A PROJECT REPORT

ON

"NEWS VALIDATION SYSTEM"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

SHAIKH MOHD NOORALAM MOHD KHAIRULALAM BILKEESH	**14CO48**
PAWASKAR SUFIYAN SIRAJ SULTANA	**14CO37**
RANE PRADNYESH NANDKISHOR NAMRATA	**14CO41**

UNDER THE GUIDANCE OF
PROF. TABREZ KHAN

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2017-2018

AFFILIATED TO

UNIVERSITY OF MUMBAI

A PROJECT II REPORT
ON

"NEWS VALIDATION SYSTEM"

Submitted to

UNIVERSITY OF MUMBAI

In Partial Fulfilment of the Requirement for the Award of

BACHELOR'S DEGREE IN
COMPUTER ENGINEERING

BY

SHAIKH MOHD NOORALAM MOHD KHAIRULALAM BILKEESH	**14CO48**
PAWASKAR SUFIYAN SIRAJ SULTANA	**14CO37**
RANE PRADNYESH NANDKISHORE NAMRATA	**14CO41**

UNDER THE GUIDANCE OF
PROF. TABREZ KHAN

DEPARTMENT OF COMPUTER ENGINEERING
Anjuman-I-Islam's Kalsekar Technical Campus
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

2017-2018
AFFILIATED TO

UNIVERSITY OF MUMBAI

Anjuman-i-Islam's Kalsekar Technical Campus

Department of Computer Engineering
SCHOOL OF ENGINEERING & TECHNOLOGY
Plot No. 2 3, Sector - 16, Near Thana Naka,
Khandagaon, New Panvel - 410206

CERTIFICATE

This is certify that the project entitled

"News Validation System"

submitted by

SHAIKH MOHD NOORALAM MOHD KHAIRULALAM BILKEESH	**14CO48**
PAWASKAR SUFIYAN SIRAJ SULTANA	**14CO37**
RANE PRADNYESH NANDKISHOR NAMRATA	**14CO41**

is a record of bonafide work carried out by them, in the partial fulfilment of the requirement for the award of Degree of Bachelor of Engineering (Computer Engineering) at ***Anjuman-I-Islam's Kalsekar Technical Campus, Navi Mumbai*** under the University of MUMBAI. This work is done during year 2017-2018, under our guidance.

Date: / /

Prof. TABREZ KHAN
Project Supervisor

Prof. KALPANA BODKE
Project Coordinator

Prof. TABREZ KHAN
HOD, Computer Department

DR. ABDUL RAZAK HONNUTAGI
Director

External Examiner

Acknowledgements

We would like to take the opportunity to express my sincere thanks to my guide **TABREZ KHAN**, Professor, Department of Computer Engineering, AIKTC , Panvel for his invaluable support and guidance throughout my project research work. Without his kind guidance & support this was not possible.

We are grateful to him for his timely feedback which helped me track and schedule the process effectively. His time, ideas and encouragement that he gave is help me to complete my project efficiently.

We would like to express deepest appreciation towards **DR. ABDUL RAZAK HONNUTAGI**, Director, AIKTC, Navi Mumbai, **Prof. TABREZ KHAN**, Head of Department of Computer Engineering and **Prof. KALPANA BODKE**, Project Coordinator whose invaluable guidance supported us in completing this project.

At last we must express our sincere heartfelt gratitude to all the staff members of Computer Engineering Department who helped me directly or indirectly during this course of work.

SHAIKH MOHD NOORALAM MOHD KHAIRULALAM BILKEESH

PAWASKAR SUFIYAN SIRAJ SULTANA

RANE PRADNYESH NANDKISHOR NAMRATA

Project I Approval for Bachelor of Engineering

This project entitled ***News Validation System*** by ***SHAIKH MOHD NOORALAM MOHD KHAIRULALAM BILKEESH, PAWASKAR SUFIYAN SIRAJ SULTANA, RANE PRADNYESH NANDKISHOR NAMRATA*** is approved for the degree of ***Bachelor of Engineering in Department of Computer Engineering.***

Examiners

1. .....
2. .....

Supervisors

1. .....
2. .....

Chairman

.....

Declaration

We declare that this written submission represents my ideas in my own words and where others ideas or words have been included, We have adequately cited and referenced the original sources. We also declare that we have adhered to all principles of academic honesty and integrity and have not misrepresented or fabricated or falsified any idea/data/fact/source in my submission. We understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources which have thus not been properly cited or from whom proper permission has not been taken when needed.

SHAIKH MOHD NOORALAM MOHD KHAIRULALAM BILKEESH
14CO48

PAWASKAR SUFIYAN SIRAJ SULTANA
14CO37

RANE PRADNYESH NANDKISHOR NAMRATA
14CO41

ABSTRACT

Title: News Validation System

The news is an information of particular interest that was not known in the past, the information implied by news should be accurate and reliable. There are millions of news shared over social media each day without validation. Currently, there exists no resource that could validate whether any news article is giving valid information or not. Here comes the existence of News Validation System. There are various news analysis methods that would validate news. Our news validation system implements methods that would be more feasible, accurate and reliable out of all methods present for news validation. Our system will go with a hybrid approach. A hybrid approach is the combination of both linguistic and network approach, this approach is developed by integrating advantages of linguistic and network approach and by eliminating their drawbacks. It has components like authentic source identification, date validation, data collection, comparison and support analysis. All these parameters will conclude in well formed results. The system will feed upon a normal text of information in the context of news articles, the output of the system will be a probability percentage, that would tell at which extent the information is reliable and valid. Our work in news validation system can help cut down potential misleading information in the news and we can make sure that everyone gets reliable, dependable, authentic and accurate information.

Keywords: Validation, Analysis, Natural Language Processing,

Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Statement of the Project

News is nothing but information about current events. The arrival of the web and the social web brings with it a tremendous number of new sources. The accessibility of these news sources generates a large amount of information which can often times be contradicting and confusing. Facebook, for example, can be seen as a social platform that allows individuals and groups of individuals to freely exchange thoughts and opinions.

When this information travels the social web, it is difficult to distinguish between valid and unsupported news. News verification aims to implement a technology that can identify fake news. Fake news detection is defined as a news which is intentionally altered. The problem of fake news detection is more challenging and complicated task than detecting deceptive news, since the political language on TV interviews, post on Facebook and Twitter are mostly short statements

There are various approaches that can be used to develop news validation or fake news detection system. Majorly there are two types of approaches, linguistic approach and network approach. In Linguistic approach, some liar uses their language skill to avoid being caught guilty. There is some leakage of words from which we can identify that whether they are saying truth or not.

The goal in the linguistic approach is to look for such words or leakages. Network approach is innovative and varied, using network properties and behavior are ways to complement content- based approaches that rely on deceptive language and leakage cues to predict deception. Hybrid approach is the combination of both network approach as well as linguistic approach. In our system we are going to use hybrid approach because individually network or linguistic approach is not too accurate to increase efficiency and for better results we shall be using hybrid approach.

1.2 Purpose

In social media articles often decontextualised from source, fact can mix freely with fiction.The sharing of hoax news often results in defamation of certain entity, it plays with emotions of readers.Unverified news often causes loss of lives, initiates riots, loss in business.Often people share unverified news or text, without verifying the source and causes above problems.The content in news report are framed in such way that it provokes emotionally and gain reaction.It is high time to stop this uncontrolled flow of such news.

1.3 Project Scope

Our system will validate news at global level where it will only support text of the news. The system will help in filtering the fake news available on the internet and will serve quality news to the readers. This will increase the efficiency and reliability of the digital news and can be as trustworthy as the printed newspapers.

1.4 Project Goals and Objectives

1.4.1 Goals

The goal of the News Validation System is to explore how artificial intelligence technologies, particularly machine learning and natural language processing, might be leveraged to combat the fake news problem.

1.4.2 Objectives

The Objective of our system is to validate the news so that the user will get correct data.

We believe that these AI technologies hold promise for significantly automating parts of the procedure human fact checkers use today to determine if a story is real or a hoax.

1.5 Organization of Report

The remaining part of the project is organized as follows.

Chapter 2 presents a review of related work.

Chapter 3 describes the time management and time utilization during the project

implementation.

Chapter 4 introduces the Software requirement Specification of our project.

Chapter 5 proposes the project design of the project.It represent the architectural design,front end design and database design of the project.

Chapter 6 presents implementation details of our project.

Chapter 7 presents various test cases that are consider.

Chapter 8 consist various screenshots of project.

Chapter 9 provides some concluding remarks and direction of our future work.

Chapter 2

Literature Survey

2.1 Fake News or Truth? Using Satirical Cues to Detect Potentially Misleading News.

2.1.1 Description

This paper's authors are Niall J. Conroy, Victoria L. Rubin, Yimin Chen was published in the Proceedings of the Workshop on Computational Approaches to Deception Detection at the 15th Annual Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL-CADD2016) on June 17, 2016 in San Diego, California.

This research paper enables the identification of deliberately deceptive misinformation in the institutional mainstream or non-institutional text-based online news. The resulting deception detection methodology will allow making predictions about each previously unseen news piece: is it likely to belong to the truthful or deceptive category? A system, based on this methodology, will alert users to most likely deceptive news in the incoming stream of news, and prompt the users to fact-check further.

Digital deception is a deliberate effort to create false beliefs or conclusions in technology- mediated environments. This research project focuses on deliberate misinformation in text- based online news, provided via mainstream media and citizen journalist websites, news archives and aggregators. Various deception types and degrees will be examined, categorized, and mod- eled: fake or fabricated news, exaggerated claims, material fact omissions, indirect responses, question-dodging, and subject-changing.

Mistaking deceptive news for authentic reports can create costly negative consequences such as sudden stock fluctuations or reputation loss. Everyday life decision-making, behavior, and mood are influenced by news we receive. When professional

analysts sift through the news, their future forecasts, fact and pattern discovery depend on veracity of the news in "big data" knowledge management and curation areas (specifically, in business intelligence, financial and stock market analysis, or national security and law enforcement). In both lay and professional contexts of news consumption, it is critical to distinguish truthful reports from deceptive ones. However, few news verification mechanisms currently exist, and the sheer volume of the infor- mation requires novel automated approaches.

Mistaking deceptive news for authentic reports can create costly negative consequences such as sudden stock fluctuations or reputation loss. Everyday life decision-making, behavior, and mood are influenced by news we receive. When professional analysts sift through the news, their future forecasts, fact and pattern discovery depend on veracity of the news in "big data" knowledge management and curation areas (specifically, in business intelligence, financial and stock market analysis, or national security and law enforcement). In both lay and professional contexts of news consumption, it is critical to distinguish truthful reports from deceptive ones. However, few news verification mechanisms currently exist, and the sheer volume of the infor- mation requires novel automated approaches.

