

A JOINT OPTIMIZATION APPROACH TO SECURITY AND INSURANCE MANAGEMENT SYSTEM ON CLOUD

Submitted in partial fulfillment of the requirements for the award of
Master of Engineering Degree in
Computer Science & Engineering

by

Setty Hareesh (Reg. No. 37110696)

Rakesh kumar(Reg. No. 37110682)



**DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
FACULTY OF COMPUTING
SATHYABAMA UNIVERSITY
JEPPIAAR NAGAR, RAJIV GANDHI SALAI,
CHENNAI – 600 119. TAMILNADU**

MAY 2021



SATHYABAMA UNIVERSITY
(Established under Section 3 of UGC Act, 1956)
Jeppiaar Nagar, Rajiv Gandhi Salai,
Chennai – 600119
www.sathyabamauniversity.ac.in



DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING

BONAFIDE CERTIFICATE

This is to certify that this project report is the bonafide work of **Setty Hareesh (Reg. No. 37110696) Rakesh kumar(Reg. No. 37110682)** who carried out the project entitled **“A JOINT OPTIMIZATION APPROACH TO SECURITY AND INSURANCE MANAGEMENT SYSTEM ON CLOUD”** under our supervision from November 2020 to April 2021.

Internal Guide

Dr. Joshila Grace L.K

Head of the Department

Dr. LAKSHMANAN

Dr. VIGNESHWARI

Submitted to Project and Viva Examination held on _____

INTERNAL EXAMINAR

EXTERNAL EXAMINAR

DECLARATION

I **Setty Hareesh (Reg. No. 37110696) Rakesh kumar(Reg. No. 37110682)**

hereby declare that the Project Report entitled **“A JOINT OPTIMIZATION APPROACH TO SECURITY AND INSURANCE MANAGEMENT SYSTEM ON CLOUD”** done by me under the guidance of Mrs. Joshila Grace M.E., is submitted in partial fulfillment of the requirements for the award of MASTER OF ENGINEERING degree in Computer Science & Engineering

DATE :

PLACE :

SIGNATURE OF THE CANDIDATE

ACKNOWLEDGEMENT

First I am pleased to acknowledge my sincere and heartfelt thanks to my beloved Founder Chancellor **Col. Dr. JEPPIAAR, M.A., B.L., Ph.D.**, and Chancellor **Tmt. REMIBAI JEPPIAAR** for their kind encouragement in doing this project and for completing it successfully. I am grateful to him. I express my sincere thanks to our respectable Vice President **Dr. MARIE JOHNSON, B.E., M.B.A., M.Phil., Ph.D.**, and Pro Chancellor **Dr. MARIAZEENA JOHNSON, B.E., M.B.A., M.Phil., Ph.D.**, for all the help and infrastructure to complete this project and for providing me the necessary facilities for the completion of my project.

I express my deep sense of gratitude and acknowledge my thanks to our vice chancellor **Dr. T. SASIPRABA Ph.D.**, for their constant support and endorsement.

I am indebted to the Registrar **Dr. S.S. RAU, Ph. D.**, for his support to the completion of my project. At this juncture I feel deeply honored in expressing my sincere thanks to Controller of Examinations **Prof. IGNI SABASTI PRABU M.E., Ph.D.**, for his support to complete my project successfully.

I express my sincere thanks to Dean, School of Computing **Dr. SASIKALA, M.E., Ph.D.**, and Faculty Head **Dr. LAKSHMANAN M.E., Ph. D, Dr. VIGNESHWARI, M.E., Ph. D.**, for thier support. I extend my sincere and deep sense of gratitude to my Project Guide **Dr. JOSHILA GRACE, M.E., Ph. D**, for her valuable guidance, suggestions and constant encouragement paved way for the successful completion of my project work.

My gratitude is extended to the department staff members, technical staffs, for prompt help when required and to all my friends who are directly and indirectly involved in the completion of my project

ABSTRACT

Millions of users share opinions on different aspects of life everyday in popular Social Network Sites (SNS) such as Twitter and Facebook. Recently, social media has become important for social networking and content sharing. Twitter, an online social network, allows users to upload short text messages, also known as tweets, with up to 140 characters. A lot of people use sentiment analysis on Twitter to do opinion mining. People choose Twitter because it serves as a good platform for sentiment analysis because of its large user base from different socio cultural zones. The objective of Sentiment Analysis is to identify any clue of positive or negative emotions in a piece of text reflective of the authors' opinions on a subject. Spurred by this growth, companies and media organization are increasingly seeking ways to mine these social media for information about what people think about their companies and products. Social organizations may ask people's opinion on current debates. All this information can be obtained from microblogging services, as their users post their opinions on many aspects of their life regularly. Sentiment analysis is concerned with identifying and grouping assumptions articulated in a piece of text to identify whether the user's view towards a particular subject, merchandise, etc. is optimistic, pessimistic, or impartial. Opinions about products influence buying behavior among users since millions of them visit micro blogging websites. In this project, an assortative model for product reviews is developed to analyze and to categorize sentiments based on the polarization.

TABLE OF CONTENTS

CHAPTER NO	TITLE	PAGE NO
	ABSTRACT	iv
	LIST OF FIGURES	vi
	LIST OF TABLES	vii
	LIST OF ABBREVIATIONS	viii
1	INTRODUCTION	1
	1.1 OVERVIEW OF SOCIAL NETWORKING SITES	1
	1.2 PROBLEM DEFINITION	2
	1.3 LITERATURE SURVEY	3
	1.4 COMPARITIVE ANALYSIS	8
2	ANALYSIS AND DESIN	9
	2.1 SYSTEM ANALYSIS	9
	2.1.1 EXISTING SYSTEM	9
	2.1.2 PROPOSED SYSTEM	10
	2.2.1 SOFTWARE REQUIREMENTS	11
	2.2.2 HARDWARE REQUIREMENTS	11
	2.2.3 SOFTWARE DESCRIPTION	11
	2.3 DESIGN	17
	2.3.1UML DIAGRAMS	17
	2.3.1.1 USE CASE DIAGRAM	17
	2.3.2.2 CLASS DIAGRAM	18
	2.3.1.3 ACTIVITY DIAGRAM	19

	2.3.1.4 SEQUENCE DIAGRAM	20
	2.3.1.5 COLLABORATION DIAGRAM	21
	2.3.1.6 COMPONENT DIAGRAM	21
	2.3.1.7 DATA FLOW DIAGRAM	22
	2.3.1.8 FLOWCHART DIAGRAM	23
	2.4 DATA BASE DESIGN	24
3	SYSTEM DESIGN	25
	3.1 DEVELOPMENT METHODOLOGY	25
	3.2 SYSTEM ARCHITECTURE	25
	3.3 MODULE DESCRIPTION	26
	3.4 ALGORITHM IMPLEMENTATION	29
	3.5 EVALUATION METRICS	31
4	RESULTS AND DISCUSSION	32
5	CONCLUSION AND FUTURE WORK	35
	REFERENCES	36
	APPENDICES	
	A.1 SCREEN SHOTS	38
	A.2 CODING	44
	PUBLICATIONS	52

LIST OF FIGURES

FIGURE NO	FIGURE NAME	PAGE NO
1.1	Sample tweet data from SNS - Twitter	3
2.1	Use Case Diagram	18
2.2	Class Diagram	19
2.3	Activity Diagram	20
2.4	Sequence Diagram	21
2.5	Collaboration Diagram	22
2.6	Component Diagram	22
2.7	Data Flow Diagram	23
2.8	Flow Chart	24
3.1	System Architecture Of Sentimental Analysis	26
3.2	Process Flow For Preprocessing Tweet	28

LIST OF TABLES

TABLE NO.	TABLES	PAGE NO.
1.1	Summary of Literature survey	8
2.2	Table Design	24
3.1	List of tokens generated on a tweet	27
3.2	List of tokens generated after stop words removal	28
3.3	List of tokens generated after Stemming	29
3.4	List of tokens generated after Case Transformation	29
4.1	Part-of-Speech tags for verbs	33

LIST OF ABBREVIATIONS

ACRONYM	EXPANSION
SNS	SOCIAL NETWORKING SITES
SWB	SOCIAL WELL- BEING
SVM	SUPPORT VECTOR MACHINE
NLTK	NATURAL LANGUAGE TOOLKIT
JSON	JAVA SCRIP OBJECT NOTATION

CHAPTER 1

INTRODUCTION

1.1 OVERVIEW OF SOCIAL NETWORKING SITE (SNS) - TWITTER

These days the textual data on the Internet is growing at a rapid pace and many companies are trying to use this deluge of data to extract people's views towards their products. Online social network platforms, with their large-scale repositories of user-generated content, can provide unique opportunities to gain insights into the emotional "pulse of the nation", and indeed the global community. A great source of unstructured text information is included in social networks, where it is unfeasible to manually analyze such amounts of data. There is a large number of social networks websites that enable users to contribute, modify and grade the content, as well as to express their personal opinions about specific topics. Some examples include blogs, forums, product reviews sites, and social networks, like Twitter (<http://twitter.com/>). Twitter (San Francisco, CA, USA) is a microblogging site that offers the opportunity for the analysis of expressed mood, and previous studies have shown that geographical, diurnal, weekly, and seasonal patterns of positive and negative affect can be observed.

Microblogging and more particularly Twitter is used for the following reasons:

- Microblogging platforms are used by different people to express their opinion about different topics, thus it is a valuable source of people's opinions.
- Twitter contains an enormous number of text posts and it grows every day. The collected corpus can be arbitrarily large.
- Twitter's audience varies from regular users to celebrities, company representatives, politicians, and even country presidents. Therefore, it is possible to collect text posts of users from different social and interests groups.
- Twitter's audience is represented by users from many countries

As the audience of microblogging platforms and services grows every day, data from these sources can be used in opinion mining and sentiment analysis tasks. For example, manufacturing companies may be interested in the following questions:

- What do people think about our product (service, company etc.)?
- How positive (or negative) are people about our product?
- What would people prefer our product to be like?

Political parties may be interested to know if people support their program or not. Social organizations may ask people's opinion on current debates. All this information can be obtained from microblogging services, as their users post everyday what they like/dislike, and their opinions on many aspects of their life.

1.2 PROBLEM DEFINITION

It is useful to begin with an initial introduction to our domain and the problems addressed in this project. Twitter is a microblogging service allowing users to post up to 140 characters at a time describing their thoughts. Users are able to "follow" other users, keeping up to date with information that user has to offer. In recent years Twitter has surged in popularity, providing a huge source of untapped, unstructured data. Users have also used it for a wide variety of purposes.

Tweet - a microblogging post up to 140 characters in length.

@username - Tweets may be addressed to a specific user by placing their username followed by the "@" symbol somewhere in the tweet. These tweets remain public, but are used to direct tweets to another user, to thank another user, or to redirect traffic. Multiple users may be addressed in a single tweet. e.g. @chennaiusrs

#hashtag - Tweets can also be labeled as pertaining to a specific topic by using "hashtags": one-word labels following a "#" symbol. Multiple hashtags can be included

in a single tweet. Twitter users can search based on hashtags to quickly find relevant content. e.g. #AUEXAMS

RT - A retweet, labeled by the abbreviation "RT", is a Tweet that is a repeat of another user's tweet. The purpose of a retweet is to take a tweet and redistribute it on one's own feed in the hopes of broadcasting the message to a larger audience or demonstrating support of the tweet's contents.

Twitter contains a very large number of very short messages created by the users of this microblogging platform. The contents of the messages vary from personal thoughts to public statements. Figure 1.1 shows examples of typical posts from Twitter.

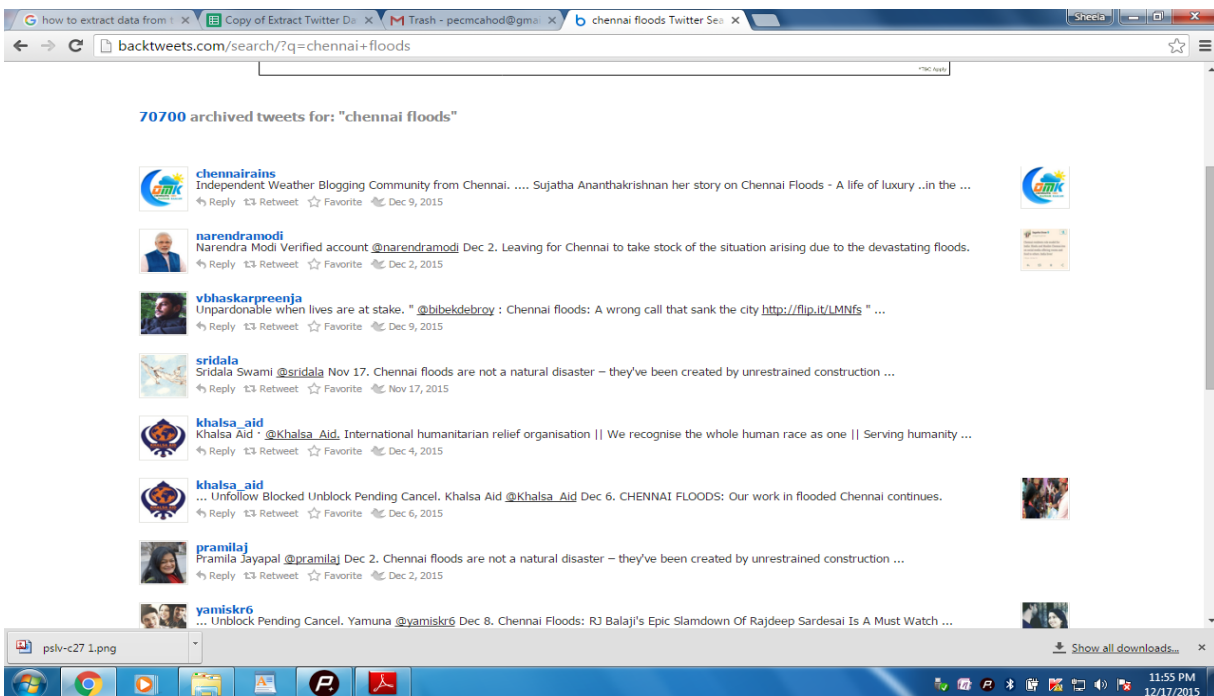


Figure 1.1 Sample tweet data from SNS - Twitter

The project focuses on using Twitter, the most popular microblogging platform, for the task of sentiment analysis. The tweets are important for analysis because data arrive at a high frequency and algorithms that process them must do so under very strict constraints of storage and time. It will be shown how to automatically collect a corpus

for sentiment analysis and opinion mining purposes and then perform linguistic analysis of the collected corpus. All public tweets posted on twitter are freely available through a set of APIs provided by Twitter. Using the corpus, a sentiment classifier, is constructed that is able to determine positive, negative and neutral sentiments.

1.3 LITERATURE SURVEY

1.3.1 Xing Fang, Justin Zhan Sentiment analysis using product review data In: Journal of Big Data (2015) 2: 5. doi:10.1186/s40537-015-0015-2

In this paper, the authors aim to tackle the problem of sentiment polarity categorization, which is one of the fundamental problems of sentiment analysis. A general process for sentiment polarity categorization is proposed with detailed process descriptions. Data used in this study are online product reviews collected from Amazon.com. Experiments for both sentence-level categorization and review-level categorization are performed with promising outcomes.

