques, the entities and relationships that act as indicators of recoverable claims are mined from management notes, call centre logs and patient records to identify medical claims that require further investigation. Text mining techniques can then be applied to find dependencies between different entities, and to combine indicators to provide scores to individual claims. Claims are scored to determine whether they involve potential fraud or abuse, or to determine whether claims should be paid by or in conjunction with other insurers or organizations. Dependencies between claims and other records can then be combined to create cases. Issues related to the design of the application are discussed, specifically the use of rule-based techniques which provide a capability for deeper analysis than traditionally found in statistical techniques.

III. PROPOSED METHODLOLGY:

The proposed architecture of four modules: user interface, log pre-processing, Feature Clustering using Modified K-means, Naïve Bayes Classification, Training and testing using KNN for more accurate categorization of opinion. This system can solve irrelevant data and more accuracy by associating Modified K means with Naïve Bayes Classification algorithm.

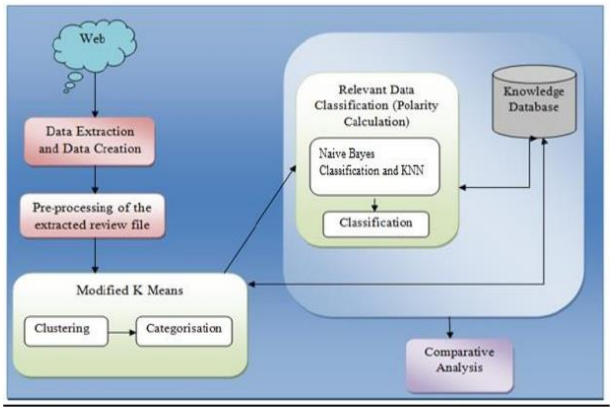


Figure 2: Proposed System Architecture

A. Naive Bayes (NB): Naive Bayes Classifier uses Bayes Theorem, which finds the probability of an event given the probability of another event that has already occurred. Naive Bayes classifier performs extremely well for problems which are linearly separable and even for problems which are non-linearly separable it performs reasonably well [3]. We used the already implemented Naive Bayes implementation in Weka2 toolkit.

Algorithm

S1: Initialize P(positive) ¬num - popozitii (positive)/ num_total_propozitii

S2: Initialize P(negative) ¬num - popozitii (negative) / num_total_propozitii

S3: Convert sentences into words

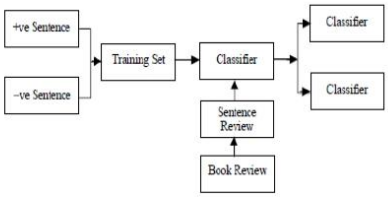
for each class of {positive, negative}:

for each word in {phrase}

P(word | class) < num_apartii (word | class) 1 | num_cuv (class) + num_total_cuvinte

P (class) ¬P (class) * P (word | class)

Returns max {P(pos), P(neg)}[1]



Naïve bayes classification

Major advantages of Naïve Bayes Classification is easy to interpret and efficient computation

Modified approach K-mean algorithm:

The K-mean algorithm is a popular clustering algorithm and has its application in data mining, image segmentation, bioinformatics and many other fields. This algorithm works well with small datasets. In this paper we proposed an algorithm that works well with large datasets. Modified k-mean algorithm avoids getting into locally optimal solution in some degree, and reduces the adoption of cluster -error criterion.

Algorithm: Modified approach (S, k), S={x1,x2,…,xn }

Input: The number of clusters k1(k1> k) and a dataset containing n objects(Xij+).

Output: A set of k clusters (Cij) that minimize the Cluster - error criterion.

Algorithm

1. Compute the distance between each data point and all other data- points in the set D

2. Find the closest pair of data points from the set D and form a data-point set Am (1<= p <= k+1) which contains these two data- points, Delete these two data points from the set D

3. Find the data point in D that is closest to the data point set Ap, Add it to Ap and delete it from D

4. Repeat step 4 until the number of data points in Am reaches (n/k)

5. If p<k+1, then p = p+1, find another pair of data points from D between which the distance is the shortest, form another data-point set Ap and delete them from D, Go to step 4.

IV. RESULT ANALYSIS / IMPLEMENTATION:

Name of Algorithm	Dataset	Accuracy(%)
Naive Bayes	500 mobile dataset	79.66
KNN	500 mobile dataset	83.59
Modified K-Means +NB	500 mobile dataset	89
Modified K-Means + NB + KNN	500 mobile dataset	91

Comparison Table 4.1

V. CONCLUSION:

Above methods has been applied on mobile review .We proposed a method using Naïve Bayes, KNN and modified k means clustering and found that it is more accurate than Naïve Bayes and KNN techniques individually. We obtained an overall classification accuracy of 91% on the test set of 500 mobile reviews. The running time of our algorithm is O (n + V log V) for training and O (n) for testing, where n is the number of words in the documents (linear) and V the size of the reduced vocabulary. It is much faster than other machine learning algorithms like Naïve Bayes classification or Support Vector Machines which take a long time to converge to the optimal set of weights. The accuracy is comparable to that of the current state-of-the-art algorithms used for sentiment classification on mobile reviews.

From our point of view MKM, Naïve Bayes and KNN is best suitable for text based classification and social interpretation. In future we will be finding out the best result of sentiment analysis by applying other method on social networking reviews.

REFERENCES

[1] G.Vinodhini and RM.Chandrasekaran, **"Sentiment Analysis and Opinion Mining: A Survey",** Volume 2, Issue 6, June 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

[2] Zhongwu Zhai, Bing Liu, Hua Xu and Hua Xu, **"Clustering Product Features for Opinion Mining"**, WSDM'11, February 9–12, 2011, Hong Kong, China. Copyright 2011 ACM 978-1-4503-0493- 1/11/02...$10.00

[3] Singh and Vivek Kumar, "**A clustering and opinion mining approach to socio-political analysis of the blogosphere"**, Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference.