News verification methods and tools are timely and beneficial to both lay and professional text- based news consumers.

2.1.2 Advantages of Paper

The research significance is four-fold:

a. Automatic analytical methods complement and enhance the notoriously poor human ability to discern information from misinformation.

b. Credibility assessment of digital news sources is improved.

c. The mere awareness of potential digital deception constitutes part of new media literacy and can prevent undesirable consequences.

d. The proposed veracity/deception criterion is also seen as a metric for information quality assessment.

2.1.3 Disadvantages of Paper

a. This method works in the context of textual news information only.

2.1.4 How to overcome the problems mentioned in Paper

a. To target other than textual means enabling support for images as well, the images may more influence than text, as it may involve visual scenes which may be obscene in nature, or not usual for normal user, indirectly it can play with reader's intellect.

b. The images involved in the form of images can be scraped by using Google's Image Vision API, and then from the results we can extract origin, date of the image, as in fake news the image must have been used from past incident or any other incident must be framed as recent. Also, the description of searched results can be compared with the input news.

2.2 Automatic Deception Detection - Methods for Finding Fake News

2.2.1 Description

This paper's authors are Niall J. Conroy, Victoria L. Rubin, Yimin Chen was published by ASIST(Association for Information Science and Technology) on November 6-10, 2015 in St. Louis, MO, USA

News verification aims to employ technology to identify intentionally deceptive news content online, and is an important issue within certain streams of library and information science.

This paper provides researchers with a map of the current landscape of veracity deception assessment methods, their major classes and goals, all with the aim of proposing a hybrid approach to system design.These methods have emerged from separate development streams, utilizing disparate techniques. In this survey, two major categories of methods emerge: 1. Linguistic Approaches in which the content of deceptive messages is extracted and analyzed to associate language patterns with deception; and 2. Network Approaches in which network information, such as message metadata or structured knowledge network queries can be harnessed to provide aggregate deception measures.

Both forms typically incorporate machine learning techniques for training classifiers to suit the analysis. It is incumbent upon researchers to understand these different areas, yet no known typology of methods exists in the current literature. The goal is to provide a survey of the existing research while proposing a hybrid approach, which utilizes the most effective deception detection methods for the implementation of a fake news detection tool.

Structured datasets are easier to verify than non-structured (or semi-structured) data such as texts.When we know the language domain (e.g., insurance claims or health-related news) we can make better guesses about the nature and use of deception. Semi-structured non-domain specific web data come in many formats and demand flexible methods For veracity verification.

2.2.2 Advantages of Paper

a. Success was measured based on whether the machine was able to assign higher true values to true statements than to false ones.

b. Linguistic and network-based approaches have shown high accuracy results in classification tasks within limited domains.

2.2.3 Disadvantages of Paper

a. A problem with this method is that statements must reside in a preexisting knowledge base.

2.2.4 How to overcome the problems mentioned in Paper

By using Hybrid approach that is combination of linguistic and network approach we can overcome the above mention problem.

2.3 Developing a News Aggregation and Validation System.

2.3.1 Description

Fake news detection is defined as the prediction of the chances of a particular news article (news report, editorial, expose, etc.) being intentionally deceptive. The Linguistic Approach can be described as a method where the content of an item gets extracted and analyzed regarding language patterns. sometimes a ”leakage” occurs, meaning that a break in the pattern is observable. Thus the following aspects need to be discussed stop words, stemming,lemmatization. Some words do not add value to the context of a text, thus those words do not need to be processed by the system.

Those words are described as common terms or stop words. Nonetheless, it is also stated that in some circumstances the use of stop words are essential in order to keep the context. Articles and other forms of text use different forms of a word (e.g. conjunctions) due to grammatical reasons. Examples, therefore, are ’am, are, is,’ which are different forms of ’be’ and ’cars, cars’, car’s’ which results in car. The process of Stemming connotes that the affixes are removed from a word, meaning the ends of words are cut off, potentially resulting in the base form of a word. Lemmatization

on the other hand uses 'vocabulary and morphological analysis of words in order to reach the goal of finding the base of a word.

2.3.2 Advantages of Paper

a. It is very important subset of Hybrid Approach and can be collectively used with other methods to detect fake news very accurately.

2.3.3 Disadvantages of Paper

a. Linguistic Approach do not validate on the basis of real time news, it just focus on linguistic leakages in the textual news.

b. Linguistic Approach do not provide accuracy if used independently with other method.

2.3.4 How to overcome the problems mentioned in Paper

Major problem with linguistic method was it only focus on linguistic leakages and not on realtime information related to that particular news and network method only focus on information about that news not about linguistic leakages so if both this method is combined as hybrid approach then major disadvantages for linguistic method can be overcomed and accuracy can also be obtained for news validation.

2.4 Technical Review

The technology that we are using here in Python such as textblob, rake nltk, goose, newsapi.

2.4.1 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.Its is most widely used in our project like in support check module for analysis sentiments of tweets extracted.

2.4.2 Advantages of Technology

a. Its very efficient

b. Since, it is built on the shoulders of NLTK and Pattern, therefore making it simple for beginners by providing an intuitive interface to NLTK.

c. It provides language translation and detection which is powered by Google Translate (not provided with Spacy).

2.4.3 Reasons to use this Technology

a. It is very well designed, fast and scalable.

b. Other library in python for do the same thing are not reliable

c. It gives result in string format that can be easily handle.

2.4.4 News API

News API is a simple HTTP REST API for searching and retrieving live articles from all over the web. Its gives the data from over 30,000 news sources.Its is freely available we just have to register on it once.After registration we get access to 1000 request per day.

2.4.5 Reasons to use this technology

a. It returns JSON metadata we can be easly use.

b. It response time is good.

c. Huge number of data sources that is over 30,000 news sources.

2.4.6 Rake NLTK

RAKE short for Rapid Automatic Keyword Extraction algorithm, is a domain independent keyword extraction algorithm which tries to determine key phrases in a body of text by analyzing the frequency of word appearance and its co-occurance with other words in the text.

2.4.7 Reasons to use this technology

a. As its name suggest its extract the keyword from the sentence given very rapidly.

b. It is efficient way to extract keyword till the date.

Chapter 3

Project Planning

3.1 Members and Capabilities

Table 3.1: Table of Capabilities

SR. No	Name of Member	Capabilities
1	Nooralam Shaikh	UI Design
2	Sufiyan Pawaskar	UI Design , Database
3	Pradnyesh Rane	Database

Work Breakdown Structure

3.2 Roles and Responsibilities

Table 3.2: Table of Responsibilities

SR. No	Name of Member	Role	Responsibilities
1	Nooralam Shaikh	Team Leader	UI Design and core modules
2	Sufiyan Pawaskar	Team Member	UI Design ,Integration ,API design
3	Pradnyesh Rane	Team Member	Support check Module, Documentation

3.3 Assumptions and Constraints

Assumption

Assumption is that the data that is coming from the scrapper is in correct from and from genuine source.

Constraint

If we provide wrong training data to the system then prediction will be wrong. If news API from where we are scrapping the data goes down then the system will

fail.

3.4 Project Management Approach

Spiral Model is a combination of a waterfall model and iterative model. Each phase in spiral model begins with a design goal and ends with the client reviewing the progress. We have use this model for version control.

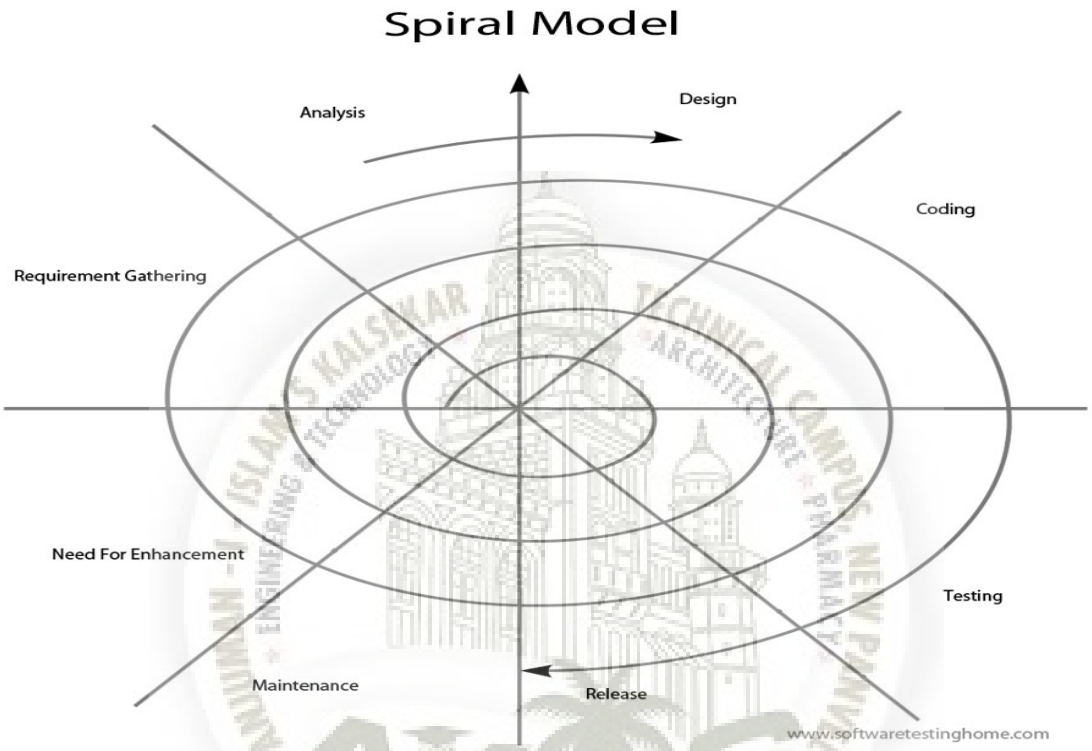

Figure 3.1: Spiral model

3.5 Ground Rules for the Project

1. We treat each other with respect.
2. We intend to develop personal relationships to enhance trust and open communication.
3. We value constructive feedback. We will avoid being defensive and give feedback in a constructive manner.
4. As team members, we will pitch in to help where necessary to help solve problems and catch-up on behind schedule work.
5. Additional meetings can be scheduled to discuss critical issues or tabled items upon discussion and agreement with the team leader.