This paper tackles a fundamental problem of sentiment analysis, namely sentiment polarity categorization. The authors contribution can be summarized as : 1) An algorithm for negation phrases identification; 2) A mathematical approach is proposed for sentiment score computation; 3) A feature vector generation method is presented for sentiment polarity categorization. 4) Two sentiment polarity categorization experiments are respectively performed based on sentence level and review level 5) Performance of three classification models are evaluated and compared based on their experimental results.

1.3.2 Bian, Jiang, Umit Topaloglu, and Fan Yu. "Towards large-scale twitter mining for drug-related adverse events.". In Proceedings of the 2012 international workshop on Smart health and wellbeing, pp. 25-32. ACM, 2012. [4]

In this paper, the authors describe an approach to find drug users and potential adverse events by analyzing the content of twitter messages utilizing Natural Language Processing (NLP) and to build Support Vector Machine (SVM) classifiers. Due to the size nature of the dataset (i.e., 2 billion Tweets), the experiments were conducted on a

High Performance Computing (HPC) platform using MapReduce, which exhibits the trend of big data analytics. The results suggest that daily-life social networking data could help early detection of important patient safety issues

The data set used is a collection of over 2 billion Tweets collected from May 2009 to October 2010, from which they try to identify potential adverse events caused by drugs of interest. The collected stream of Tweets was organized by a timeline. The raw Twitter messages were crawled using the Twitter's user timeline API that contains information about the specific Tweet and the user. The work is indexed only with the following four fields for each Tweet:

- 1) the Tweet id that uniquely identifies each Tweet;
- 2) the user identifier associated with each Tweet;
- 3) the timestamp of the Tweet; and 4) the Tweet text.

They utilized the Amazon Elastic Compute Cloud (EC2) to run the Twitter indexers on 15 separate EC2 instances, 34.2 GB of memory, and 13 EC2 Compute Units) in parallel, which were able to parse and index all 2 billion Tweets within two days. The size of the Lucene indexes is 896 GB.

To mine Twitter messages for AEs, the process can be separated into two parts:

- 1) identifying potential users of the drug;
- 2) finding possible side effects mentioned in the users' Twitter timeline that might be caused by the use of the drug concerned.

Both processes involve building and training classification models based on features extracted from the users' Twitter messages. Two-sets of features (i.e., textual and semantic features) are extracted from Twitter users' timeline for both classification models.

Evaluation of the SVM was done using parameters such as , the Area Under the Curve (AUC) value, and the Receiver operating characteristic (ROC) curve. The ROC

curve using the mean values of the 1000 iterations was drawn. The prediction accuracy on average over the 1000 iterations was evaluated to 0.74 and the mean AUC value is 0.82.

1.3.3 Hu M, Liu B (2004) Mining and summarizing customer reviews In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 168–177.. ACM, New York, NY, USA

The authors focus on three important aspects of Sentimental Analysis (1) mining product features that have been commented on by customers (2) identifying opinion sentences in each review and deciding whether each opinion sentence is positive or negative (3) summarizing the results. These aspects can be achieved by Subjective Genre Classification, Sentiment Classification, Text Summarization, Frequent Features Identification, Orientation Identification for Opinion and Words Infrequent Feature Identification

The authors conducted experiments on electronics data sets retrieved from twitter and the average of precision and recall is 0.693 0.642 0.842.

1.3.4 ÁlvaroCuesta, David F., and María D. R-Moreno. "A Framework For Massive Twitter Data Extraction And Analysis." In Malaysian Journal of Computer Science, pp 50-67 (2014):1. [6]

The authors propose an open framework to automatically collect and analyze data from Twitter's public streams. This is a customizable and extensible framework, so researchers can use it to test new techniques. The framework is complemented with a language-agnostic sentiment analysis module, which provides a set of tools to perform sentiment analysis of the collected tweets.

The capabilities of this platform are illustrated with two study cases in Spanish, one related to a high impact event (the Boston Terror Attack), and another one related to regular political activity on Twitter. The first case study involves the activity on Twitter around a high impact event, the Boston Terror Attacks. In this case, they tracked a

hashtag. The second case study was focused on regular Twitter usage, tracking the activity around well-known Spanish political actors, i.e. politicians, political parties, journalists and activist organizations as well. The authors have selected controversial accounts to have a good foundation for sentiment analysis.

There are several layers of processing and these modules need to interchange data among them, using open data formats such as JSON. Most tools in the framework are implemented in Python, but the Classifier and Tester web interfaces run on NodeJS and are programmed in CoffeeScript (a language which can be pre-processed into JavaScript). The chosen backend database is MongoDB, which is a good fit for our purposes since its atomic representation is JSON, just like tweets. The implementation was based on the Natural Language Toolkit (NLTK) framework.

A complete procedure of data extraction and sentiment analysis is divided into three separate steps: data acquisition, training for sentiment analysis and report generation. The first step is, gathering data from Twitter with the Miner. Then the classifier is trained and the sentiment analysis carried out. Finally, the platform generates a set of reports, including the sentiment analysis if it is enabled. Classification was done according to three classes, “positive”, “negative” and “neutral”. Several Naïve Bayes classifiers using a set of ngrams in order to select the one with the best performance. In particular, they have tried {1}, {2}, {3}, {1, 2}, {1, 3} and {2, 3} ngrams and minimum score of 0, 1, 2, 3, 4, 5, 6 and 10.

All these different options were tried using ten-fold cross-validation to avoid biases induced by the partition of the training set. The parameters such as accuracy mean and variance, precision, recall and fmeasure mean and variance were used for evaluation. The conclusion is that the best trainers had 1-grams included and a minimum score between 2 and 4.

1.3.5 Johan Bollen, Bruno Gonçalves, Guangchen Ruan, & Huina Mao Happiness is assortative in online social networks. *Artificial Life* volume 17 issue 3, 2011 237-251

The authors relate each individual with either similar or dissimilar characteristics in social networks. They define this propensity as assortative mixing or homophily and it is expressed as the correlation between attribute values of nearest neighbour vertices in a graph. They also focused on recent results that indicate that beyond demographic features such as age, sex and race, even psychological states such as “loneliness” can be assortative in a social network. They stress the importance of general happiness or Subjective Well-Being (SWB) of Twitter users measured from a 6 month record of their individual tweets, is indeed assortative across the Twitter social network. Their implementation shows that assortative mixing in online networks at the level of SWB and their results also imply that online social networks may be equally subject to the social mechanisms that cause assortative mixing in real social networks and that such assortative mixing takes place at the level of SWB.

1.3.6 Pennacchiotti, M., and Popescu, A. 2011. A machine learning approach to twitter user classification. In Proceedings of the International Conference on Weblogs and Social Media."

This paper addresses the task of user classification in social media, with an application to Twitter. The authors automatically infer the values of user attributes such as political orientation or ethnicity by leveraging observable information such as the user behavior, network structure and the linguistic content of the user's Twitter feed. The authors employ a machine learning approach which relies on a comprehensive set of features derived from such user information and report an encouraging experimental results on 3 tasks with different characteristics: political affiliation detection, ethnicity identification and detecting affinity for a particular business. Finally, their analysis shows that rich linguistic features prove consistently valuable across the 3 tasks and show great promise for additional user classification needs.

1.4 COMPARITIVE ANALYSIS

Table 1.1 presents an analysis of the various approaches studied in the literature survey.

Table 1.1 Summary of Literature survey

S.NO	TITLE , AUTHOR,YEAR	METHODOLOGY	REMARKS
1.	Xing Fang, Justin Zhan Sentiment analysis using product review data (2015)	• Negation Phrase identification • Sentiment Score Computation • Sentiment Polarity Categorization	Polarity classification experiments showed accuracy in the range 77% to 82% with varying data set size
2.	Bian, Jiang, Umit Topaloglu, and Fan Yu. Towards large-scale twitter mining for drug-related adverse events. (2012)	• Describes an approach to find drug users and potential adverse events by analyzing the content of twitter messages • Utilizes Natural Language Processing (NLP) to build Support Vector Machine (SVM) classifiers	The prediction accuracy on average over the 1000 iterations was evaluated to 0.74 and the mean AUC value is 0.82.
3.	Hu M, Liu B Mining and summarizing customer reviews (2004)	• Subjective Genre Classification • Frequent Features Identification	average of precision and recall is 0.693 0.642 0.842.
4.	ÁlvaroCuesta, David F., and María D. R-Moreno. "A Framework For Massive Twitter Data Extraction And Analysis (2014)	• Tracking the activity around well-known Spanish political actors • The framework is implemented in Python, but the Classifier and Tester web interfaces run on NodeJS	The conclusion is that the best trainers had 1-grams included and a minimum score between 2 and 4.
5.	Johan Bollen, Bruno Gonçalves, Guangchen Ruan, & Huina Mao Happiness is assortative in online social networks (2011)	• Friend Follower Network • Assortative characteristics also known as Social Wellbeing Characteristics	The probability distribution of Subjective Well-Being values across all Twitter users in the sample.

6.	Pennacchiotti, M., and Popescu, A. 2011. A machine learning approach to twitter user classification (2011)	• General model for user profiling • Linguistic Content	The Political affiliation, ethnicity is evaluated to identify the relationship among users
----	--	--	--

CHAPTER 2

DESIGN AND ANALYSIS

2.1 SYTEM ANALYSIS

2.1.1 EXISTING SYSTEM

Sentimental Analysis is all about to get the real voice of people towards specific product, services, organization, movies, news, events, issues and their attributes. Sentiment Analysis includes branches of Computer Science like Natural Language

Processing, Machine Learning, Text Mining and Information Theory and Coding. By using approaches, methods, techniques and models of defined branches, unstructured data can be categorized which may be in the form of news articles, blogs, tweets, movie reviews, product reviews etc. into positive, negative or neutral sentiment according to the sentiment expressed in them.

Sentiment Analysis is done on three levels .

- Document level
- Sentence level
- Aspect or Entity level

1) Document Level Sentiment Analysis is performed for the whole document and then decide whether the document express positive or negative sentiment .

2) Entity or Aspect Level Sentiment Analysis performs fine-grained analysis. The goal of entity or aspect level Sentiment Analysis is to find sentiment on entities and/or aspect of those entities. For example consider a statement “My HTC Wildfire S phone has good picture quality but it has low phone memory storage.” so sentiment on HTC’s camera and display quality is positive but the sentiment on its phone memory storage is negative.

3) Sentence level Sentiment Analysis is related to find sentiment form sentences whether each sentence expressed a positive, negative or neutral sentiment Sentence level Sentiment Analysis is closely related to subjectivity classification. Many of the statements about entities are factual in nature and yet they still carry sentiment.

Traditional Business Intelligence platforms can work with Relational data and can produce better results. However, in a platform to work with text files and cross-platform files, existing relational warehouses encounter problems in processing data. Applying existing data mining algorithms and techniques to real-world problems has been

recently running into many challenges due to the inadequate scalability . Not only the scale of data generated today is unprecedented, the produced data is often continuously made in the form of streams that require being processed and mined in (nearly) real time. Delayed discovery of even highly valuable knowledge invalidates the usefulness of the discovered knowledge.

2.1.2 PROPOSED SYSTEM

In this project, a systematic approach to analyze twitter data is proposed. It includes the following steps: 1) To identify the flaws in the twitter data that potentially deters the development of sentiment analysis. The primary fault is quality of content since users enjoy certain degree of freedom to liberally place their own views hence value of their opinions cannot be assured. Removal and identification of irrelevant content and spam becomes very important. 2) Social well being attributes (SWB) are highly influential in determining the polarity of the opinions. 3) A classifier to analyze the sentiments

Data from Twitter extracted using REST API, Search API and Data from Flipkart.com also retrieved to understand the similarity between how reviews are posted in twitter and online merchant websites.

WordNet version 2.1 is used as a sense warehouse and each word is mapped according Parts-of-Speech Tags and SentiWordNet assigns the sentiment scores. Finally, Weka tool is used to conduct experiments.

This project addresses a essential difficulty of sentiment analysis, called polarity categorization using assortative model properties. In this project three phase course of action is proposed. In phase 1, tweets are retrieved using APIs. In Phase 2: i) An algorithm is proposed and implemented to identify homophily among users using assortative model; ii) A mathematical approach is proposed to calculate similarity index; In Phase 3: i) Polarity categorization experiments based on SWB characteristic are

carried out; ii) Performance analysis of classification methods for assortative model and comparison of results.

2.2 HARDWARE AND SOFTWARE REQUIREMENTS

2.2.1 SOFTWARE REQUIREMENTS

- Programming Language : C#
- Database Server : SQL SERVER
- Operating System : WINDOWS 8.1

2.2.2 HARDWARE REQUIREMENTS

- Processor : Intel Pentium Dual Core
- Clock Speed : 2.5 GHz
- Ram capacity : 2 GB
- Hard Drive Capacity : 250 GB

2.2.3 SOFTWARE DESCRIPTION

2.2.3.1 MICROSOFT.NET FRAMEWORK- FEATURES OF .NET

Microsoft .NET is a set of Microsoft software technologies for rapidly building and integrating XML Web services, Microsoft Windows-based applications, and Web solutions. The .NET Framework is a language-neutral platform for writing programs that can easily and securely interoperate. There's no language barrier with .NET: there are numerous languages available to the developer including Managed C++, C#, Visual Basic and Java Script. The .NET framework provides the foundation for components to interact seamlessly, whether locally or remotely on different platforms. It standardizes common data types and communications protocols so that components created in different languages can easily interoperate.

“.NET” is also the collective name given to various software components built upon the .NET platform. These will be both products (Visual Studio.NET and

Windows.NET Server, for instance) and services (like Passport, .NET My Services, and so on).

2.2.3.2 THE .NET FRAMEWORK

The .NET Framework has two main parts:

1. The Common Language Runtime (CLR).
2. A hierarchical set of class libraries.

The CLR is described as the “execution engine” of .NET. It provides the environment within which programs run. The most important features are

- ◆ Conversion from a low-level assembler-style language, called Intermediate Language (IL), into code native to the platform being executed on.
- ◆ Memory management, notably including garbage collection.
- ◆ Checking and enforcing security restrictions on the running code.
- ◆ Loading and executing programs, with version control and other such features.
- ◆ The following features of the .NET framework are also worth description:

Managed Code

The code that targets .NET, and which contains certain extra Information - “metadata” - to describe itself. Whilst both managed and unmanaged code can run in the runtime, only managed code contains the information that allows the CLR to guarantee, for instance, safe execution and interoperability.

Managed Data

With Managed Code comes Managed Data. CLR provides memory allocation and Deal location facilities, and garbage collection. Some .NET languages use Managed Data by default, such as C#, Visual Basic.NET and JScript.NET, whereas others, namely C++, do not. Targeting CLR can, depending on the language you’re

using, impose certain constraints on the features available. As with managed and unmanaged code, one can have both managed and unmanaged data in .NET applications - data that doesn't get garbage collected but instead is looked after by unmanaged code.