[4] Alexander Pak and Patrick Paroubek, **"Twitter as a Corpus for Sentiment Analysis and Opinion Mining"**

[5] Bing Liu. **"Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012."**

[6] V. S. Jagtap and Karishma Pawar, **"Analysis of different approaches to Sentence-Level Sentiment Classification",** International Journal of Scientific Engineering and Technology (ISSN: 2277-1581) Volume 2 Issue 3, PP: 164-170 1 April 2013

[7]. K. Bun and M. Ishizuka, **"Topic extraction from news archive using TF*PDF algorithm",** In Proceedings of Third International Conference on Web Information System Engineering.

[8]. Jacques Savoy, Olena Zubaryeva, **"Classification Based on Specific Vocabulary"** published in 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology 978-0-7695-4513-4/11 2011 IEEE

[9]. Dengya Zhu, Jitian XIAO, **"R-tfidf, a Variety of tf-idf Term Weighting Strategy in Document Categorization",** published in 2011 Seventh International Conference on Semantics, Knowledge and Grids.

[10]. Catherine Blake **"A Comparison of Document, Sentence, and Term Event Spaces"** published in IEEE 2010

[11]. Ying Chen, Wenping Guo, Xiaoming Zhao, **"A semantic Based Information Retrieval Model for Blog",** Third International Symposium on Electronic Commerce and Security, 2010, IEEE

[12]. Mukhrjee, A. and B. Liu, **"Improving gender classification of weblog authors. Proceedings of Conference on Empirical Methods in Natural Language Processing",** (EMNLP' 10), 10RDF Primer. W3C Recommendation. http://www.w3.org/TR/rdf-primer, 2004.

[13] Jalaj S. Modha, Prof & Head Gayatri S. Pandi Sandip J. Modha, **"Automatic Sentiment Analysis for Unstructured Data",** International Journal of Advanced Research in Computer Science and Software Engineering , Volume 3, Issue 12, December 2013

[14] R M. Chandrasekaran, G.Vinodhini, **"Sentiment Analysis and Opinion Mining: A Survey",** International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 6, June 2012

[15] Bing Liu., **"Sentiment Analysis and Opinion Mining",** Morgan & Claypool Publishers, May 2012.

REFERENCES

[1] G.Vinodhini and RM.Chandrasekaran, “Sentiment Analysis and Opinion Mining: A Survey”, Volume 2, Issue 6, June 2012 ISSN: 2277 128X International Journal of Advanced Research in Computer Science and Software Engineering

[2] Zhongwu Zhai, Bing Liu, Hua Xu and Hua Xu, “Clustering Product Features for Opinion Mining”, WSDM’11, February 9–12, 2011, Hong Kong, China. Copyright 2011 ACM 978-1-4503- 0493- 1/11/02...$10.00

[3] Singh and Vivek Kumar, ―A clustering and opinion mining approach to socio-political analysis of the blogosphere”, Computational Intelligence and Computing Research (ICCIC), 2010 IEEE International Conference.

[4] Alexander Pak and Patrick Paroubek, “Twitter as a Corpus for Sentiment Analysis and Opinion Mining”

[5] Bing Liu. “Sentiment Analysis and Opinion Mining, Morgan & Claypool Publishers, May 2012.”

[6] V. S. Jagtap and Karishma Pawar, “Analysis of different approaches to Sentence-Level Sentiment Classification”, International Journal of Scientific Engineering and Technology (ISSN: 2277-1581) Volume 2 Issue 3, PP: 164-170 1 April 2013 Onam Bharti et al, International Journal of Computer Science and Mobile Computing, Vol.5 Issue.6, June- 2016, pg. 601-609 © 2016, IJCSMC All Rights Reserved 609

[7]. K. Bun and M. Ishizuka, “Topic extraction from news archive using TF*PDF algorithm”, In Proceedings of Third International Conference on Web Information System Engineering.

[8]. Jacques Savoy, Olena Zubaryeva, “Classification Based on Specific Vocabulary” published in 2011 IEEE/WIC/ACM International Conferences on Web Intelligence and Intelligent Agent Technology 978-0-7695-4513-4/11 2011 IEEE

[9]. Dengya Zhu, Jitian XIAO, “R-tfidf, a Variety of tf-idf Term Weighting Strategy in Document Categorization”, published in 2011 Seventh International Conference on Semantics, Knowledge and Grids.

[10]. Catherine Blake “A Comparison of Document, Sentence, and Term Event Spaces” published in IEEE 2010

[11]. Ying Chen, Wenping Guo, Xiaoming Zhao, “A semantic Based Information Retrieval Model for Blog”, Third International Symposium on Electronic Commerce and Security, 2010, IEEE

[12]. Mukhrjee, A. and B. Liu, “Improving gender classification of weblog authors. Proceedings of Conference on Empirical Methods in Natural Language Processing”, (EMNLP’ 10), 10RDF Primer. W3C Recommendation. http://www.w3.org/TR/rdf-primer, 2004.

[13] Jalaj S. Modha, Prof & Head Gayatri S. Pandi Sandip J. Modha, “Automatic Sentiment Analysis for Unstructured Data”, International Journal of Advanced Research in Computer Science and Software Engineering , Volume 3, Issue 12, December 2013 [14] R M. Chandrasekaran, G.Vinodhini, “Sentiment Analysis and Opinion Mining: A Survey”, International Journal of Advanced Research in Computer Science and Software Engineering, Volume 2, Issue 6, June 2012 [15] Bing Liu., “Sentiment Analysis and Opinion Mining”, Morgan & Claypool Publishers, May 2012