6. One person talks at a time, there are no side discussions.

7. When we pose an issue or a problem, we will also try to present a solution.

3.6 Project Budget

The budget for this project is very low as most of the tools we have use are open source.Following are the budget for the project

1. Operating System: linux mint (Open Source).
2. Python Programming language (Open Source)
3. IDE:Andriod Studio (Open Source).
4. News API(Open Source)

3.7 Project Timeline

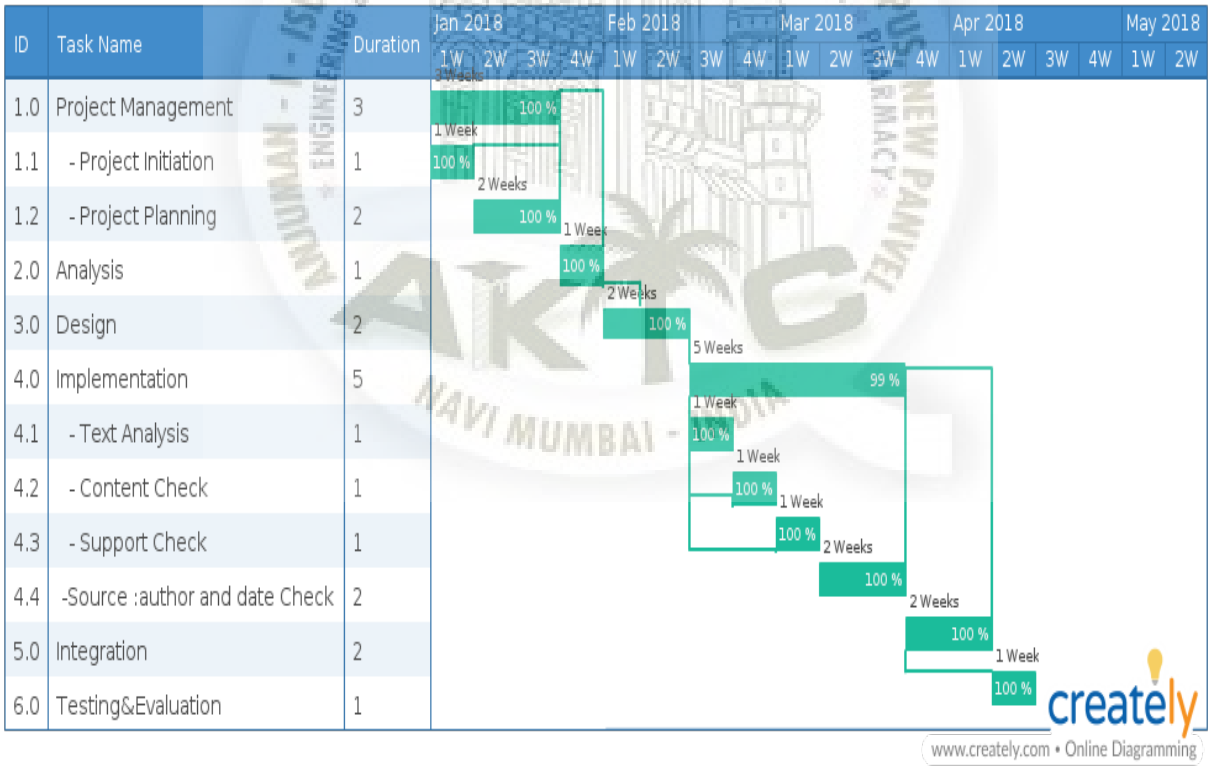

Figure 3.2: Gantt chart

Chapter 4

Software Requirements Specification

4.1 Overall Description

4.1.1 Product Perspective

The origin of our product or our system is the need for validation of the news. It is a web based system implementing client-server model. The News Validation System provides simple mechanism for users to validate the news data.

4.1.2 Product Features

News Validation System will provide api key to the client.Its will also validate the news given input to it showing the estimated value of the give data is true or false.

4.1.3 User Classes and Characteristics

There are various users that can use the system such as normal user i.e any day to day life user who wants to check the content of the news is correct or not, any third party which use our project as a sub module of theirs.

4.1.4 Operating Environment

Our System will provide both web based and app based support. One can use the website from theirs mobiles or computer they just required a normal web browser and also one can use mobiles application to do the same.

4.1.5 Design and Implementation Constraints

The main decision was to select that which language should we use for developing our system.Since most of the programming language are very time consuming when it comes to data processing . So the complete system is made by using python language which is very efficient for processing the data as compare to the other language which are available in market since its provide various library to do our work simple.

4.2 System Features

Our system will provide an application programming interface to our client using which they can have a idea whether the input news is correct or not. There are various features as mention below.

4.2.1 Scrapper

It is the one of the main feature of our system.It will scrap the given data from various news sources or website in java script object notation format.

Description and Priority

It's is key player in our system it will extract news from over 4000 news website and get its data in java script object notation format to us.It is high priority module. Its main benefits is that it scrap the data from large number of reliable websites.

4.2.2 Keyword Extractor

It will only extract the keyword from the data given to it from scrapper.

Description and Priority

It will extract the keyword from the given the data given from the scrapper and forward those key word to properties generator. It is medium priority module.Its extract the keyword using Rapid Automatic Keyword Extraction algorithm.

4.2.3 Properties Generator

It will generate the properties using the data or the keyword from the keyword extractor.

Description and Priority

It will generate the values based on properties and then forward this values to the classifier.It is high priority module.

Stimulus/Response Sequences

1. The User must register in our system.
2. After Registration user must login in our system using its login credentials.In response to login the system will provide a api key.
3. By using api key provided by the system user can input for the data to validate.

4. In response to enter data the system will predict whether the enter data is correct or false.And it will return a value.
5. In case if the data is correct it will display that which is the source of the data.

Functional Requirements

1. The user should register on the system User should login in the system.
2. The data input by the user must be in textual from.
3. The servers should response quickly.

4.3 External Interface Requirements

4.3.1 User Interfaces

1. User must first register itself on our system.
2. After registration user will get a API key through will he/she will be authenticated.
3. By using that key user can search for the data he/she needs to validate.

4.3.2 Hardware Interfaces

There is no special kind of hardware required to run this system, A regular desktop or any smart phone can be enough to access our system.

4.3.3 Software Interfaces

One must need a web browser or our mobile application to access our system. For database we used sequel database that is mysql. For developing core modules we used python programming language. We have used news API for extracting the required data. On the client side only a browser or our mobile application can be used.

4.3.4 Communications Interfaces

The major communication between the client or user with our system is done using web-browser or an mobile application provided by us.
The communication between the system and database is done by using mysql.

4.4 Nonfunctional Requirements

4.4.1 Performance Requirements

The main performance of this system depend on how much time it spends on extracting and processing the data. extracting the data from various news sources is very time consuming task but python made it simple by providing library. And also for processing the used language that is python is very efficient.

4.4.2 Safety Requirements

Once the client login in our system we will provide it a API key for authentication purpose.This API key is unique key given by our system to client.

4.4.3 Security Requirements

All the clients on the our system are authenticated. The data scrapped is also collected from verified sources only. API key is hashed when sent in network.No more security is required for the data since its not that much important if leaked.

Chapter 5

System Design

5.1 System Requirements Definition

System requirement definitions specify what the system should do, its functionality and its essential and desirable system properties. The techniques applied to elicit and collect information in order to create system specifications and requirement definitions involve consultations, interviews, requirements workshop with customers and end users. The objective of the requirements definition phase is to derive the two types of requirement:

5.1.1 Functional requirements

They define the basic functions that the system must provide and focus on the needs and goals of the end users.

Use-case Diagram

Use case diagrams are usually referred to as behavior diagrams used to describe a set of actions (use cases) that some system or systems (subject) should or can perform in collaboration with one or more external users of the system (actors). Each use case should provide some observable and valuable result to the actors or other stakeholders of the system.

Use case diagrams are in fact twofold they are both behavior diagrams, because they describe behavior of the system, and they are also structure diagrams as a special case of class diagrams where classifiers are restricted to be either actors or use cases related to each other with associations.

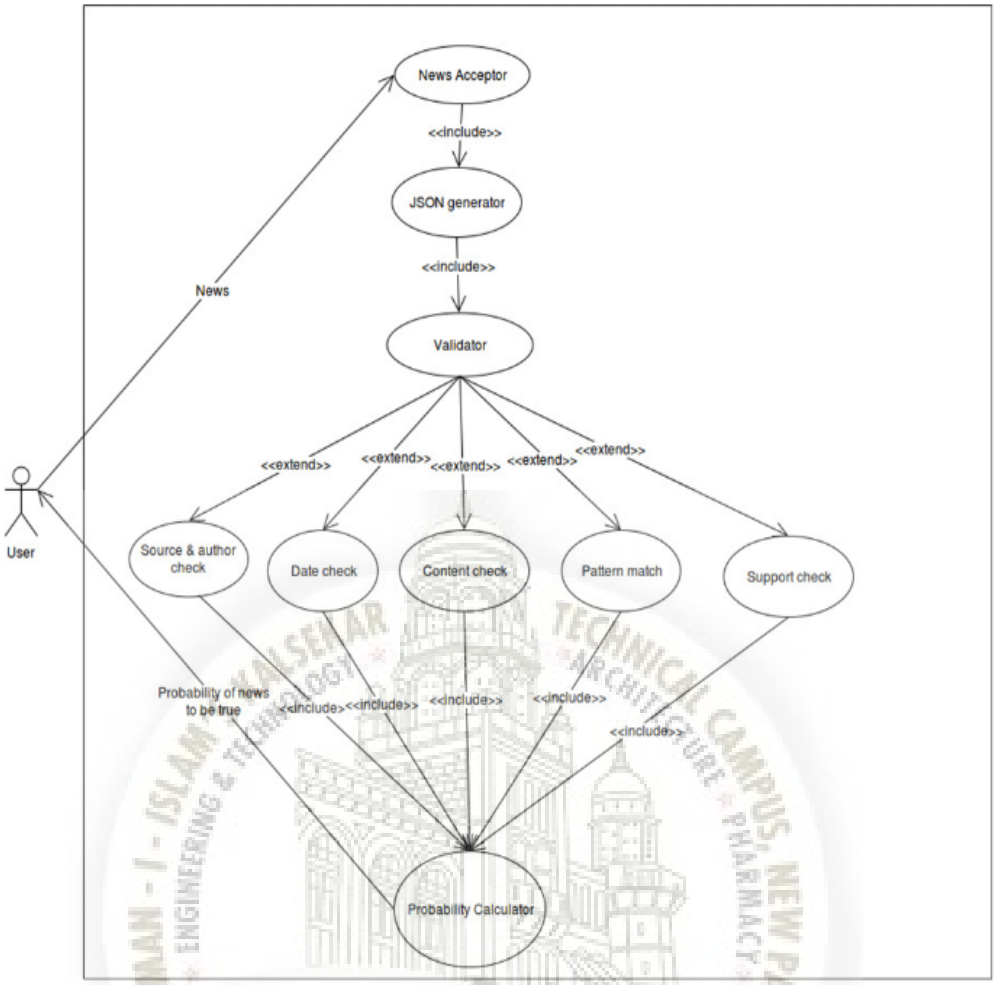

Figure 5.1: Use Case

Data-flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination. Data flowcharts can range from simple, even hand-drawn process overviews, to in-depth, multi-level DFDs that dig progressively deeper into how the data is handled. They can be used to analyze an existing system or model a new one. Like all the best diagrams and charts, a DFD can often visually "say" things that would be hard to explain in words, and they work for both technical and nontechnical audiences, from developer to CEO. That's why DFDs remain so popular after all these years. While they work well for data flow software and systems, they are less applicable nowadays to visualizing interactive, real-time or database-oriented software or systems.