Common Type System

The CLR uses something called the Common Type System (CTS) to strictly enforce type-safety. This ensures that all classes are compatible with each other, by describing types in a common way. CTS define how types work within the runtime, which enables types in one language to interoperate with types in another language, including cross-language exception handling. As well as ensuring that types are only used in appropriate ways, the runtime also ensures that code doesn't attempt to access memory that hasn't been allocated to it.

Common Language Specification

The CLR provides built-in support for language interoperability. To ensure that you can develop managed code that can be fully used by developers using any programming language, a set of language features and rules for using them called the Common Language Specification (CLS) has been defined. Components that follow these rules and expose only CLS features are considered CLS-compliant.

2.2.3.4 THE CLASS LIBRARY

.NET provides a single-rooted hierarchy of classes, containing over 7000 types. The root of the namespace is called System; this contains basic types like Byte, Double, Boolean, and String, as well as Object. All objects derive from System. Object. As well as objects, there are value types. Value types can be allocated on the stack, which can provide useful flexibility. There are also efficient means of converting value types to object types if and when necessary.

The set of classes is pretty comprehensive, providing collections, file, screen, and network I/O, threading, and so on, as well as XML and database connectivity.

The class library is subdivided into a number of sets (or namespaces), each providing distinct areas of functionality, with dependencies between the namespaces kept to a minimum.

2.2.3.5 LANGUAGES SUPPORTED BY .NET

The multi-language capability of the .NET Framework and Visual Studio .NET enables developers to use their existing programming skills to build all types of applications and XML Web services. The .NET framework supports new versions of Microsoft's old favorites Visual Basic and C++ (as VB.NET and Managed C++), but there are also a number of new additions to the family.

Visual Basic .NET has been updated to include many new and improved language features that make it a powerful object-oriented programming language. These features include inheritance, interfaces, and overloading, among others. Visual Basic also now supports structured exception handling, custom attributes and also supports multi-threading.

Visual Basic .NET is also CLS compliant, which means that any CLS-compliant language can use the classes, objects, and components you create in Visual Basic .NET.

Managed Extensions for C++ and attributed programming are just some of the enhancements made to the C++ language. Managed Extensions simplify the task of migrating existing C++ applications to the new .NET Framework. C# is Microsoft's new language. It's a C-style language that is essentially "C++ for Rapid Application Development". Unlike other languages, its specification is just the grammar of the language. It has no standard library of its own, and instead has been designed with the intention of using the .NET libraries as its own.

Microsoft Visual J# .NET provides the easiest transition for Java-language developers into the world of XML Web Services and dramatically improves the interoperability of Java-language programs with existing software written in a variety of

other programming languages. Active State has created Visual Perl and Visual Python, which enable .NET-aware applications to be built in either Perl or Python. Both products can be integrated into the Visual Studio .NET environment. Visual Perl includes support for Active State's Perl Dev Kit.

Other languages for which .NET compilers are available include

- FORTRAN
- COBOL
- Eiffel

C#.NET is also compliant with CLS (Common Language Specification) and supports structured exception handling. CLS is set of rules and constructs that are supported by the CLR (Common Language Runtime). CLR is the runtime environment provided by the .NET Framework; it manages the execution of the code and also makes the development process easier by providing services. C#.NET is a CLS-compliant language. Any objects, classes, or components that created in C#.NET can be used in any other CLS-compliant language. In addition, Users can use objects, classes, and components created in other CLS-compliant languages in C#.NET. The use of CLS ensures complete interoperability among applications, regardless of the languages used to create the application.

Constructors and Destructors

Constructors are used to initialize objects, whereas destructors are used to destroy them. In other words, destructors are used to release the resources allocated to the object. In C#.NET the sub finalize procedure is available. The sub finalize procedure is used to complete the tasks that must be performed when an object is destroyed. The sub finalize procedure is called automatically when an object is destroyed. In addition, the sub finalize procedure can be called only from the class it belongs to or from derived classes.

Garbage Collection

Garbage Collection is another new feature in C#.NET. The .NET Framework monitors allocated resources, such as objects and variables. In addition, the .NET Framework automatically releases memory for reuse by destroying objects that are no longer in use. In C#.NET, the garbage collector checks for the objects that are not currently in use by applications. When the garbage collector comes across an object that is marked for garbage collection, it releases the memory occupied by the object.

Overloading

Overloading is another feature in C#. Overloading enables us to define multiple procedures with the same name, where each procedure has a different set of arguments. Besides using overloading for procedures, users can use it for constructors and properties in a class.

Multithreading

C#.NET also supports multithreading. An application that supports multithreading can handle multiple tasks simultaneously, users can use multithreading to decrease the time taken by an application to respond to user interaction.

Structured Exception Handling

C#.NET supports structured handling, which enables us to detect and remove errors at runtime. In C#.NET, users need to use Try...Catch...Finally statements to create exception handlers. Using Try...Catch...Finally statements, users can create robust and effective exception handlers to improve the performance of our application.

2.2.3.6 SQL SERVER

The OLAP Services feature available in SQL Server version 7.0 is now called SQL Server 2000 Analysis Services. The term OLAP Services has been replaced with the term Analysis Services. Analysis Services also includes a new data mining component. The Repository component available in SQL Server version 7.0 is now called Microsoft SQL Server 2000 Meta Data Services. References to the component

now use the term Meta Data Services. The term repository is used only in reference to the repository engine within Meta Data Services

SQL-SERVER database consist of six type of objects, They are:

1. TABLE
2. QUERY
3. FORM
4. REPORT
5. MACRO

Table

A database is a collection of data about a specific topic.

Views of Table

Users can work with a table in two types,

1. Design View
2. Datasheet View

Design View

To build or modify the structure of a table users work in the table design view. Users can specify what kind of data will be hold.

Datasheet View

To add, edit or analyses the data itself, users work in tables datasheet view mode.

Query

A query is a question that has to be asked the data. Access gathers data that answers the question from one or more table. The data that make up the answer is either dynaset (if you edit it) or a snapshot (it cannot be edited).Each time user run

query, user get latest information in the dynaset. Access either displays the dynaset or snapshot for us to view or perform an action on it, such as deleting or updating

2.3 DESIGN

2.3.1 UML DIAGRAM

The **Unified Modeling Language (UML®)** is a standard visual modeling language intended to be used for

1. modeling business and similar processes,
2. analysis, design, and implementation of software-based systems

UML is a common language for business analysts, software architects and developers used to describe, specify, design, and document existing or new business processes, structure and behavior of artifacts of software systems.

UML can be applied to diverse **application domains** (e.g., banking, finance, internet, aerospace, healthcare, etc.) It can be used with all major object and component **software development methods** and for various **implementation platforms** (e.g., J2EE, .NET).

UML is a standard modeling **language**, not a **software development process**. It

1. provides guidance as to the order of a team's activities,
2. specifies what artifacts should be developed,
3. directs the tasks of individual developers and the team as a whole,

and

4. offers criteria for monitoring and measuring a project's products and

activities.

2.3.1.1 USE CASE DIAGRAM

The purpose of use case diagram is to capture the dynamic aspect of a system. But this definition is too generic to describe the purpose. Use case diagrams are used to gather the requirements of a system including internal and external influences.

These requirements are mostly design requirements. So when a system is analyzed to gather its functionalities use cases are prepared and actors are identified.

Now when the initial task is complete use case diagrams are modelled to present the outside view.

So in brief, the purposes of use case diagrams can be as follows:

1. Used to gather requirements of a system.
2. Used to get an outside view of a system.
3. Identify external and internal factors influencing the system.
4. Show the interacting among the requirements are actors.

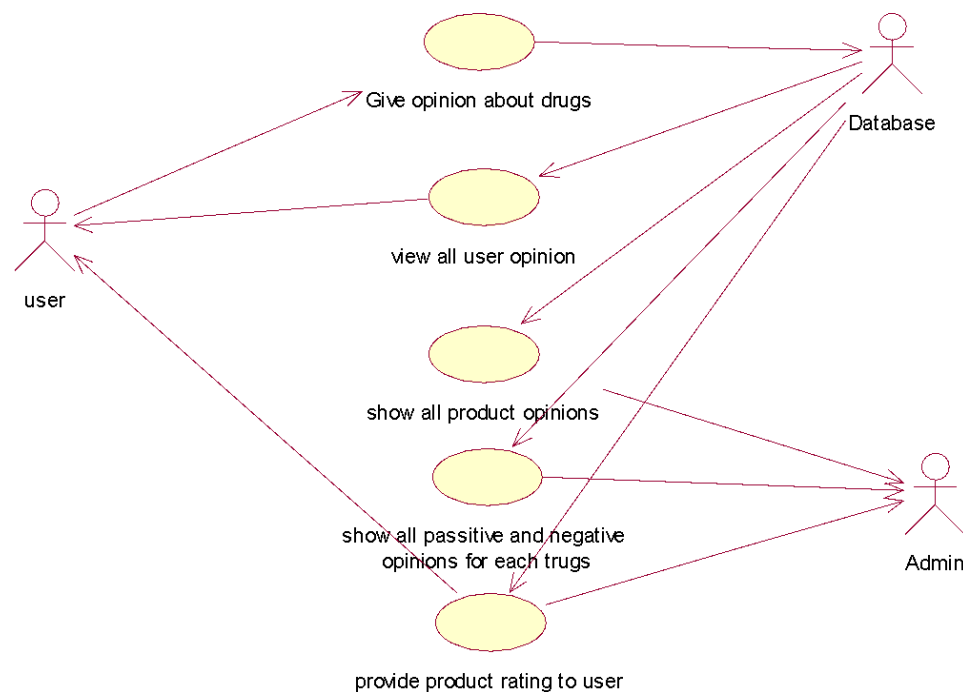


Fig 2.1 **USE CASE DIAGRAM**

2.3.1.2 CLASS DIAGRAM

The class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing and documenting different aspects of a system but also for constructing executable code of the software application.

The class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The class diagrams are widely used in the modelling of object oriented systems because they are the only UML diagrams which can be mapped directly with object oriented languages.

The class diagram shows a collection of classes, interfaces, associations, collaborations and constraints. It is also known as a *structural diagram*.

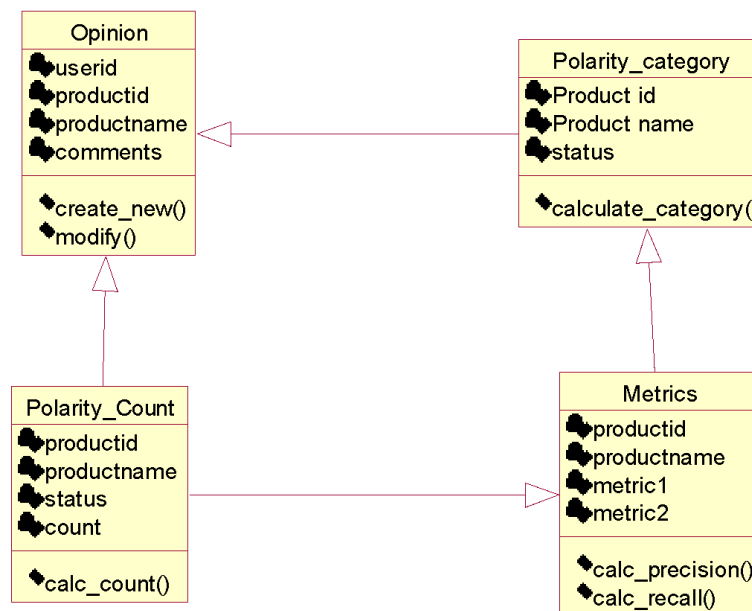


Fig 2.2 CLASS DIAGRAM

2.3.1.3 ACTIVITY DIAGRAM

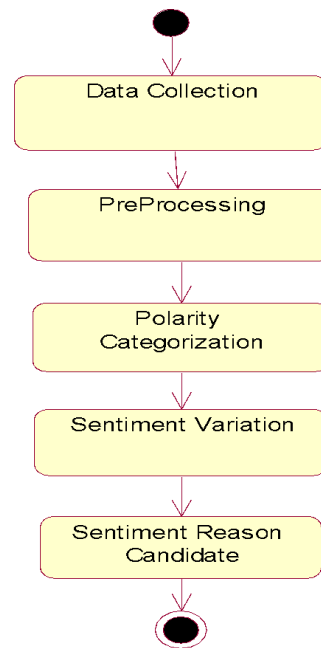


Fig 2.3 ACTIVITY DIAGRAM

2.3.1.4 SEQUENCE DIAGRAM

A **Sequence diagram** is an interaction diagram that shows how processes operate with one another and in what order. It is a construct of a Message Sequence Chart. A sequence diagram shows object interactions arranged in time sequence. It depicts the objects and classes involved in the scenario and the sequence of messages exchanged between the objects needed to carry out the functionality of the scenario. Sequence diagrams are typically associated with use case realizations in the Logical View of the system under development. Sequence diagrams are sometimes called **event diagrams** or **event scenarios**.