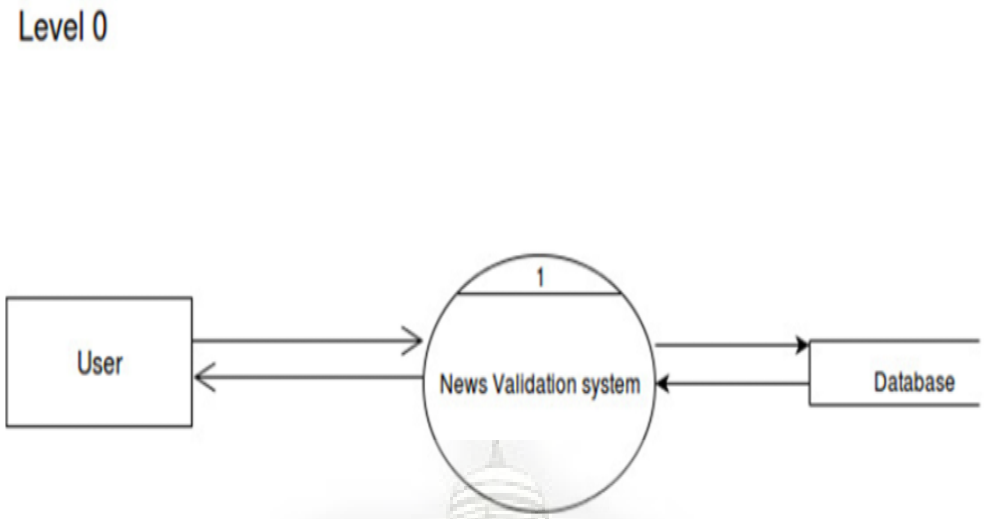

Figure 5.2: Level 0 dfd for News Validation System

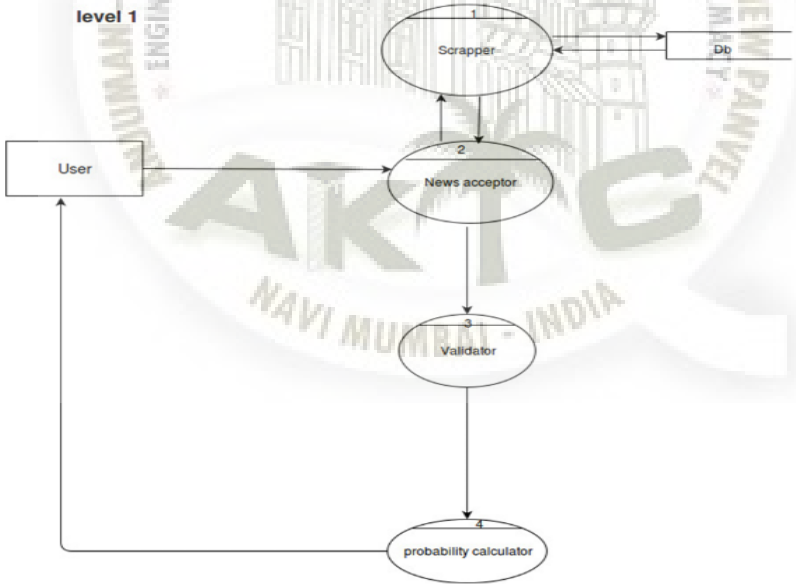

Figure 5.3: Level 1 dfd for News Validation System

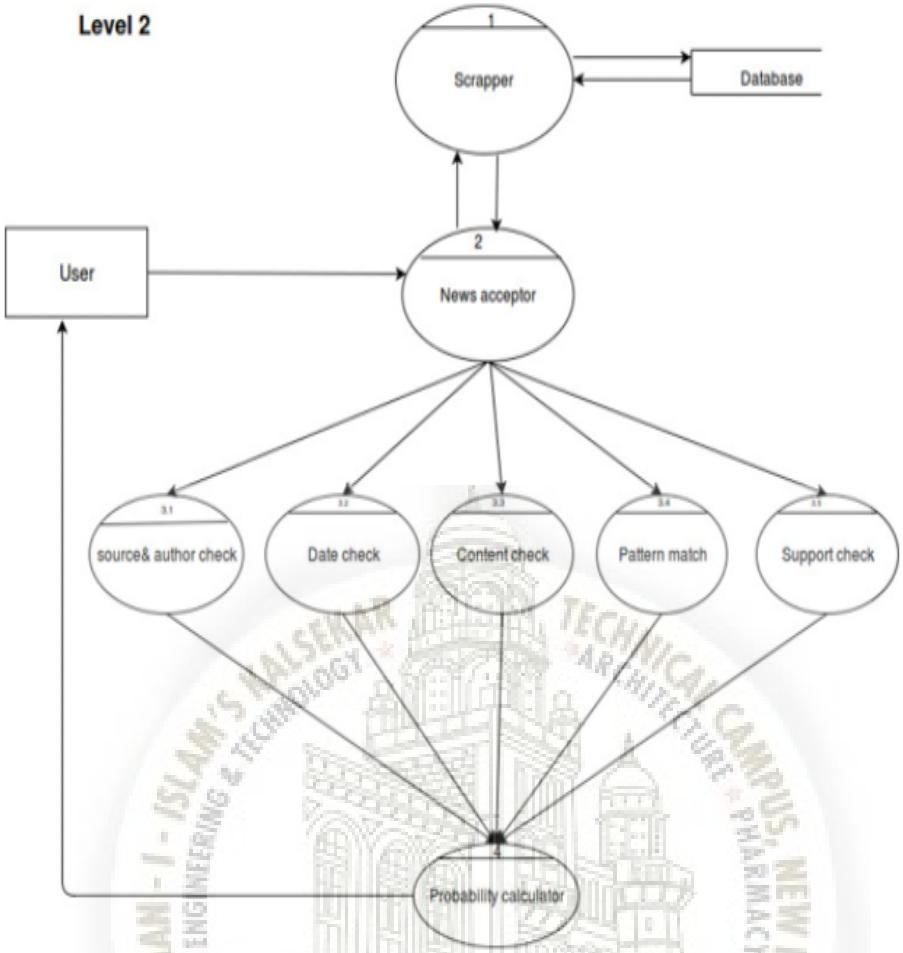

Figure 5.4: Level 2 dfd for News Validation System

5.1.2 System requirements (non-functional requirements)

Performance Requirements

The main performance of this system depend on how much time it spends on extracting and processing the data. Extracting the data from various news sources is very time consuming task but python made it simple by providing library. And also for processing the used language that is python is very efficient.

Safety Requirements

Once the news API server goes down then the whole system will go down.So to prevent from this we has to periodically check whether the server is up or not.

Security Requirements

The major security requirements for the system will be the safeguarding of the user data from any kind of exploit. In order to protect the user data the data is not stored in local databases we will be storing in the cloud for better security.And also the API key which is given to user once register for authentication is also encrypted.

Database Schema/ E-R Diagram

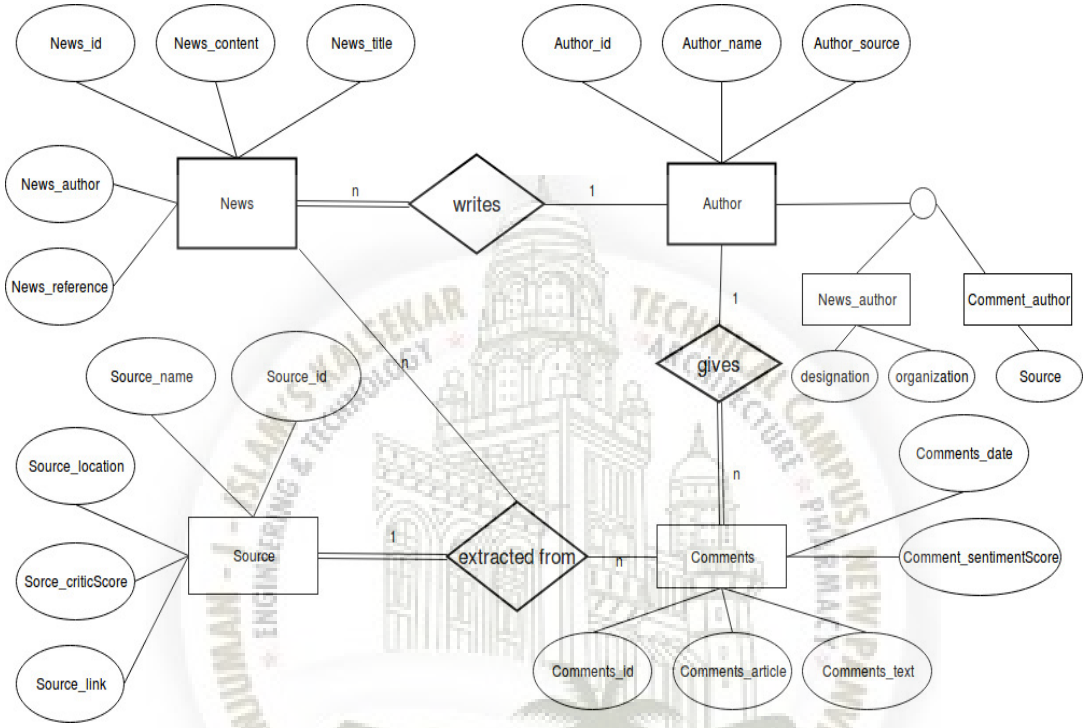

Figure 5.5: Entity Relationship for News Validation System

This is the E-R diagram of the system in which the modules which will be there after the deployment are shown .Its is shown in a very easy way to have a brief overview of project.

5.2 System Architecture Design

The system architecture is mainly divided in two sub modules, the first sub module consists of client side and the main module is news validation server.

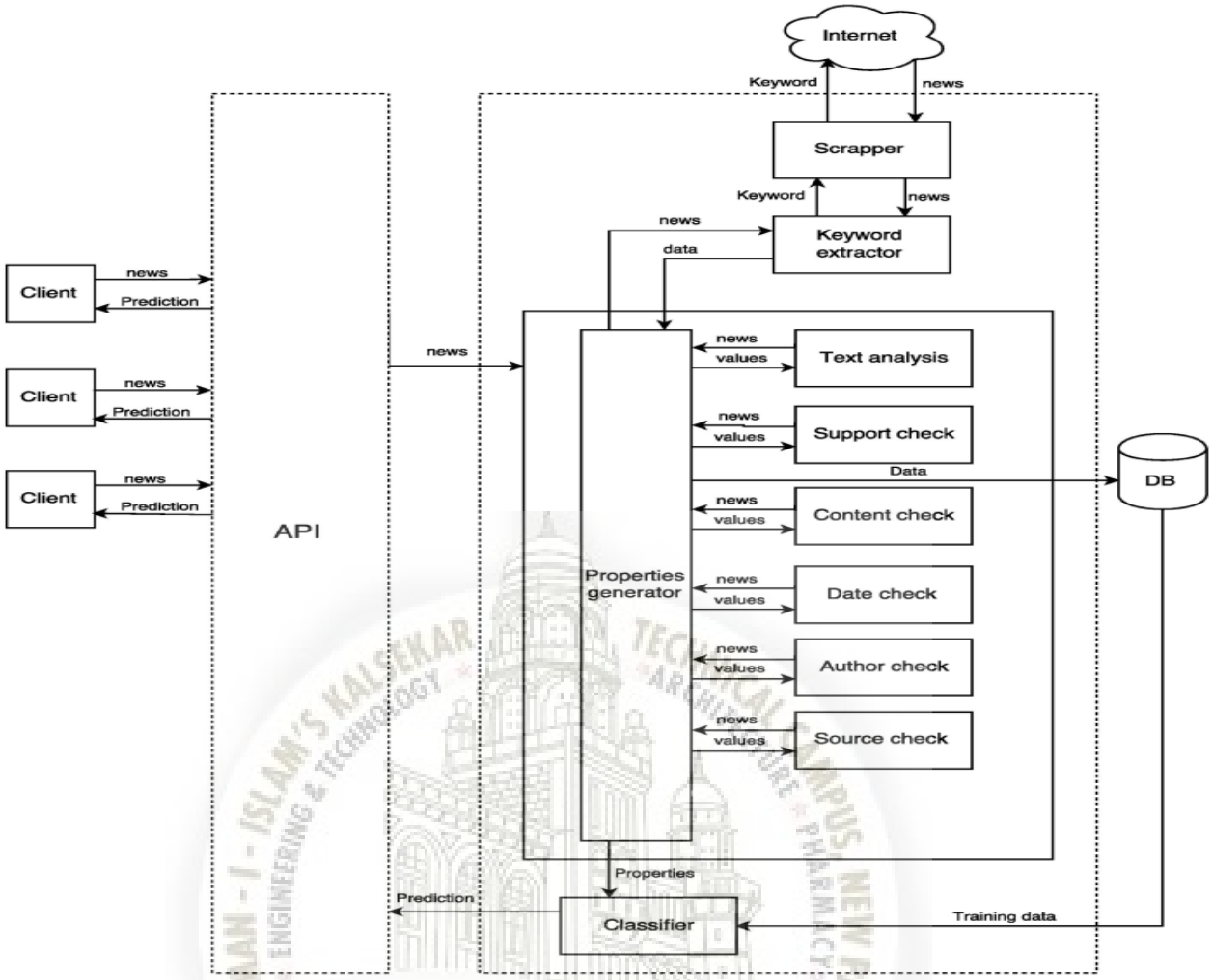

Figure 5.6: System Architecture for News Validation System

5.3 Sub-system Development

There are total four main modules in system architecture namely scrapper,keyword extractor,properties generator and classifier. scrapper will scrap the news from various news website using api ,keyword extractor will extract the keywords from the scrap data to process it,properties generator will generate a properties of news based on the extracted keywords,based on properties generated from properties generator and classifier the system will predict the prediction.

5.3.1 Authentic Source Identification

Data from authentic and authorized source plays very important role in news validation. Authenticity of source and author of training data set can increase the accuracy of results. This technique will be implemented in Source and author check sub-module in validator module of news validation system.

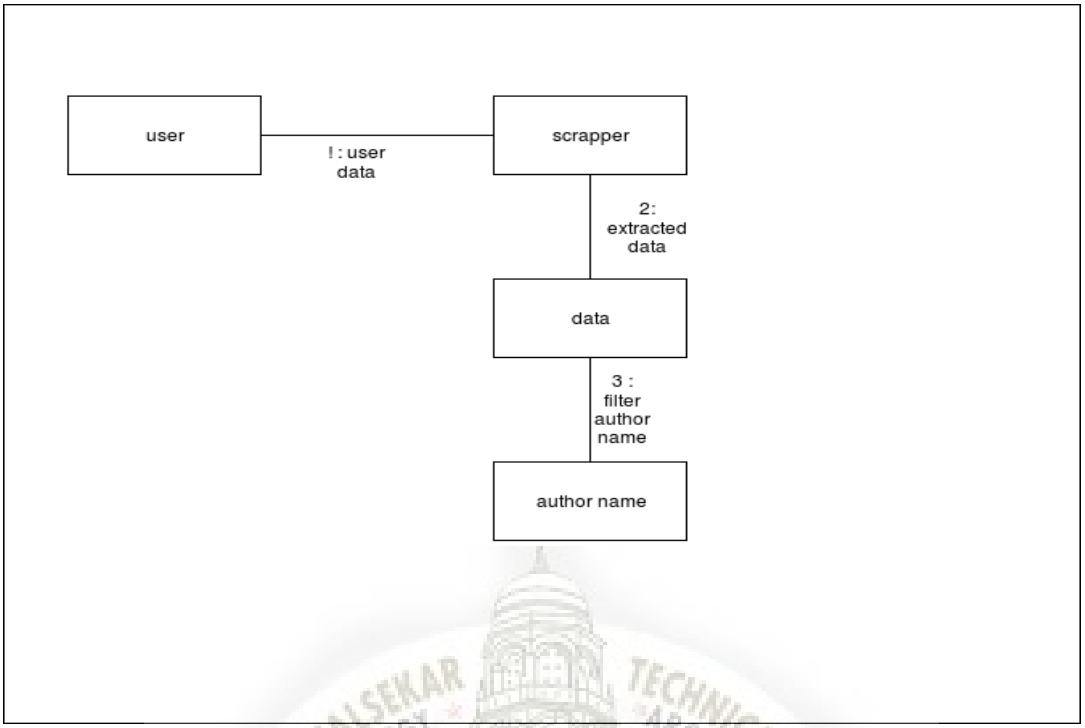

Figure 5.7: Modular diagram for authentic source identification

5.3.2 Date Validation

Sometimes it happens like old news get recreated and shared intentionally or some old news is shared without checking its date, so to tackle this problem date validation is very important. there are chances that some old news is updated by positive results but not mentioned in news, hence it is very important to check date. This technique will be implemented in date check sub-module in validator module of news validation system.

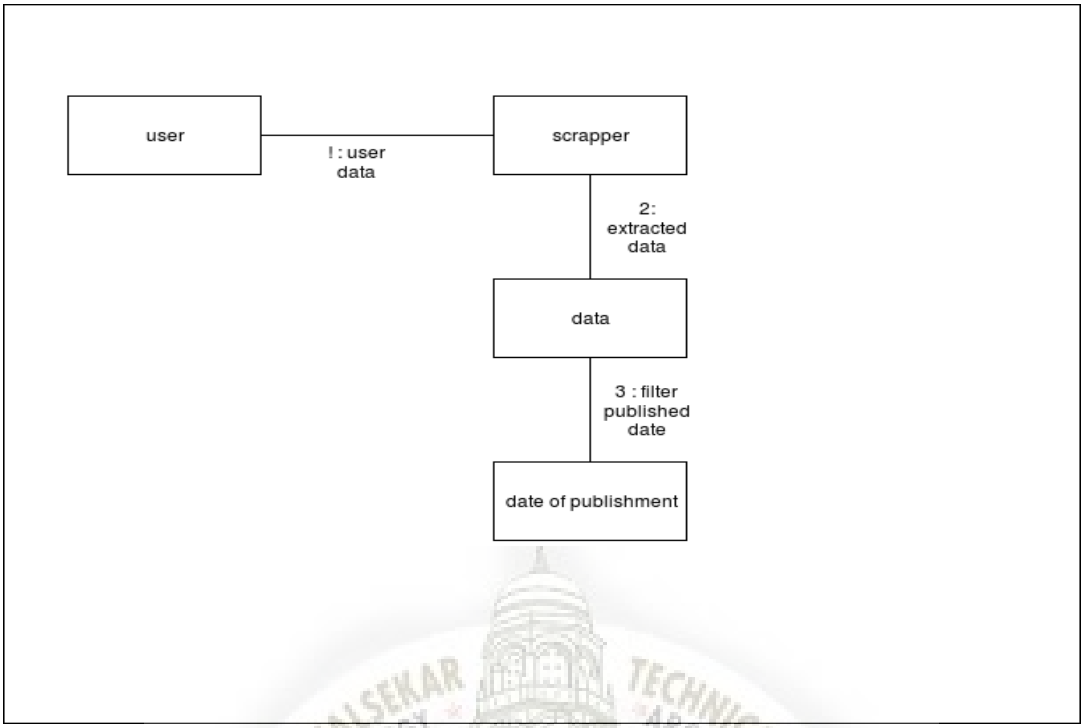

Figure 5.8: Modular diagram for date validation

5.3.3 Data collection and comparison

Collection of articles from authentic and genuine source and analyzing that plays a very important role in news validation. Data is collected based on keyword extracted from heading and paragraph of target news article and comparison is done between scrapped articles and target news articles. Matching rate of both articles is inversely proportional to chances of news to be fake, more match rate can result to less chance of news article to be fake. This technique will be implemented in content check and pattern match sub-module in validator module of news validation system. Grammatical leakage is also analyzed in this module.