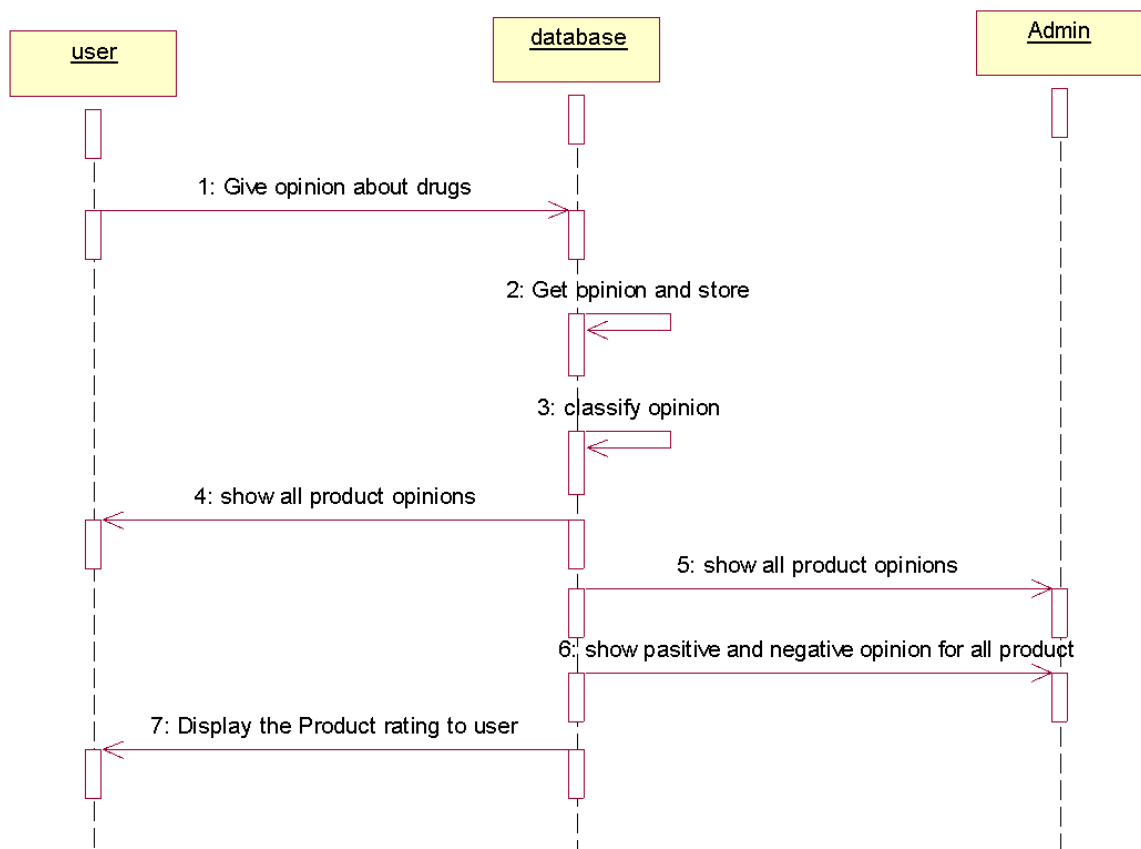


Fig 2.4 SEQUENCE DIAGRAM

2.3.1.5 COLLABORATION DIAGRAM

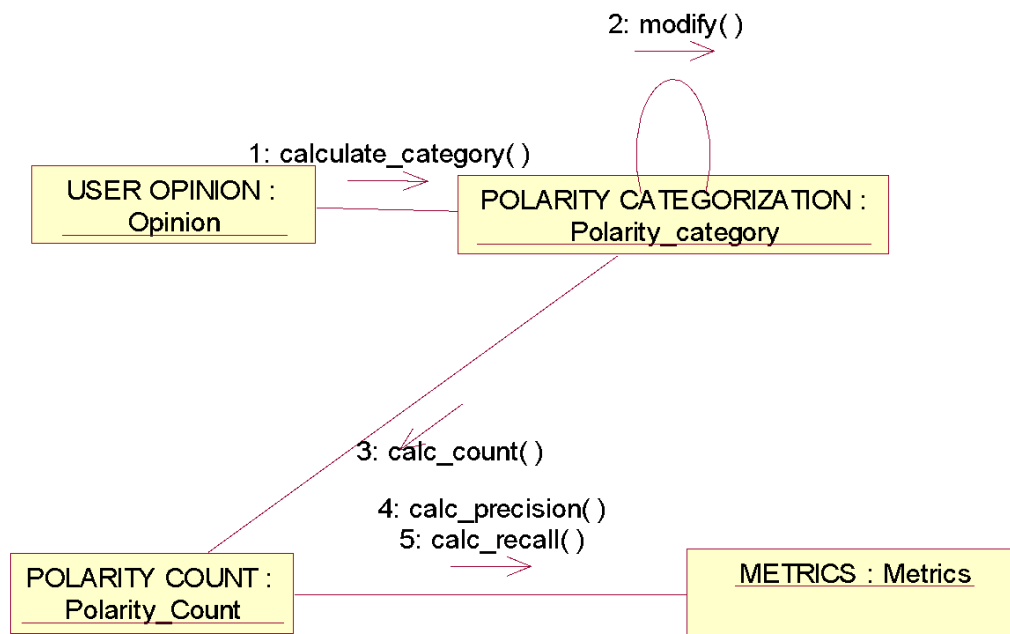


Fig. 2.5 COLLABORATION DIAGRAM

2.3.1.6 COMPONENT DIAGRAM

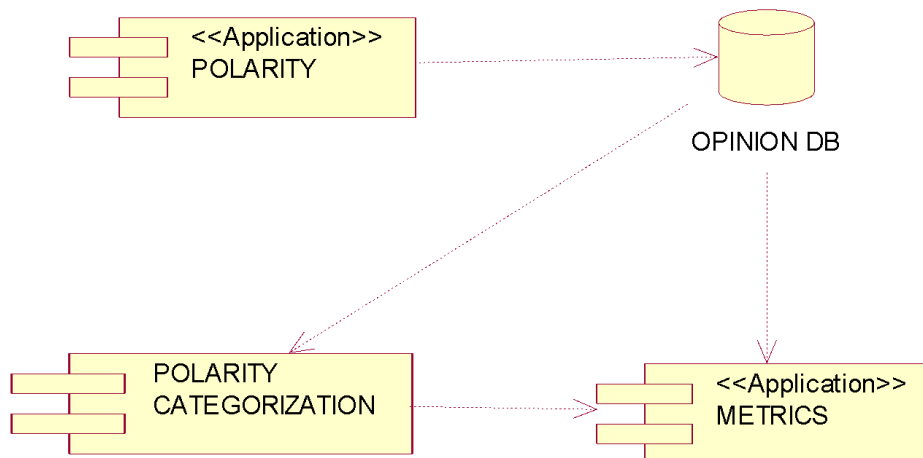


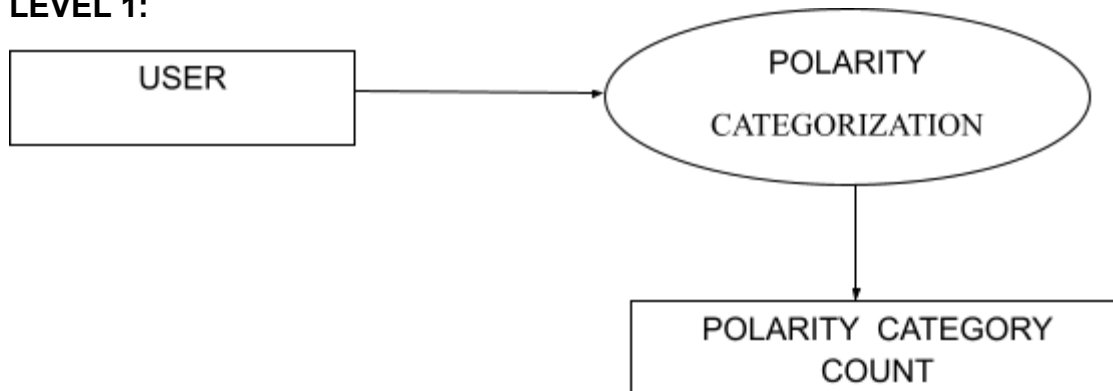
Fig. 2.6 COMPONENT DIAGRAM

2.3.1.7 DATA FLOW DIAGRAM

LEVEL 0:



LEVEL 1:



LEVEL 2:

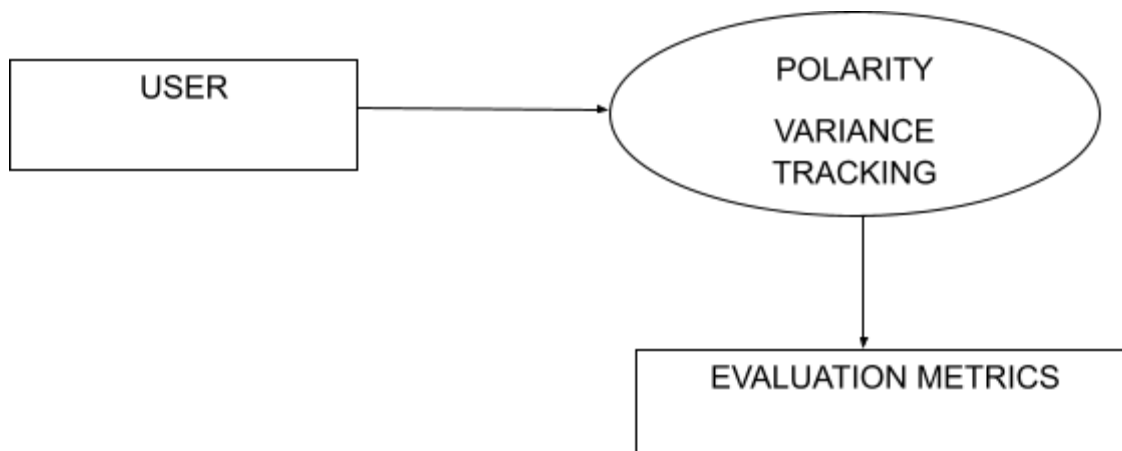


Fig. 2.7 DATA FLOW DIAGRAM

2.3.1.8 FLOW CHART DIAGRAM

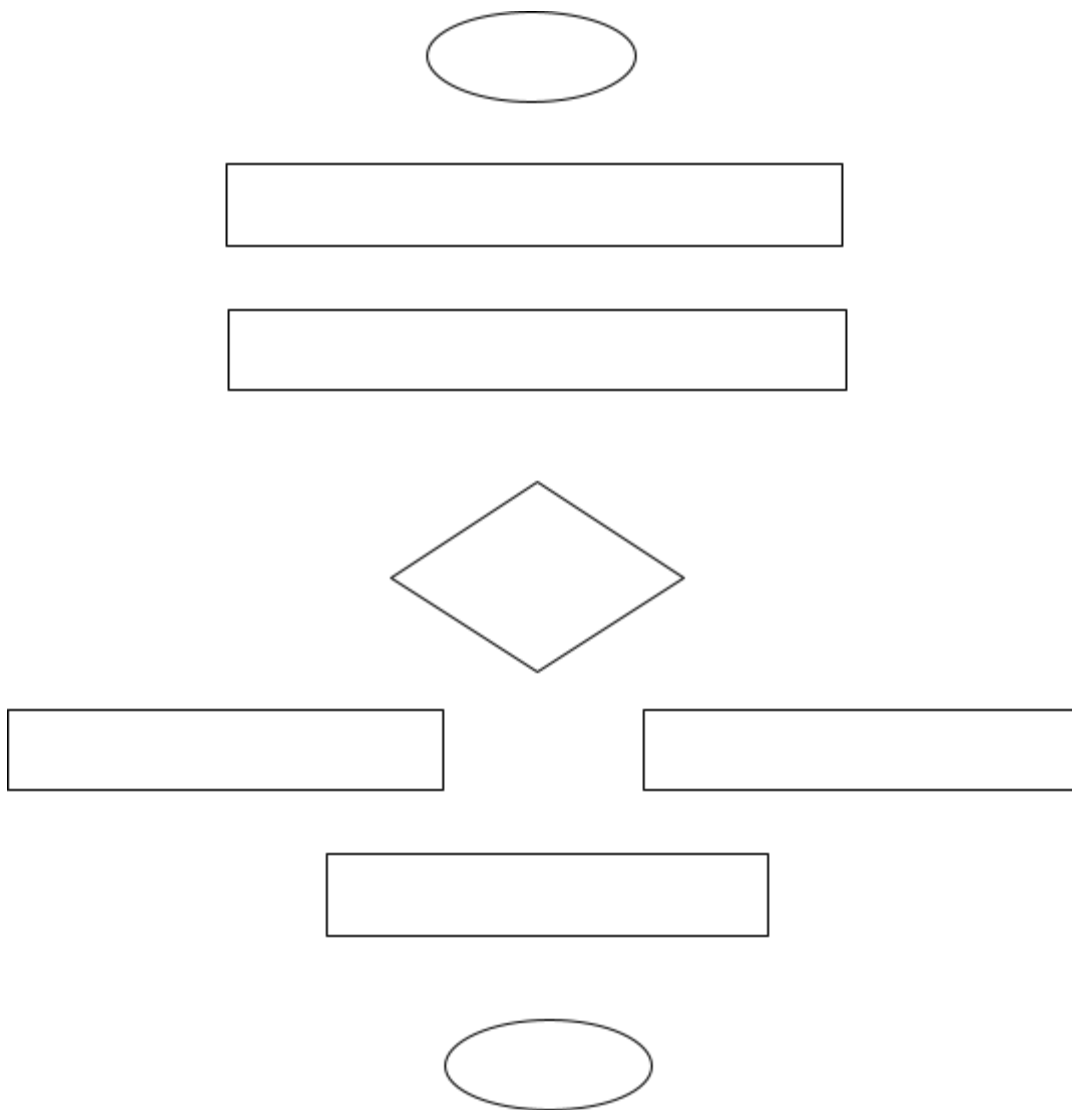


Fig. 2.8 FLOW CHART DIAGRAM

2.4 DATABASE DESIGN

Database design is a detailed data model of a database. The logical data model contains the needed logical and physical design and physical storage parameters to generate a design in a data definition language(DDL), which can then be used to create a database. A fully attributed data model contains detailed attributes for each entity.

The term database design use different parts of the design of a database system. The logical design of the base data structures used to store the data. In the relational model these are the tables and views. In an object database the entities and relationships map directly to object classes and named relationships. The term database design used to apply to the process of designing, not just the base data structures, but also the forms and queries used as part of the database application within the database management system(DBMS)

Table 2.1: Opinion

Column Name	Data Type	Key	Allows Null
Productid	Number(3)	Primary Key	No
ProductName	Varchar2(2)		No
Userid	Varchar2(15)	Unique	No
Comments	Varchar2(50)		No

Table 2. 2: Polarity_Category

Column Name	Data Type	Key	Allows Null
ProductName	Varchar2(20)	Primary Key	No
Review	Varchar2(15)		No
Status	Varchar2(50)		No

Table 2.3: Polarity_Count

Column Name	Data Type	Key	Allows Null
ProductName	Varchar2(20)	Primary Key	No
Status	Varchar2(50)		No
Count	Number(2)		No

Table 2.4: Metrics

Column Name	Data Type	Key	Allows Null
ProductName	Varchar2(20)	Primary Key	No
Metric1	Number(2)		No
Metric2	Number(2)		No

CHAPTER 3

SYSTEM DESIGN

3.1 DEVELOPMENT METHODOLOGY

1. Collect unstructured data from Social Media sources.
2. Real-Time Processing with a sentiment analysis engine based on keyword search.
3. Store processed data (with sentiment) in SQL SERVER database.
4. Extract sentiments from SQL SERVER to visualization layer.

5. Visualize with a tool of choice .

3.2 SYSTEM ARCHITECTURE

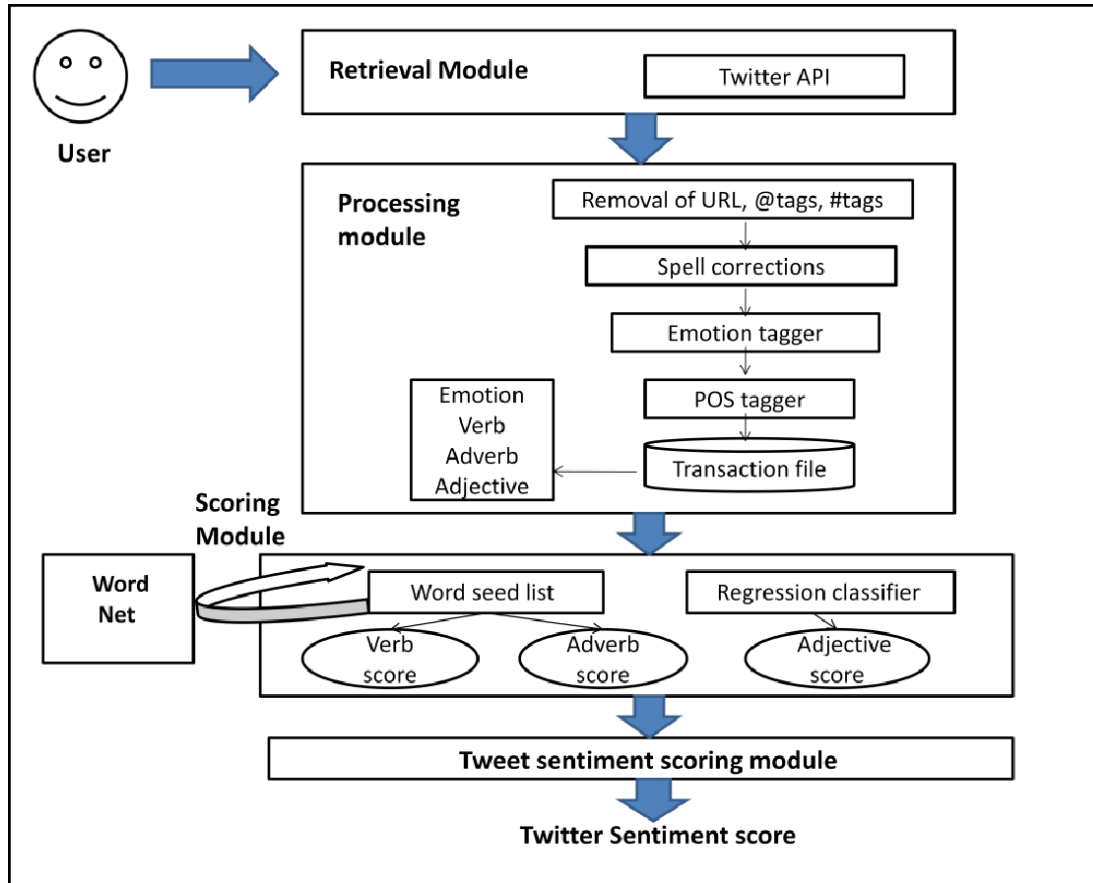


Fig 3.1 SYSTEM ARCHITECTURE OF SENTIMENTAL ANALYSIS

3.3 MODULE DESCRIPTION

The proposed system has the following modules ;

1. DATA STREAMING
2. PREPROCESSING
3. SENTIMENT POLARITY ANALYSIS

4. EVALUATION METRICS

The details of the modules are presented below.

3.3.1. DATA STREAMING: Extracting real time tweets using Twitter Streaming API.

For classification and training the classifier users need Twitter data. For this purpose user make use of API's twitter provides. Twitter provides two API's; Stream API¹ and REST API². The difference between Streaming API and REST APIs are: Streaming API supports long-lived connection and provides data in almost real -time. The REST APIs support short-lived connections and are rate-limited (one can download a certain amount of data [*150 tweets per hour] but not more per day). REST APIs allow access to Twitter data such as status updates and user info regardless of time. However, Twitter does not make data older than a week or so available. Thus REST access is limited to data Twittered not before more than a week. Therefore, while REST API allows access to these accumulated data, Streaming API enables access to data as it is being twittered.

Streaming Twitter API and the public Streams version of it used in this project. All the public data flowing through Twitter currently could be collected using this Public Streams API. Though it is a low latency process it provides considerable amount of data to work on when run continuously for several hours. Twitter Feeds are accessed using the Streaming API and then tweets related to the query passed is obtained and the content is piped in to the next phase of the pre-processing framework.

3.3.2. PREPROCESSING :

In this phase, the tweets are available as text data and each line contains a tweet. Initially tweets are cleaned up or remove retweets as that will induce a bias in the classification process. the punctuations need to be removed and other symbols that doesn't make any sense as it may result in inefficiencies and may affect the accuracy of the overall process. Also removal of tweets that contain both a positive and negative emoticons is necessary as it may cause a lot of confusion as far as automatic

classification is considered. This results in a clean Twitter Text Corpus which is then sent to the second phase of pre-processing.

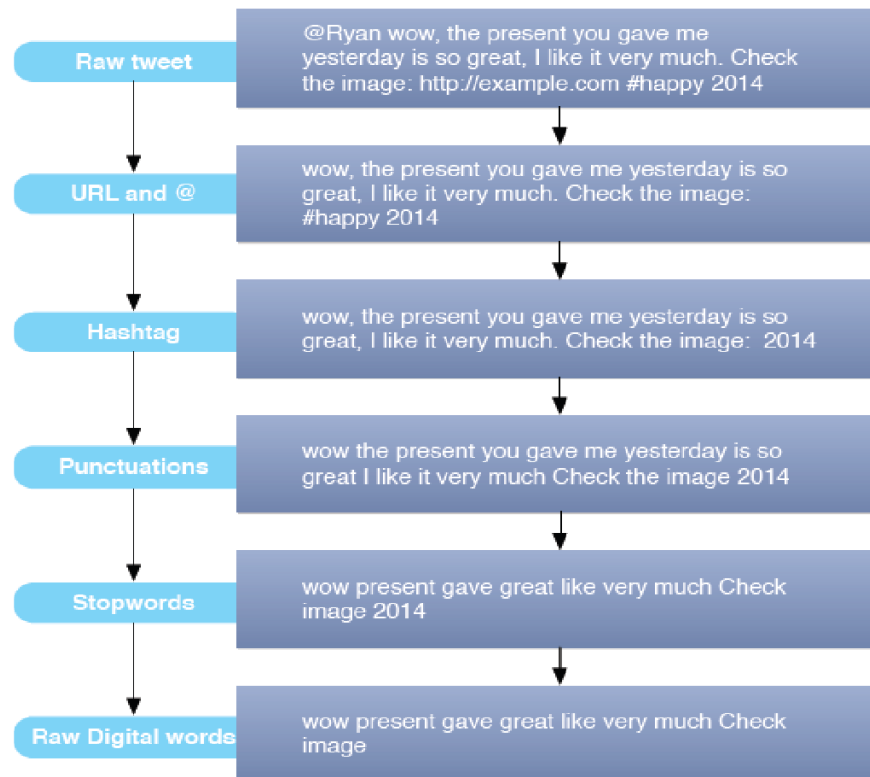


Fig 3.2 PROCESS FLOW FOR PREROCESSING TWEETS

a. TOKENIZATION:

In this phase, tweets are tokenized (split them into individual words of importance). Tokenization is the process dividing entire text into meaningful words referred as tokens. A token may be a symbol, phrase or simply a word. Tweet: “Sachin Tendulkar is best cricketer in the history of cricket”

Table 3.1: List of tokens generated on a tweet

TOKEN ID	TOKEN
1	Sachin
2	Tendulkar
3	is
4	best
5	cricketer
6	in
7	the

8	history
9	of
10	cricket

Once tokenized, the next step to be performed is normalization which is an important part of the text pre-processing process, which is normalization. Here repeat letters are eliminated from words that are put in to provide stress or a better context to the tweet. Also expressions are normalized and these expressions are usually used to express laughter, sorrow etc. which may contain a series of repetitive characters.

b. STOP WORDS REMOVAL:

This step removes the various stop words that constitutes the grammar of the sentence as they do not give us significant information (like the, for, who etc). For this a corpus of stop words are gathered which is used for removing the stopwords from the tweets. Removing stop words makes text processing easy. Stop words based on different criteria was evaded. Secondly, the tokens based on length to avoid some freaky terms can be neglected. Finally tokens based on POS tags are filtered. This will help in getting meaningful tokens. After applying filtering on the above mentioned tweet the result is as follows

Table 3.2: List of tokens generated after stop words removal

TOKEN ID	TOKEN
1	Sachin
2	Tendulkar
3	best
4	cricketer
5	history
6	cricket

The removal of special characters (like #, . , ^ , \$) which are unnecessary for analysis is also eliminated. Retweets look like normal Tweets with the author's name and username next to it, but are distinguished by the Retweet icon and the name of the

user who retweeted the Tweet. Retweets contain the copy of the original tweet, so it adds to redundancy of information. So in this step, removal of the retweets carried out so that training data contains the unique tweets only.

c. STEMMING: Stemming is the process of replacing each word. Thus, in this usage, the English word *cricketer* contains the stem *cricket*, to which the derivational suffix - *er* is attached. Both the tokens cricketer and cricket reflects the same mean of a game.

Table 3.3: List of tokens generated after Stemming

TOKEN ID	TOKEN
1	Sachin
2	Tendulkar
3	Best
4	History
5	Cricket

d. CASE TRANSFORMATION - involves transferring lower case data into upper case and vice versa. This will maintain uniqueness in the training data.

Table 3.4: List of tokens generated after Case Transformation

TOKEN ID	TOKEN
1	sachin
2	tendulkar
3	best
4	history
5	cricket

Some words start with a digit, like '1990', '4:00pm' etc. These words also have no relationship with attitudes or feelings. So these words should be removed.

3.3.3 SENTIMENT POLARITY ANALYSIS

Polarity analysis is a process of identifying the reviews into three. Positive, Negative and Partial. The classical Naive Bayes based sentiment analysis algorithm is adjusted to fit into polarity model. In this project, a Naive Bayes classifier is employed and also an English lexical dictionary SentiWordNet.

3.4. ALGORITHM IMPLEMENTATION

3.4.1 Sentiment Calculation Algorithm:

Sentiment calculation is done for every tweet and a polarity score is given to it. If the score is greater than 0 then it is considered to a positive sentiment on behalf of the user, if less than 0 then negative else neutral. The polarity score is calculated by using algorithm 1 by using sentiment analysis programming model.

Algorithm 1: ALGO_SENTICAL

Input: Tweets, SentiWord_Dictionary

Output: Sentiment (positive, negative or neutral)

BEGIN

1) For each tweet T_i do the following

2) Initialize SentiScore = 0;

3) For each word W_j in T_i that exists in Sentiword_Dictionary.

If polarity[W_j] = blind negation then Return negative.

Else

b.1. If polarity[Wj] = positive && strength[Wj] = Strongsubj then increment seniscore by 1.

b.2 Else If polarity[Wj] = positive && strength[Wj] = Weaksubj then add 0.5 to sentiscore.

b.3. Else If polarity[Wj] = negative && strength[Wj] = Strongsubj then decrement sentiscore by 1.

b.4 .Else If polarity[Wj] = negative && strength[Wj] = Weaksubj then subtract 0.5 from sentiscore.

b.5. If polarity[Wj] = negation multiply sentiscore by -1.

If Sentiscore of Ti >0 then Sentiment = positive.

Else If Sentiscore of Ti <0 then Sentiment = negative.

Else Sentiment = neutral

END

4) Return Sentiment

3.4.2 NAÏVE BAYES Algorithm:

The equation of Naive Bayes for sentiment classification is given by:

$$P(x_i | c) = \frac{\text{Count of } x_i \text{ in documents of class } c}{\text{Total no of words in documents of class } c}$$

It can be seen from the equation that, two most important values needs to be computed are: the posterior probabilities ,class prior probabilities $P(C)$. For a training set, it seems that in order to get these two values, there is a need to have some counts. Here variable C refers to the label of the tweet, and X to refer the features (words). The following values need to be calculated:

1. Number of tweets: the number of all tweets.
2. Number of C : the number of unique labels.
3. Number of $C = c$: the number of tweets with label c .
4. Number of X in $C = c$: the number of words that are in all the tweets with label c .
5. Number of $X = x$ in $C = c$: the number of word x that is in all the tweets with label c .
6. $|V|$: the total number of unique words in all tweets. This is actually the size of the vocabulary.

Using the above data, it is possible to get the prior probabilities $P(C)$ and likelihoods $P(D|C)$. It is obvious that only one mapper and reducer job is not enough. The goal is to build the prediction model, with the training phase and testing phase. In general there are two jobs: the word job and the label job. For each job there is a map function and a reduce function. The word job is to compute the likelihood of each word, and the label job is to compute the prior probability of each label.

3.5 EVALUATION METRICS

The following Information Retrieval matrices are used to evaluate the experiment results

- Precision = $TP / (TP + FP)$
- Recall = $TP / (TP + FN)$
- F-measure = $2 * Precision * recall / (Precision + recall)$
- Accuracy = $(TP + TN) / (TP + TN + FP + FN)$

CHAPTER 4

RESULTS AND DISCUSSION

4.1 DESCRIPTION OF EXPERIMENTS CONDUCTED

Data used in this project is a set of product reviews collected from Social Networking Sites(SNS) from November 2016 to April 2017. Those online reviews were about medicines and nearly 8000 reviews towards 20 products. Each review includes the following information: 1) reviewer ID; 2) product ID; 3) rating; 4) time of the review; 5) helpfulness; 6) review text. Every rating is based on a 5-star scale resulting all the ratings to be ranged from 1-star to 5-star with no existence of a half-star or a quarter-star.

Sentiment sentences extraction and POS tagging, Negation phrases identification, Sentiment score computation for sentiment tokens and Polarity tagging

procedures were completed on the dataset. Finally, Classification of user reviews done with a help of K nearest neighbor and Naïve Bayesian classifiers.

4.2 OUTPUT WITH DESCRIPTION

4.2.1 *Sentiment sentences extraction and POS tagging*

Every word of a sentence has its syntactic role that defines how the word is used. The syntactic roles are also known as the parts of speech. There are 8 parts of speech in English: the verb, the noun, the pronoun, the adjective, the adverb, the preposition, the conjunction, and the interjection. In natural language processing, part-of-speech (POS) taggers have been developed to classify words based on their parts of speech. For sentiment analysis, a POS tagger is very useful because of the following two reasons: 1) Words like nouns and pronouns usually do not contain any sentiment. It is able to filter out such words with the help of a POS tagger; 2) A POS tagger can also be used to distinguish words that can be used in different parts of speech. For instance, as a verb, "enhanced" may conduct different amount of sentiment as being of an adjective. The POS tagger used for this research is a max-entropy POS tagger developed for the Penn Treebank Project. The tagger is able to provide 46 different tags indicating that it can identify more detailed syntactic roles than only 8. As an example, Table [1](#) is a list of all tags for verbs that has been included in the POS tagger.

Table 4.1: Part-of-Speech tags for verbs

Tag	Definition
VB	base form
VBP	present tense, not 3rd person singular
VBZ	present tense, 3rd person singular
VBD	past tense
VBG	present participle

VCN	past participle
-----	-----------------

4.2.2Negation phrases identification

Words such as adjectives and verbs are able to convey opposite sentiment with the help of negative prefixes. For instance, consider the following sentence that was found in an electronic device’s review: “The built in speaker also has its uses but so far nothing revolutionary.” The word, “revolutionary” is a positive word however, the phrase “nothing revolutionary” gives more or less negative feelings. Therefore, it is crucial to identify such phrases. In this work, there are two types of phrases have been identified, namely negation-of-adjective (NOA) and negation-of-verb (NOV).

4.2.3Sentiment score computation for sentiment tokens

A sentiment token is a word or a phrase that conveys sentiment. A word token consists of a positive (negative) word and its part-of-speech tag. In total, 200 word tokens are selected with each of them that occur at least 30 times throughout the dataset. For phrase tokens, 12 phrases were selected of the 486 identified sentiment phrases, which each of the 486 phrases also has an occurrence that is no less than 30.