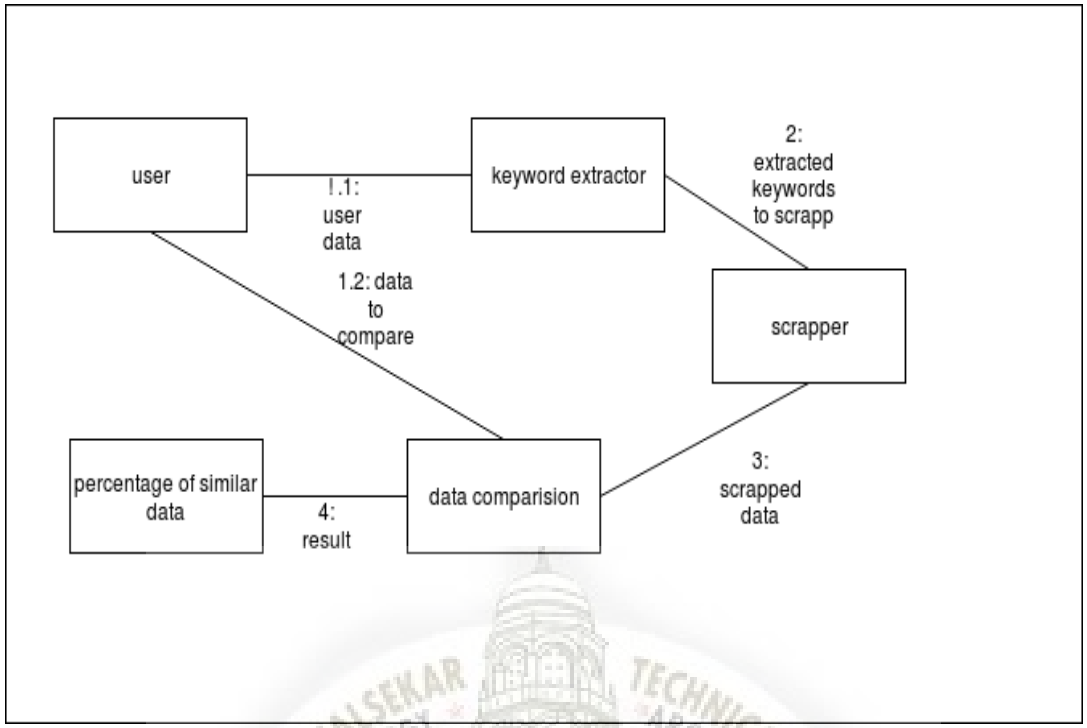

Figure 5.9: Modular diagram for data collection and comparision

5.3.4 Support Analysis

Reaction of users on a news also plays an important role in defining its validity. Reaction will be analyzed using sentimental analysis on comments of users on related news articles fetched from genuine news websites Design is the first step in the development phase for any techniques and principles for the purpose of defining a device, a process or system in sufficient detail to permit its physical realization. Once the software requirements have been analyzed and specified the software design, coding, implementation and testing that are required to build and verify the software.

The design activities are of main importance in this phase, because in this activity decision ultimately affecting the success of the software implementation and it ease of maintenance are made. These decision have the final bearing upon reliability and maintainability of the system. Design is the place where quality is fostered in development. Software design is the process through which requirements are translated into a representation of software. Software design is conducted in two steps.

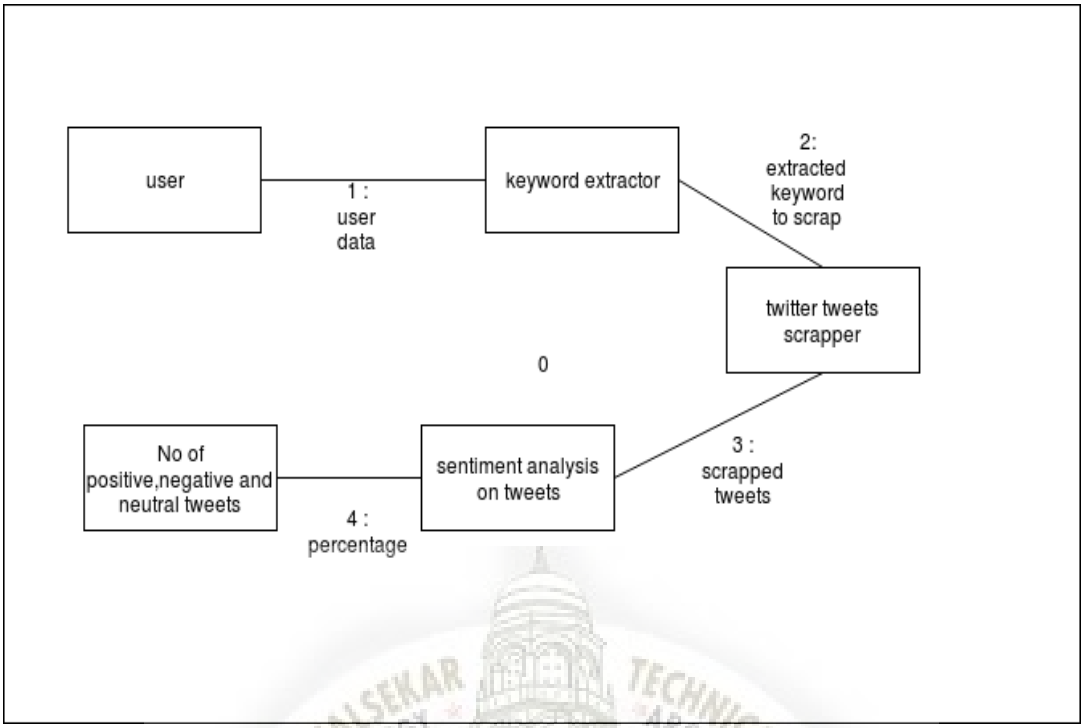

Figure 5.10: Modular diagram for support analysis

5.4 Systems Integration

There are mainly five main module in our system.Scrapper this module extract the data from the various news sources,Keyword extractor this module will extract the keyword from the given input data,properties generator it will generate the properties depends of the given data, classifier in this module by using the data from properties generator and training data the system will predict whether the news is correct or not.

5.4.1 Class Diagram

This is the Class diagram of the system in which the modules which will be there after the deployment are shown. This class diagram is an illustration of the relationships and source code dependencies among classes in the Unified Modeling Language (UML). In this context, a class defines the methods and variables in an object, which is a specific entity in a program or the unit of code representing that entity.

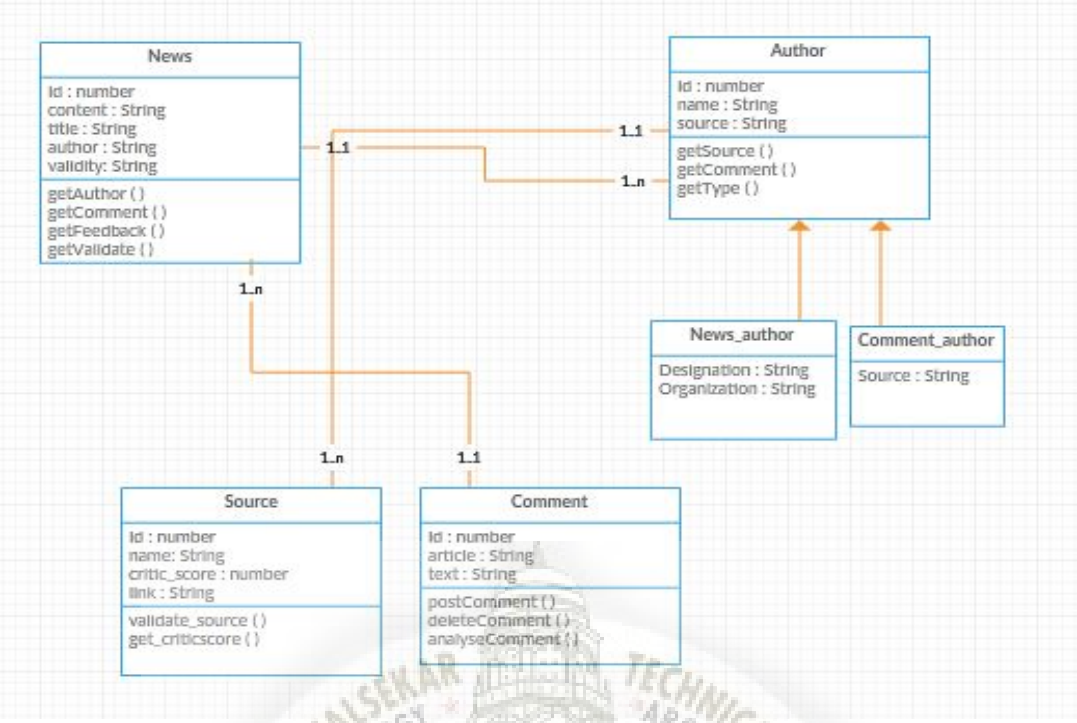

Figure 5.11: Class diagram for News Validation System

5.4.2 Sequence Diagram

To understand what a sequence diagram is, it's important to know the role of UML. UML, or the Unified Modeling Language, is a modeling toolkit that guides the creation and notation of many types of diagrams, including behavior diagrams, interaction diagrams, and structure diagrams. Sequence diagrams are a kind of interaction diagram, because they describe how and in what order a group of objects works together. These diagrams are used by software developers and business people alike to understand requirements for a new system or to document an existing process. Sequence diagrams are sometimes known as event diagrams or event scenarios.

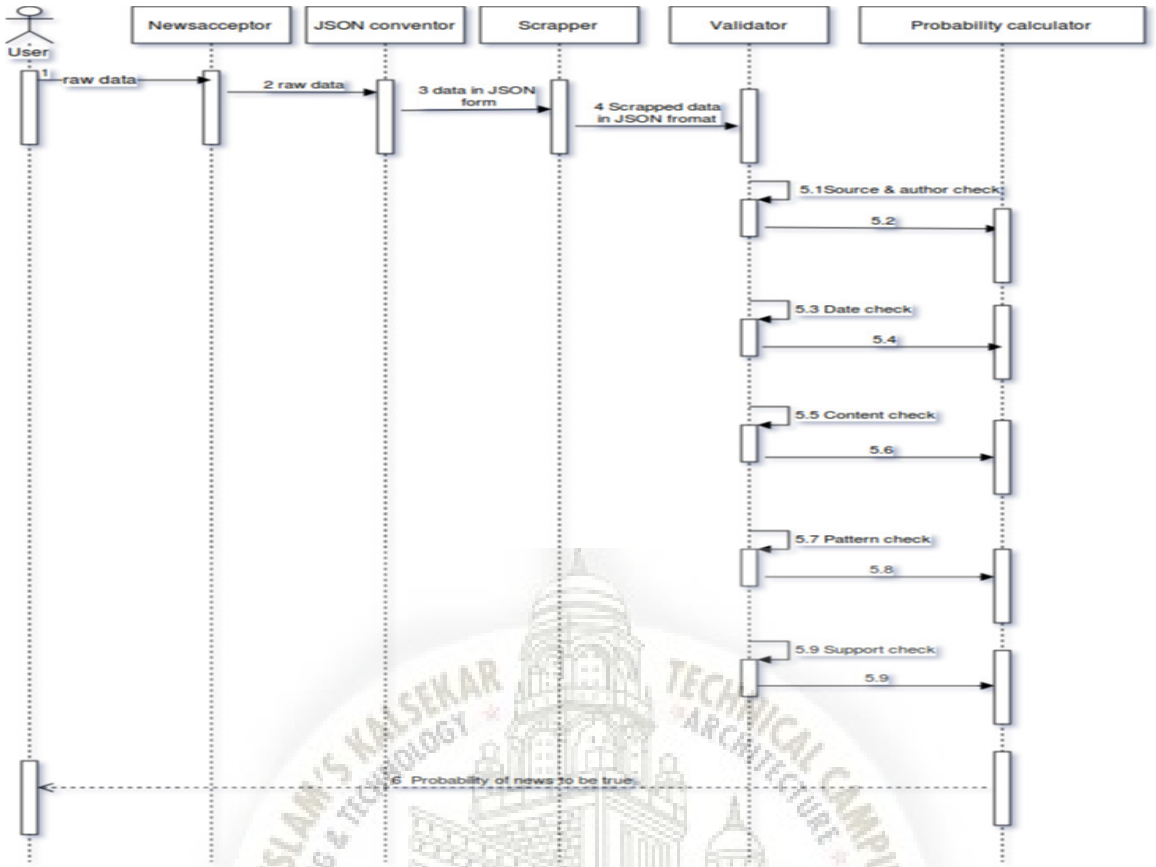

Figure 5.12: Sequence Diagram for News Validation System

5.4.3 Component Diagram

Component diagrams are different in terms of nature and behavior. Component diagrams are used to model the physical aspects of a system. Now the question is, what are these physical aspects? Physical aspects are the elements such as executables, libraries, files, documents, etc. which reside in a node.Component diagrams are used to visualize the organization and relationships among components in a system. These diagrams are also used to make executable systems.