4.2.4Sentiment Calculation Algorithm:

Sentiment calculation is done for every tweet and a polarity score is given to it. If the score is greater than 0 then it is considered to a positive sentiment on behalf of the user, if less than 0 then negative else neutral. The polarity score is calculated by using algorithm 1 by using sentiment analysis programming model

CHAPTER 5

CONCLUSION AND FUTURE WORK

5.1 SUMMARY

Sentiment Analysis is the process of determining whether a piece of writing is positive, negative or neutral. It's also known as opinion mining, deriving the opinion or

attitude of a speaker. A common use case for this technology is to discover how people feel about a particular topic. Say you want to know if people on Twitter think that Chinese food in San Francisco is good or bad. Twitter sentiment analysis will answer this question.

Sentiment Analysis can let you learn why people think the food is good or bad, by extracting the exact words that indicate why people did or didn't like the food. If too salty shows as a common theme, for example, you immediately have a better idea of why consumers aren't happy.

5.2 CONCLUSION

Sentiment analysis or opinion mining is a field of study that analyzes people's sentiments, attitudes, or emotions towards certain entities. This project tackles a fundamental problem of sentiment analysis, sentiment polarity categorization. Online product reviews from Twitter, Flipkart and Amazon.com are selected as data used for this project. A sentiment polarity categorization process has been proposed along with detailed descriptions of each step. Experiments for both sentence-level categorization and review-level categorization have been performed.

5.3 FUTURE WORK

Subjective Well Being factors of sentiment analysis can be further investigated to identify why peoples' opinion differ. The latent attributes of Social Networking Site(SNS) users could be considered in a detailed manner to further analyze the happiness factor of users. A Friend Follower network of SNS users could be constructed and further research can be done.

REFERENCES

- [1]. Xing Fang, Justin Zhan Sentiment analysis using product review data In: Journal of Big Data (2015) 2: 5. doi:10.1186/s40537-015-0015-2
- [2]. Hu M, Liu B (2004) Mining and summarizing customer reviews In: Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining, 168–177.. ACM, New York, NY, USA.
- [3]. Gann W-JK, Day J, Zhou S (2014) Twitter analytics for insider trading fraud detection system In: Proceedings of the second ASE international conference on Big Data. ASE.
- [4]. Johan Bollen, Bruno Gonçalves, Guangchen Ruan, & Huina Mao Happiness is assortative in online social networks. Artificial Life volume 17 issue 3, 2011 237-251
- [5]. Pennacchiotti, M., and Popescu, A. 2011. A machine learning approach to twitter user classification. In Proceedings of the International Conference on Weblogs and Social Media
- [6]. Mahmoud Elgamal , Sentiment Analysis Methodology of Twitter Data with an application on Hajj season International Journal of Engineering Research & Science [Vol-2, Issue-1, January- 2016 ISSN - [2395-6992]
- [7]. Agarwal, A., Xie, B., Vovsha, I., Rambow, O., and Passonneau, R., "Sentiment analysis of twitter data", In Proceedings of the Workshop on Language in Social Media (LSM 2011), pages 30–38, Portland, Oregon. Association for Computational Linguistics
- [8]. Rasheed M. Elawady, Sherif Barakat, Nora M.Elrashidy, "Different Feature Selection for Sentiment Classification, "International Journal of Information Science and Intelligent System, 3(1): 137-150, 2014
- [9]. Liu B. In: Web data mining; exploring hyperlinks, contents and usage data. Carey MJ, Ceri S, editors. Berlag Berlin Heidelberg: Springer; 2006

[10]. Agarwal, N.; Liu, H.; Murthy, S.; Sen, A.; and Wang, X. 2009. A social identity approach to identify familiar strangers in a social network. Proceedings of International Conference on Weblogs and Social Media (ICWSM 2009) 1–8

[11]. N. Barbieri, G. Manco, and F. Bonchi. Who to follow and why: Link prediction with explanations. In Proceedings of the International Conference on Knowledge Discovery and Data Mining (KDD 2014), New York City, USA, 2014.

[12]. Stefano Faralli, Giovanni Stilo and Paola Velardi, Large Scale Homophily Analysis in Twitter Using a Twixonomy. In Proceedings of the Twenty-Fourth International Joint Conference on Artificial Intelligence (IJCAI 2015)

APPENDICES

A.1 SCREENSHOTS

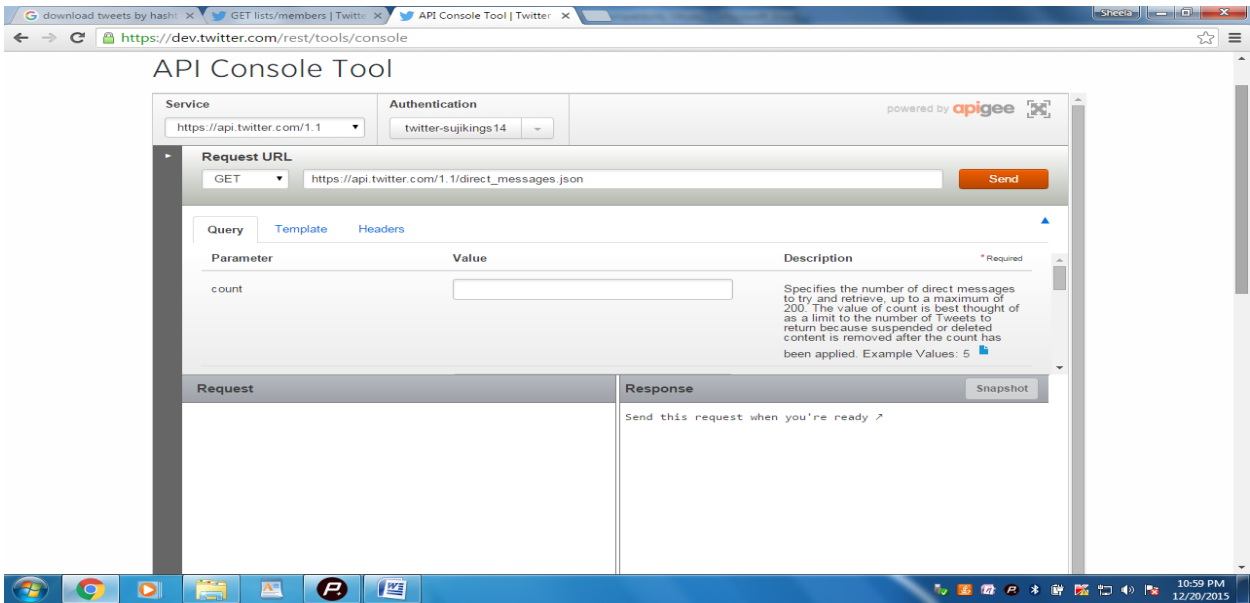


Fig. A.1.1: API Console Tool

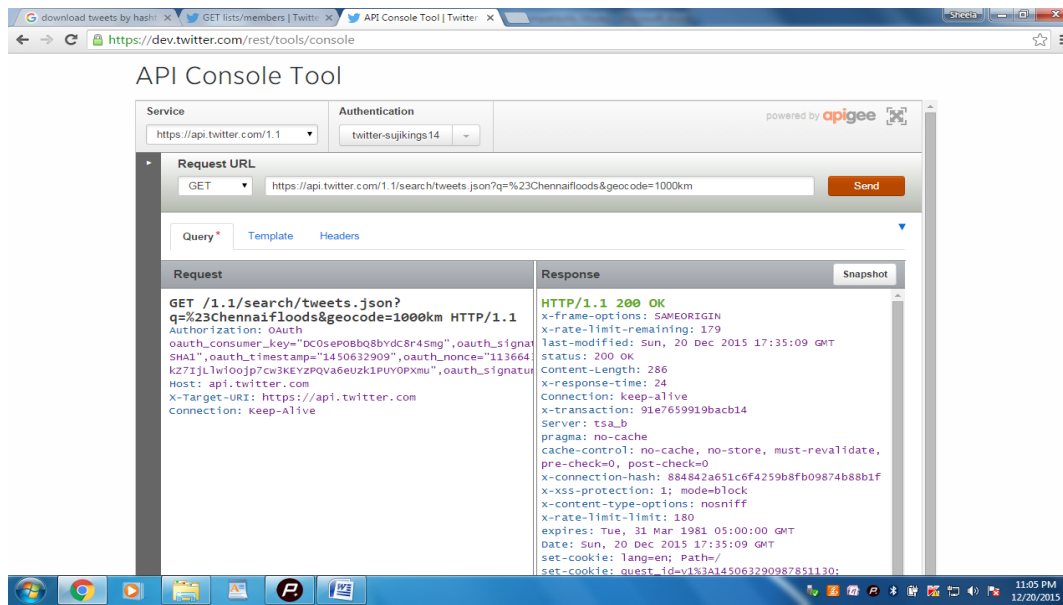


Fig. A.1.2: GET method & JSON

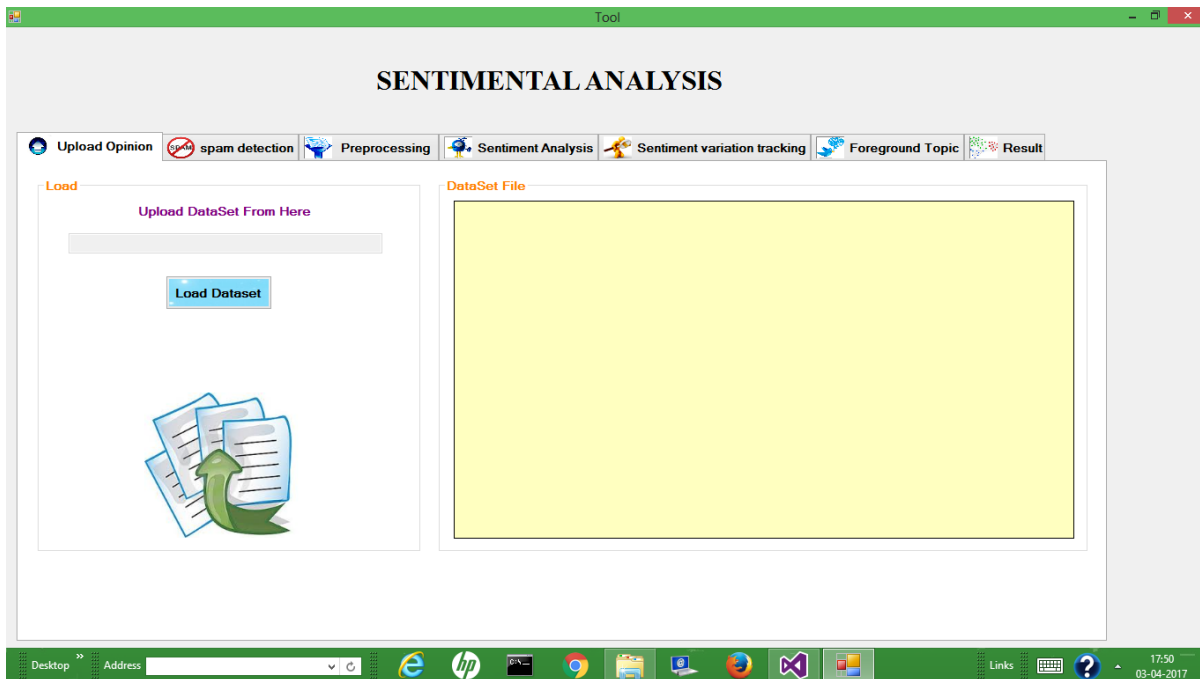


Fig A.1.3: GUI of Sentimental Analysis

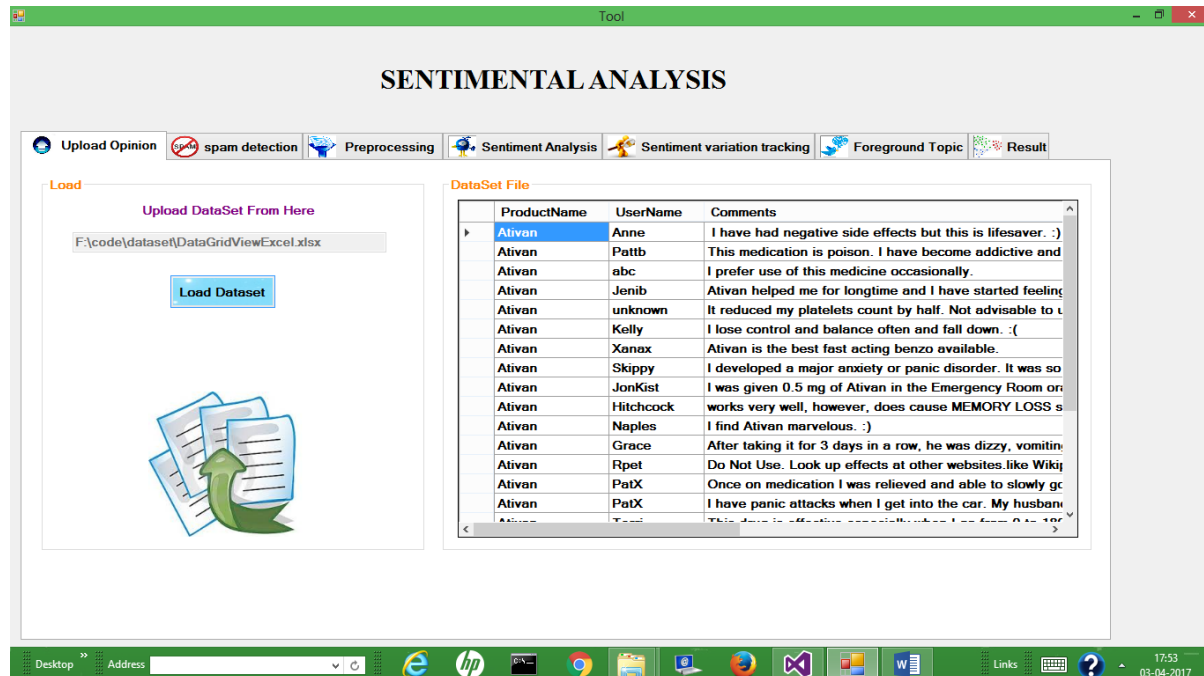


Fig. A.1.4: Loading Dataset

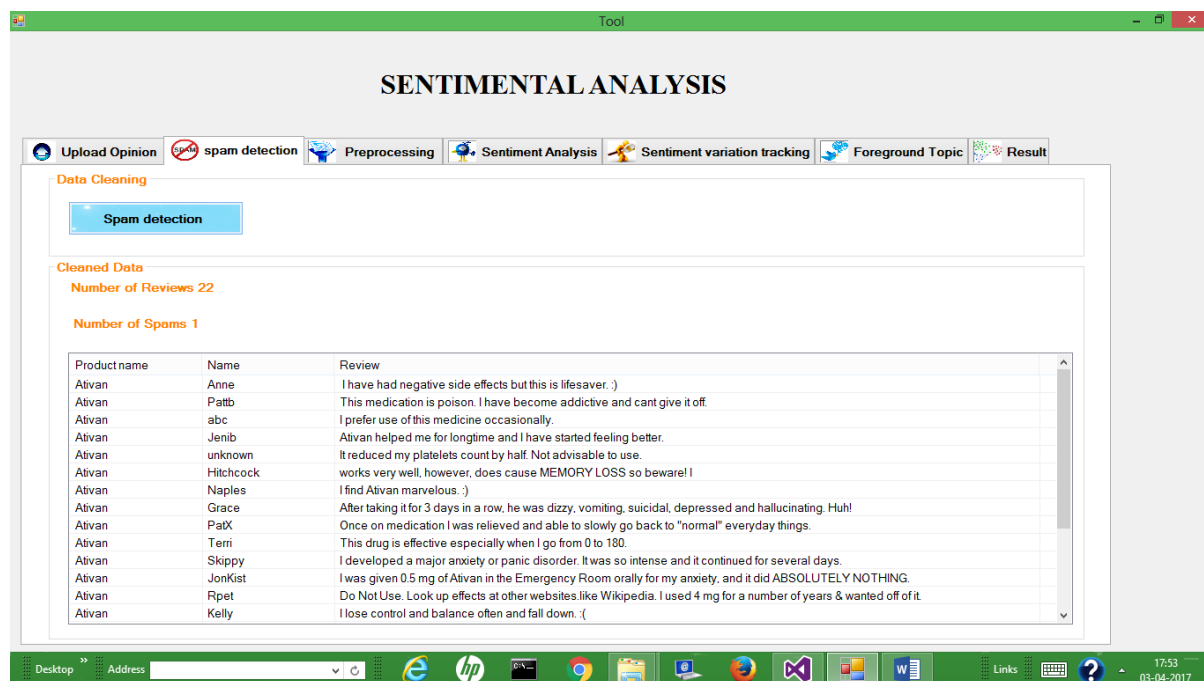


Fig. A.1.5: Spam Detection

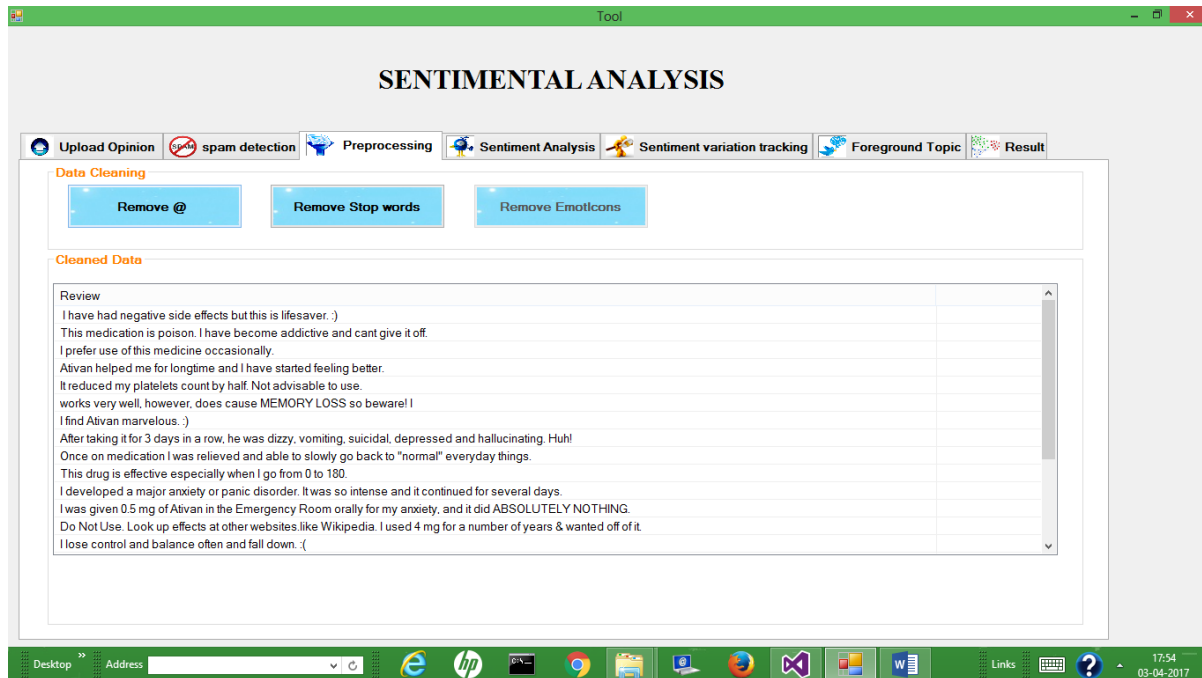


Fig. A.1.6: Removal of symbols, stop words and emonticons

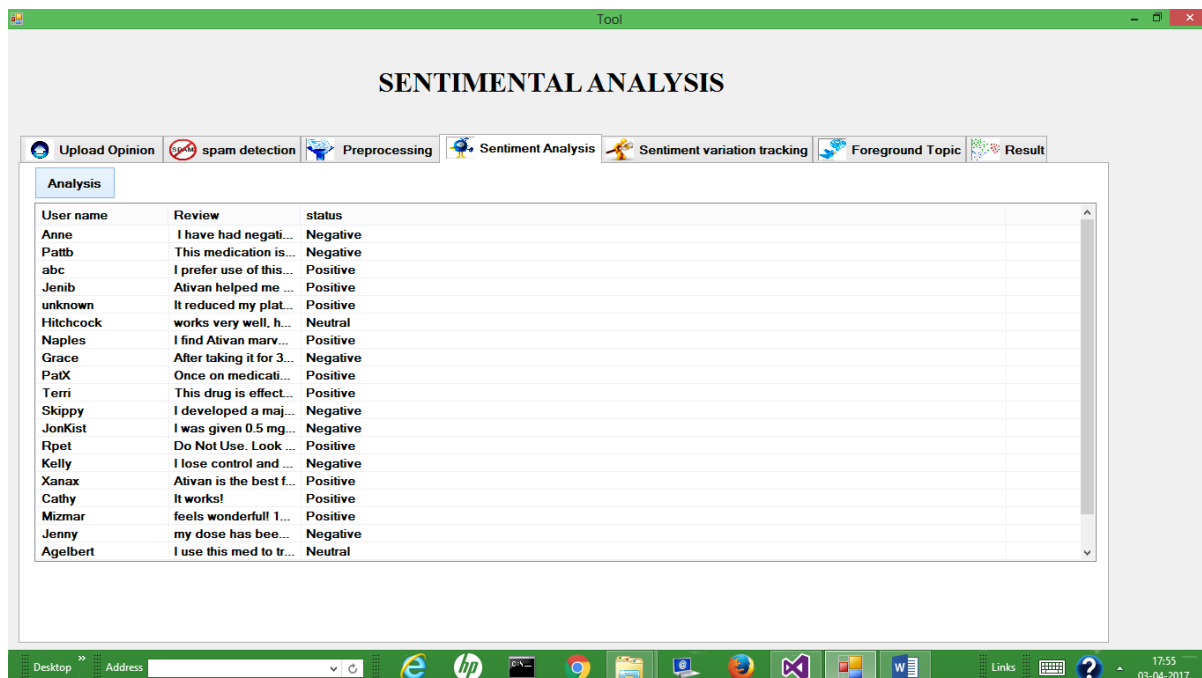


Fig. A.1.7 : Polarity of tweets

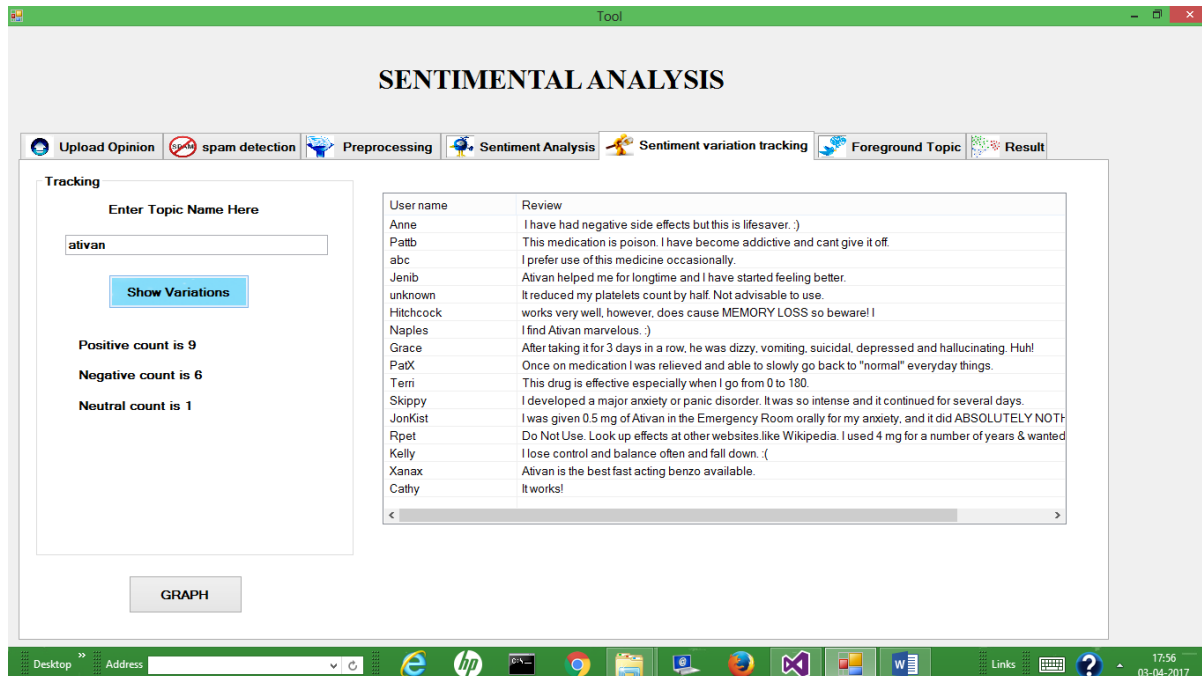


Fig.A.1.8: Sentiment Variation Tracking

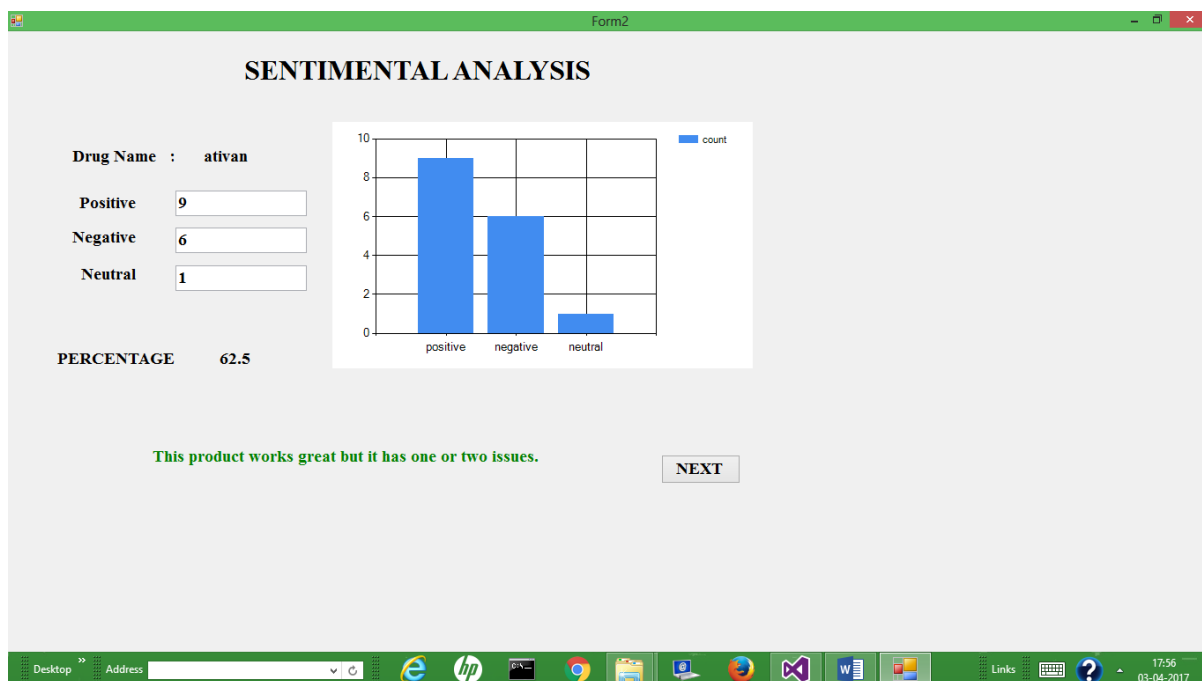


Fig A.1.9: Chart showing the count of polarities

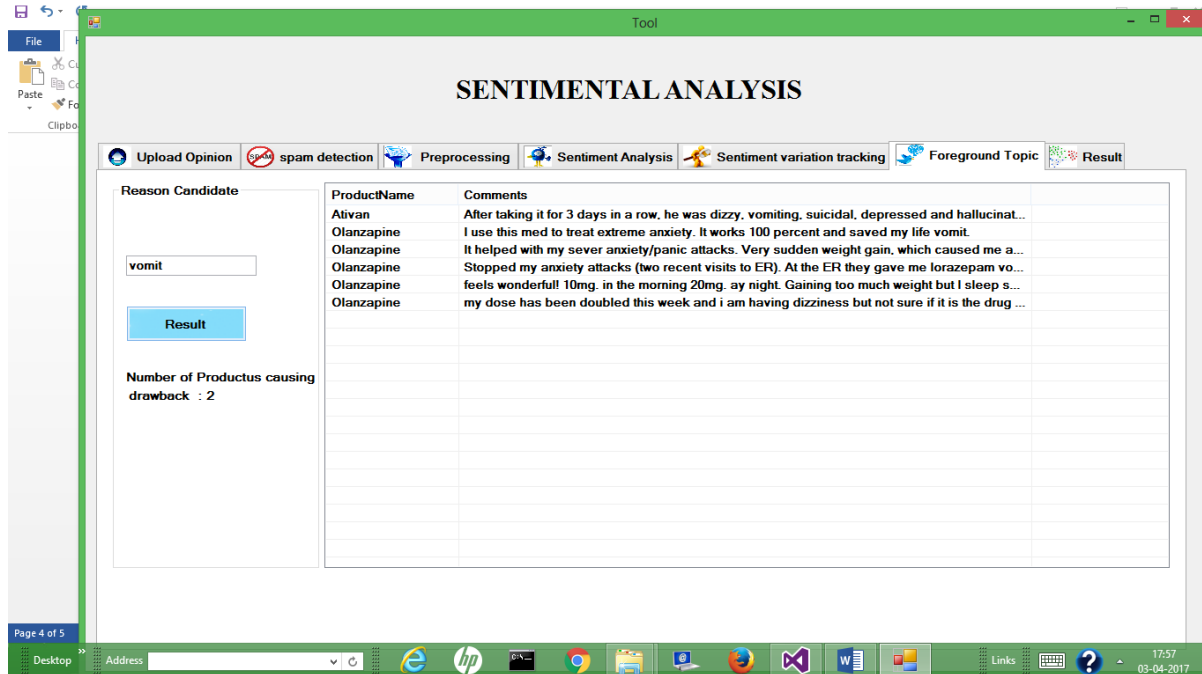


Fig. A.1.10: Reason Candidate

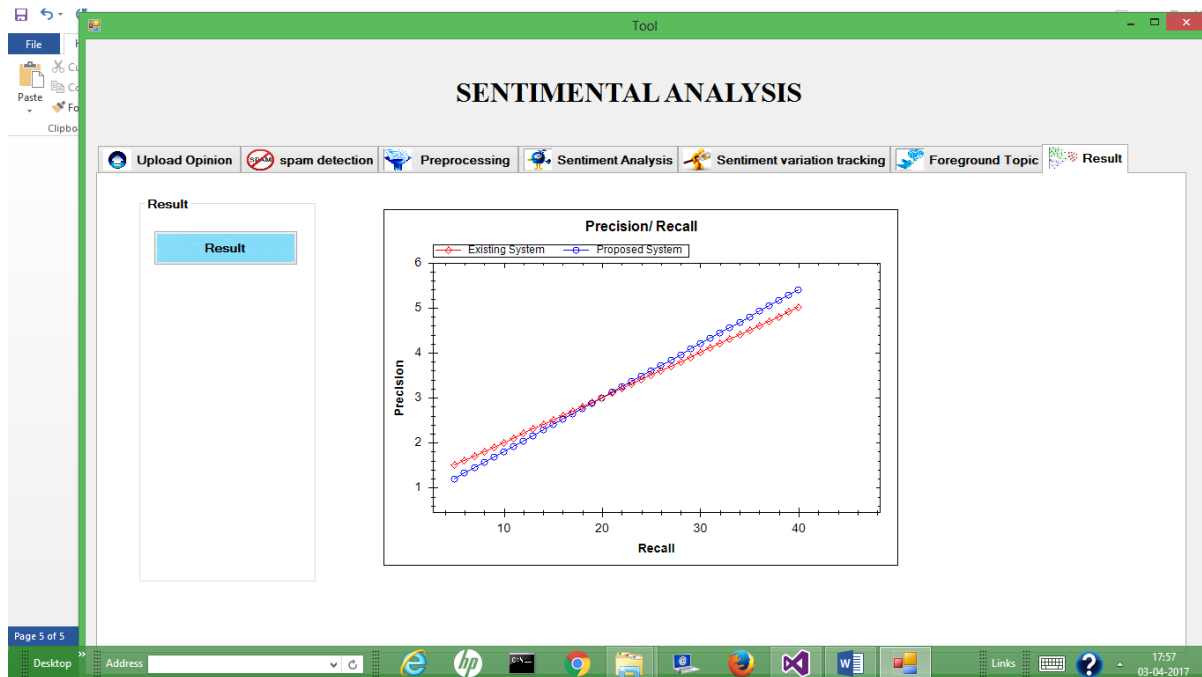


Fig. A.1.11: Evaluation Metrics

A.2 SAMPLE CODE

Getting tweets from twitter

```
using System;
using System.Collections.Generic;
using System.Linq;
using System.Net.Http;
using System.Text;
using System.Threading.Tasks;
using System.Web.Script.Serialization;

namespace Twitter
{
    public class Twitter
```

```

{
    public string OAuthConsumerSecret { get; set; }
    public string OAuthConsumerKey { get; set; }

    public async Task<IEnumerable<string>> GetTwitts(string userName,int count,
string accessToken = null)
    {
        if (accessToken == null)
        {
            accessToken = await GetAccessToken();
        }

        var requestUserTimeline = new HttpRequestMessage(HttpMethod.Get,
string.Format("https://api.twitter.com/1.1/statuses/user_timeline.json?count={0}&screen
_name={1}&trim_user=1&exclude_replies=1", count, userName));
        requestUserTimeline.Headers.Add("Authorization", "Bearer " + accessToken);
        var httpClient = new HttpClient();
        HttpResponseMessage responseUserTimeLine = await
httpClient.SendAsync(requestUserTimeline);
        var serializer = new JavaScriptSerializer();
        dynamic json = serializer.Deserialize<object>(await
responseUserTimeLine.Content.ReadAsStringAsync());
        var enumerableTwitts = (json as IEnumerable<dynamic>);

        if (enumerableTwitts == null)
        {
            return null;
        }
        return enumerableTwitts.Select(t => (string)(t["text"]).ToString());
    }

    public async Task<string> GetAccessToken()
    {
        var httpClient = new HttpClient();
        var request = new HttpRequestMessage(HttpMethod.Post,
"https://api.twitter.com/oauth2/token ");
        var customerInfo = Convert.ToBase64String(new
UTF8Encoding().GetBytes(OAuthConsumerKey + ":" + OAuthConsumerSecret));
        request.Headers.Add("Authorization", "Basic " + customerInfo);
        request.Content = new StringContent("grant_type=client_credentials",
Encoding.UTF8, "application/x-www-form-urlencoded");

        HttpResponseMessage response = await httpClient.SendAsync(request);

        string json = await response.Content.ReadAsStringAsync();
    }
}

```

```

        var serializer = new JavaScriptSerializer();
        dynamic item = serializer.Deserialize<object>(json);
        return item["access_token"];
    }
}

using System;
using System.Collections.Generic;
using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
namespace medical
{
    class Class1
    {
        SqlConnection con;
        SqlDataReader rs;
        string conn = "Data Source=SHRUHI;Initial Catalog=sentiment;Integrated
Security=True";
        int b,a,c;
        public int value1(string p_name, string d_name)
        {
            try
            {
                con = new SqlConnection(conn);
                //string s1 = "insert into user1 values('" + TextBox1.Text + "','" +
                TextBox2.Text + "','" + TextBox3.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','"
                + DropDownList1.Text + "')";
                con.Open();
                //Checks the number of times the same review is posted by same patient
                on same drug.
                string s1 = "select count(*) from temp where UserName='" + p_name + "'
                and ProductName='" + d_name + "'";
                SqlCommand cmd = new SqlCommand(s1, con);

                a = Convert.ToInt32(cmd.ExecuteScalar());
            }
            catch(Exception ex)
            {

            }
            return a;
        }
    }
}

```

```

public int value2(string p_name, string review)
{
    try
    {
        con = new SqlConnection(conn);
        //string s1 = "insert into user1 values('" + TextBox1.Text + "','" +
        TextBox2.Text + "','" + TextBox3.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','"
        + DropDownList1.Text + "')";
        con.Open();
        //Checks the number of reviews posted by a patient on a speciic drug.
        string s1 = "select count(*) from temp where UserName='" + p_name + "'
and Comments='" + review + "'";
        SqlCommand cmd = new SqlCommand(s1, con);
        b = Convert.ToInt32(cmd.ExecuteScalar());

    }
    catch (Exception ex)
    {

    }
    return b;
}
public int value3(string review)
{

```

```

    //string s1 = "select count(*) from temp where DrugName='" + d_name + "' and
    Comments='" + review + "'";
    try
    {
        string s1 = "select count(*) from temp where Comments='" + review + "'";
        SqlCommand cmd = new SqlCommand(s1, con);
        c = Convert.ToInt32(cmd.ExecuteScalar());
    }
    catch(Exception ex)
    {

    }
    return c;
}
public int value4(string p_name)
{

```

```

    string s1 = "select count(*) from temp where ProductName='" + p_name + "' and
    status='Positive' ";

```

```

        SqlCommand cmd = new SqlCommand(s1, con);
        int c = Convert.ToInt32(cmd.ExecuteScalar());
        return c;
    }
    public int value5(string p_name)
    {
        string s1 = "select count(*) from temp where ProductName='" + p_name + "' and
status='Negative' ";
        SqlCommand cmd = new SqlCommand(s1, con);
        int c = Convert.ToInt32(cmd.ExecuteScalar());
        return c;
    }
    public int value6()
    {
        //string s1 = "select count(*) from temp where DrugName='" + d_name + "' and
Comments='" + review + "'";
        try
        {
            string s1 = "select count(*) from temp ";
            SqlCommand cmd = new SqlCommand(s1, con);
            c = Convert.ToInt32(cmd.ExecuteScalar());
        }
        catch (Exception ex)
        {
        }
        return c;
    }
    public int value7()
    {
        //string s1 = "select count(*) from temp where DrugName='" + d_name + "' and