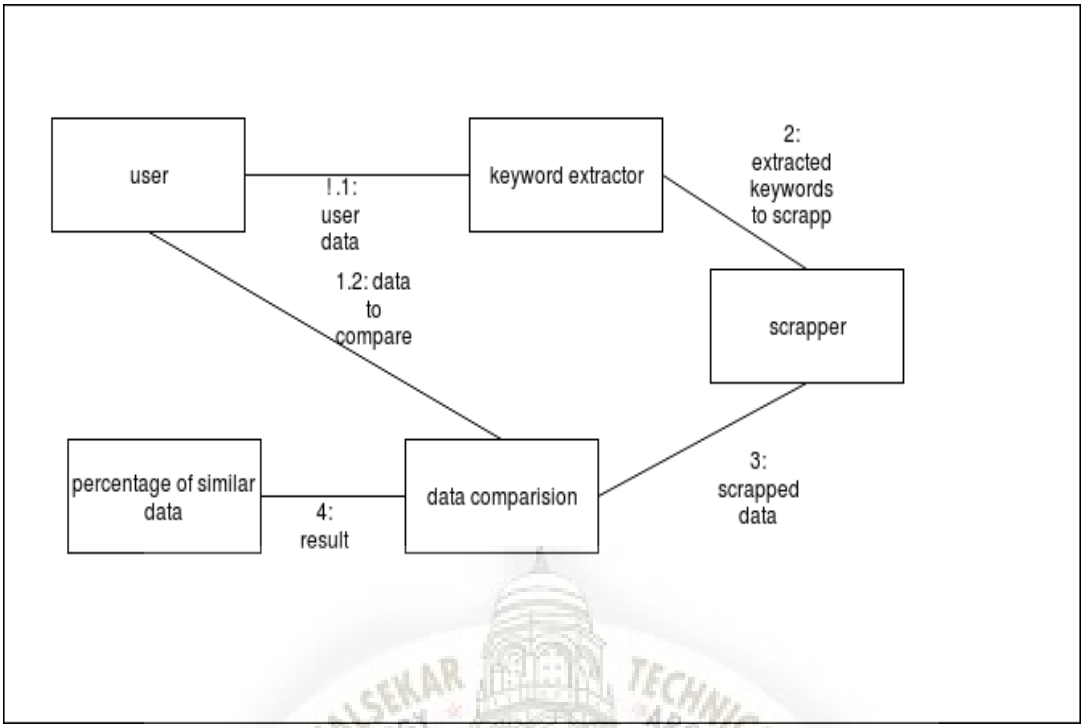

Figure 5.13: Component diagram for News Validation System

5.4.4 Deployment Diagram

Deployment diagram is a structure diagram which shows architecture of the system as deployment (distribution) of software artifacts to deployment targets. The below figure describe the Deployment Diagram of our system.

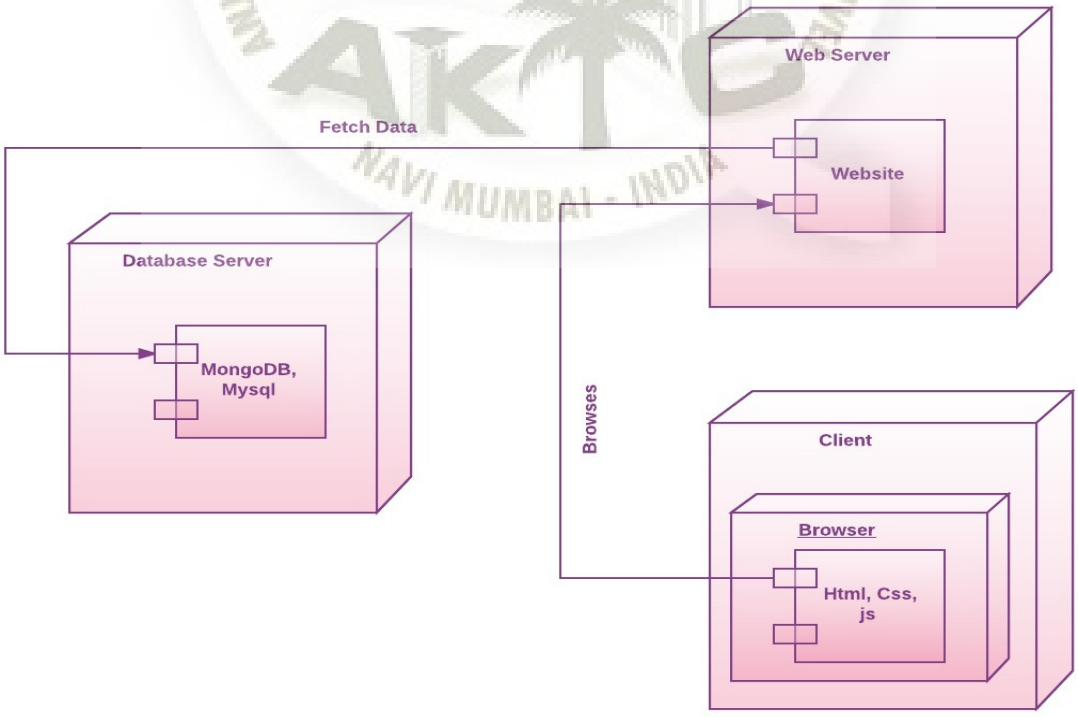

Figure 5.14: Deployment diagram for News Validation System

Chapter 6

Implementation

6.1 Text Analysis Module

This module will analyse the text which is given input to it. It will check the spelling of input data ,grammar of the input data and also punctuation of the input data.

Grammer Check

```
"""
this sub-module checks for grammar errors.
"""
import sys
import urllib.parse
import urllib.request
from urllib.error import HTTPError
from urllib.error import URLError
import json

def get_ginger_url(text):
    """Get URL for checking grammar using Ginger.
    @param text English text
    @return URL
    """
    API_KEY = "6ae0c3a0-afdc-4532-a810-82ded0054236"

    scheme = "http"
    netloc = "services.gingersoftware.com"
    path = "/Ginger/correct/json/GingerTheText"
    params = ""
    query = urllib.parse.urlencode([
        ("lang", "US"),
        ("clientVersion", "2.0"),
        ("apiKey", API_KEY),
        ("text", text)])
    fragment = ""

    return(urllib.parse.urlunparse((scheme, netloc, path, params, query,
        fragment)))

def get_ginger_result(text):
    """Get a result of checking grammar.
    @param text English text
    @return result of grammar check by Ginger
```

```
    """
    url = get_ginger_url(text)

    try:
        response = urllib.request.urlopen(url)
    except HTTPError as e:
        print("HTTP Error:", e.code)
        quit()
    except URLError as e:
        print("URL Error:", e.reason)
        quit()

    try:
        result = json.loads(response.read().decode('utf-8'))
    except ValueError:
        print("Value Error: Invalid server response.")
        quit()

    return(result)

get_ginger_result("this sub-module checks for gramar errrs???")
```

```
"""
this sub module checks for misspelled words in article
"""
from nltk import word_tokenize
import enchant
import re
article = "this sub module checks for misspelled words in article"

d = enchant.Dict("en_US")

non_dict_words = list(set([word.encode('ascii', 'ignore') for word in
    word_tokenize(article) if d.check(word) is False and re.match('^[a-zA-Z ]*$'
    ,word)] ))

non_dict_words
```

```
 """
this sub module will check sentiments of text in between dialogues.
"""
import re
from textblob import TextBlob

content = 'he exclaimed "what the FUCK"'

text = re.findall('"([^"]*)"', content)
blob = TextBlob(text[0])

blob.tags

for sentence in blob.sentences:
    print(sentence)
    print(sentence.sentiment.polarity)
```

6.2 Source, Author , Date Check Module

This module will extract the date of publish, name of author, name of source of the given news from different news website.

```
'''This module will give author,source and date of the data
'''
# -*- coding: utf-8 -*-
print("Initializing...:")

import requests
from goose import Goose
from nltk import word_tokenize, pos_tag
from nltk.corpus import wordnet as wn
from rake_nltk import Rake


authenticate_sources = ["NewsExpress"]
# semantic similarity definitions


def penn_to_wn(tag):
    """ Convert between a Penn Treebank tag to a simplified Wordnet tag """
    if tag.startswith('N'):
        return 'n'

    if tag.startswith('V'):
        return 'v'

    if tag.startswith('J'):
        return 'a'

    if tag.startswith('R'):
        return 'r'

    return None


def tagged_to_synset(word, tag):
    wn_tag = penn_to_wn(tag)
    if wn_tag is None:
        return None

    try:
        return wn.synsets(word, wn_tag)[0]
    except BaseException:
        return None


def sentence_similarity(sentence1, sentence2):
    """ compute the sentence similarity using Wordnet """
    # Tokenize and tag
    sentence1 = pos_tag(word_tokenize(sentence1))
    sentence2 = pos_tag(word_tokenize(sentence2))

    # Get the synsets for the tagged words
    synsets1 = [tagged_to_synset(*tagged_word) for tagged_word in sentence1]
    synsets2 = [tagged_to_synset(*tagged_word) for tagged_word in sentence2]