Comments='" + review + "'";
        try
        {
            string s1 = "select count(*) from review ";
            SqlCommand cmd = new SqlCommand(s1, con);
            c = Convert.ToInt32(cmd.ExecuteScalar());
        }
        catch (Exception ex)
        {

```

```

    }
    return c;
}
public int value8(string p_name)
{
    string s1 = "select count(*) from temp where ProductName='" + p_name + "' and
status='Neutral' ";
    SqlCommand cmd = new SqlCommand(s1, con);
    int c = Convert.ToInt32(cmd.ExecuteScalar());
    return c;
}
public void insert(string d_name, string p_name, string review)
{
    try
    {
        string s1 = "insert into temp values('" + d_name + "', '" + p_name + "', '" +
review + "', 'null')";
        SqlCommand cmd = new SqlCommand(s1, con);
        cmd.ExecuteNonQuery();
    }
    catch (Exception ex)
    {
    }
}

public int value9(string p_name)
{
    string s1 = "select count(*) from temp1 where Comments='" + p_name + "'
and status='Negative' ";
    SqlCommand cmd = new SqlCommand(s1, con);
    int c = Convert.ToInt32(cmd.ExecuteScalar());
    return c;
}

}
}

using System;
using System.Collections.Generic;

```

```

using System.Linq;
using System.Text;
using System.Threading.Tasks;
using System.Data.SqlClient;
using System.Collections;

using System.Text.RegularExpressions;
namespace medical
{
    class Class2
    {
        string id, usnam, seo, uid, finalseo, s;
        string pn;
        int eid, pcount = 0, ncount = 0;
        SqlConnection con;
        SqlDataReader rs;
        string conn = "Data Source=SHRUHI;Initial Catalog=sentiment;Integrated
Security=True";
        int b1, c;
        StopwordTool st = new StopwordTool();
        public void extract(string p_name, string tx)
        {
            StringBuilder sb = new StringBuilder();
            StringBuilder sb1 = new StringBuilder();
            pcount = 0;
            ncount = 0;
            string s4 = tx;
            //string s2 = s4.Trim(new Char[] { ' ', '@', '.', ',' });

            string text = s4.Replace(":", " ").Replace(":-(", " ").Replace(":-)", " ").Replace(";", " ").Replace(":/", " ").Replace(":-D", " ").Replace("=D", " ").Replace("=", " ").Replace("S", " ");
            string splitby = " ";
            Regex r = new Regex(splitby);
            string[] s = r.Split(text);
            //TextBox2.Text = s[0].ToString();
            //TextBox3.Text = s[1].ToString();
            //TextBox4.Text = s[2].ToString();
            int m = s.Count();
            string[] p = {"good", "nice", "safe", "effective", "excellent", "supportive", "relief", "relieve", "prefer", "support", "awesome", "lifesaver", "positive", "fortune", "correct", "superior", "brilliant", "wonderful", "marvelous", "miracle", "miraculous", "incredible", "healing", "best", "better",

```

"useful", "well", "helps", "controls", "works",
"controls", "beneficial", "energy", "energetic", "healthy", "improves", "speeds", "recovery",
"solves",
"resolves",
"redeems", "normal", "agree", "appreciate", "benefits", "comfort", "comfortable", "ease", "favorable", "helpful", "hoping", "honest", "hopes",
"improves", "like", "positive", "success", "support", "thanks", "use", "wonderful", "recovers", "solves", "preserves", "feasible", "fortunate", };

```
int n = p.Count();  
string[] ne = { "palliative", "unsafe", "ineffective", "loss", "lose", "lost", "not", "kills",  
"anxiety", "poison", "panic", "attacks", "pain", "nothing", "dizzy",  
"depressed", "suicidal", "hallucination", "vomiting", "nausea",  
"lymphedema", "fatigue", "anaemia", "tired", "disorders", "mental", "anxious",  
"infections", "fear", "depression", "infertility", "infertile", "paralyze",  
"side-effects", "bad", "fatal", "death", "toxic", "addictive", "intense", "problem",  
"uncontrollable", "sleepy", "sucks", "jittery", "weakness",  
"sleeplessness", "dangerous", "danger", "weight", "thinning", "adverse", "accident",  
"constraint", "risk", "risky", "perilous",  
"error", "confusion", "failure", "allergic", "iatrogenic", "lapses", "mistake", "negligence", "recklessness", "slow", "sentinel", "slip",
```

"violation", "frustration", "anger", "destroys", "spoils", "never", "depreciate", "decease", "deforms", "defeats", "fails", "not", "discomfort", "severe", "sleepless",

"terrific", "terrible", "horrific", "horrible", "afraid", "vomit", "fatigue", "fails", "kills", "died", "cannot", "cant", "concern", "concerns", "damage", "difficult", "deteriorates",

"didn't", "discomfort", "doubt", "Not", "not", "error", "hate", "hurt", "lack", "poor", "scary", "unfortunately", "wrong", "negative", "affects", };

```
b1 = ne.Count();  
int k = 0;  
con = new SqlConnection(conn);  
string s1 = "insert into user1 values('" + TextBox1.Text + "','" + TextBox2.Text + "','" +  
TextBox3.Text + "','" + TextBox4.Text + "','" + TextBox5.Text + "','" + DropDownList1.Text  
+ "')";  
con.Open();
```

```
for (int i = 0; i < m; i++)  
{  
    string s5 = s[i].ToString();  
    string a1 = s5.Trim(new Char[] { ' ', '@', '.', '?', '!', ':', '"', ';', ',' });  
    string a = a1.ToLower();
```

```

for (int j = 0; j < n; j++)
{
    string b2 = p[j].ToString();
    string b = b2.ToLower();
    int cmp = System.String.Compare(a, b);
    if (cmp == 0)
    {
        pcount++;
        sb.Append(s[i].ToString());
        //TextBox5.Text = "Positive";
    }
}
}

```

```

for (int i = 0; i < m; i++)
{
    string s5 = s[i].ToString();
    string a1 = s5.Trim(new Char[] { ' ', '@', '!', '?', '!', ':', '"', ';', ',' });
    string a = a1.ToLower();
    for (int j = 0; j < b1; j++)
    {
        string b2 = ne[j].ToString();
        string b = b2.ToLower();
        int cmp = System.String.Compare(a, b);
        if (cmp == 0)
        {
            ncount++;
            sb1.Append(s[i].ToString());
            //TextBox5.Text = "Positive";
        }
    }
}

```

```

if (pcount < ncount )
{

```

```

    seo = "Negative";

```

```

    string s1 = "update temp set status='Negative' where UserName='" + p_name
+ "'";

```

```

    SqlCommand cmd = new SqlCommand(s1, con);

```

```

        cmd.ExecuteNonQuery();
    }
    else if(pcount == ncount )
    {
        seo = "Neutral";

        string s1 = "update temp set status='Neutral' where UserName=" + p_name +
        """;

        SqlCommand cmd = new SqlCommand(s1, con);
        cmd.ExecuteNonQuery();

    }
    else
    {

        seo = "Positive";

        string s1 = "update temp set status='Positive' where UserName=" + p_name
        + """;

        SqlCommand cmd = new SqlCommand(s1, con);
        cmd.ExecuteNonQuery();
    }
}
}
}

```