```

```
    # Filter out the Nones
    synsets1 = [ss for ss in synsets1 if ss]
    synsets2 = [ss for ss in synsets2 if ss]

    score, count = 0.0, 0

    # For each word in the first sentence
    for synset in synsets1:
        # Get the similarity value of the most similar word in the other
        # sentence
        best_score = max([synset.path_similarity(ss) for ss in synsets2])

        # Check that the similarity could have been computed
        if best_score is not None:
            score += best_score
            count += 1

    # Average the values
    score /= count
    return score

# taking input from user


input_news_content = "Late last year, lawyers for President Donald Trump
    expressed optimism that special counsel Robert Mueller was nearing the end
    of his probe of Russia's interference in the 2016 election. But if there was
     hope in the White House that Trump might be moving past an investigation
    that has dogged his presidency from the start, 2018 is beginning without
    signs of abatement. In fact, the new year set off a flurry of developments
    in the probes by Mueller and Congress"
#input_news_content = "Forensic doctors in Dubai concluded that Sridevi died of
    a heart attack and added there is nothing suspicious about the way the
    superstar passed away , official sources in Dubai said."
#input_news_content = "Sehwag apologises for his tweet that gave Kerala lynching
     a communal colour"
#input_news_content = "Scientific research ascertains mercury toxicity but
    Sadhguru continues to endorse it for Indian traditional medicines"
rake = Rake()
rake.extract_keywords_from_text(input_news_content)
keyword = rake.get_ranked_phrases()
print("Extracting Keywords")
print(keyword)

# newsapi starting link declaration
news_api_link = "https://newsapi.org/v2/everything?q="

news_api_query = keyword[0]
print("Getting news data:")

news_data_request = requests.get(
    news_api_link +
    input_news_content +
    "&sortBy=publishedAt&apiKey=de67235049564f0f8d206f8aff2476a1")
news_api_data = news_data_request.json()

if len(news_api_data['articles']) == 0:
    print("No Authenticate news sources found.")
else:

    g = Goose()
```

```
    match = 0
    url = ""
    source = ""
    for i in range(0, len(news_api_data['articles'])):
        url = news_api_data['articles'][i]['url']
        article = g.extract(url=url)
        article_text = article.cleaned_text
        if article_text != '':
            similarity = sentence_similarity(article_text, input_news_content)
            if match < similarity:
                match = similarity
                matched_news = article.title
                date = str(news_api_data['articles'][i]['publishedAt'])
                url = news_api_data['articles'][i]['url']
                source = news_api_data['articles'][i]['source']['name']
                author = news_api_data['articles'][i]['author']
                # print(len(words))
                print(article.title)
                print("Published At:", date[0: 10])
                print("Semantic Similarity: ", match)
                print("URL: ", url)
                print("Source: ", source)
                print("Author: ", author)
        # extracted_articles.append(article.cleaned_text)
```

6.3 Support Check Module.

This Module will extract the twitter tweets of verified user on particular topic and return the numbers of positive tweets,negative tweets and neutral tweets.

```
'''
python3 tweep.py

Twitter sentiment analysis

'''

#!/usr/bin/python3
from bs4 import BeautifulSoup
from time import gmtime, strftime
import argparse
import aiohttp
import asyncio
import async_timeout
import csv
import datetime
import json
import re
import sys
import json
from textblob import TextBlob


async def getUrl(init):
    '''
    URL Descision:
    Tweep utilizes positions of Tweet's from Twitter's search feature to
    iterate through a user's Twitter feed. This section decides whether
    this is the first URL request or not and develops the URL based on the
    args given.

    Returns complete URL.

    Todo: Make everything URL encoded at the end.
    '''
    if init == -1:
        url = "https://twitter.com/search?f=tweets&vertical=default&lang=en&q="
    else:
        url = "https://twitter.com/i/search/timeline?f=tweets&vertical=default"
        url+= "&lang=en&include_available_features=1&include_entities=1&reset_"
        url+= "error_state=false&src=typd&max_position={}&q=".format(init)

    if arg.u != None:
        url+= "from%3A{0.u}".format(arg)
    if arg.g != None:
        arg.g = arg.g.replace(" ", "")
        url+= "geocode%3A{0.g}".format(arg)
    if arg.s != None:
        arg.s = arg.s.replace(" ", "%20").replace("#", "%23")
        url+= "%20{0.s}".format(arg)
    if arg.year != None:
        url+= "%20until%3A{0.year}-1-1".format(arg)
    if arg.since != None:
        url+= "%20since%3A{0.since}".format(arg)
```

```
    if arg.fruit:
        url+= "%20myspace.com%20OR%20last.fm%20OR"
        url+= "%20mail%20OR%20email%20OR%20gmail%20OR%20e-mail"
        url+= "%20OR%20phone%20OR%20call%20me%20OR%20text%20me"
        url+= "%20OR%20keybase"
    if arg.verified:
        url+= "%20filter%3Averified"

    return url

async def fetch(session, url):
    '''
    Basic aiohttp request with a 30 second timeout.
    '''
    with async_timeout.timeout(30):
        async with session.get(url) as response:
            return await response.text()

async def initial(response):
    '''
    Initial response parsing and collecting the position ID
    '''
    soup = BeautifulSoup(response, "html.parser")
    feed = soup.find_all("li", "js-stream-item")
    init = "TWEET-{}-{}".format(feed[-1]["data-item-id"], feed[0]["data-item-id"])

    return feed, init

async def cont(response):
    '''
    Regular json response parsing and collecting Position ID
    '''
    json_response = json.loads(response)
    html = json_response["items_html"]
    soup = BeautifulSoup(html, "html.parser")
    feed = soup.find_all("li", "js-stream-item")
    split = json_response["min_position"].split("-")
    split[1] = feed[-1]["data-item-id"]
    init = "-".join(split)

    return feed, init

async def getFeed(init):
    '''
    Parsing Descision:
    Responses from requests with the position id's are JSON,
    so this section decides whether this is an initial request
    or not to use the approriate response reading for parsing
    with BeautifulSoup4.

    Returns html for Tweets and position id.
    '''
    async with aiohttp.ClientSession() as session:
        response = await fetch(session, await getUrl(init))
    feed = []
    try:
        if init == -1:
            feed, init = await initial(response)
        else:
            feed, init = await cont(response)
```

```
115     except:
116         # Tweep will realize that it's done scraping.
117         pass
118 
119     return feed, init
120 
121 def outTweet(tweet,dat):
122     '''
123     Parsing Section:
124     This function will create the desired output string and
125     write it to a file or csv if specified.
126 
127     Returns output.
128     '''
129     #print("hello ",dat)
130     tweetid = tweet["data-item-id"]
131     global lst
132     # Formatting the date & time stamps just how I like it.
133     datestamp = tweet.find("a", "tweet-timestamp")["title"].rpartition(" - ")
        [-1]
134     d = datetime.datetime.strptime(datestamp, "%d %b %Y")
135     date = d.strftime("%Y-%m-%d")
136     timestamp = str(datetime.timedelta(seconds=int(tweet.find("span", "
        _timestamp")["data-time"]))).rpartition(", ")[-1]
137     t = datetime.datetime.strptime(timestamp, "%H:%M:%S")
138     time = t.strftime("%H:%M:%S")
139     # The @ in the username annoys me.
140     username = tweet.find("span", "username").text.replace("@", "")
141     timezone = strftime("%Z", gmtime())
142     # The context of the Tweet compressed into a single line.
143     text = tweet.find("p", "tweet-text").text.replace("\n", "").replace("http",
        " http").replace("pic.twitter", " pic.twitter")
144     # Regex for gathering hashtags
145     hashtags = ",".join(re.findall(r'(?i)\#\w+', text, flags=re.UNICODE))
146     replies = tweet.find("span", "ProfileTweet-action--reply u-hiddenVisually").
        find("span")["data-tweet-stat-count"]
147     retweets = tweet.find("span", "ProfileTweet-action--retweet u-hiddenVisually
        ").find("span")["data-tweet-stat-count"]
148     likes = tweet.find("span", "ProfileTweet-action--favorite u-hiddenVisually")
        .find("span")["data-tweet-stat-count"]
149     '''
150     This part tries to get a list of mentions.
151     It sometimes gets slow with Tweets that contain
152     40+ mentioned people.. rather than just appending
153     the whole list to the Tweet, it goes through each
154     one to make sure there arn't any duplicates.
155     '''
156     try:
157         mentions = tweet.find("div", "js-original-tweet")["data-mentions"].split
            (" ")
158         for i in range(len(mentions)):
159             mention = "@{}".format(mentions[i])
160             if mention not in text:
161                 text = "{}".format(text)
162     except:
163         pass
164 
165     # Preparing to output
166 
167     '''
168     There were certain cases where I used Tweep
```

```
    to gather a list of users and then fed that
    generated list into Tweep. That's why these
    modes exist.
    '''
    if arg.users:
        output = username
    elif arg.tweets:
        output = tweets
    else:
        '''
        The standard output is how I like it, although
        this can be modified to your desire. Uncomment
        the bottom line and add in the variables in the
        order you want them or how you want it to look.
        '''
        output = ""
        output = "{}".format(text)


    if arg.o == None:
      dat=[text]

    return dat

async def getTweets(init,var):
    '''
    This function uses the html responses from getFeed()
    and sends that info to the Tweet parser outTweet() and
    outputs it.

    Returns response feed, if it's first-run, and Tweet count.
    '''
    tweets, init = await getFeed(init)
    count = 0
    dat=[]
    for tweet in tweets:
        '''
        Certain Tweets get taken down for copyright but are still
        visible in the search. We want to avoid those.
        '''
        copyright = tweet.find("div","StreamItemContent--withheld")
        if copyright is None:
            count +=1
            var.append(outTweet(tweet,dat))
            if(count==20):
              break
           # print(var)
    return count,var

async def getUsername():
    '''
    This function uses a Twitter ID search to resolve a Twitter User
    ID and return it's corresponding username.
    '''
    async with aiohttp.ClientSession() as session:
        r = await fetch(session, "https://twitter.com/intent/user?user_id={0.
            userid}".format(arg))
    soup = BeautifulSoup(r, "html.parser")
    return soup.find("a", "fn url alternate-context")["href"].replace("/", "")

async def main():
```

```
    '''
    Putting it all together.
    '''
    if arg.userid is not None:
        arg.u = await getUsername()

    feed = [-1]
    init = -1
    num = 0
    var=[]
    neu=pos=neg=0
    tweets_senti=[]
    if len(feed) > 0:
              num,var = await getTweets(init,var)
              for i in range (num):
                tweets=""
                tweets="".join(var[i])
                tweets1=tweets.split('http')
                #print(tweets1[0])
                tweets_senti.append([TextBlob(tweets1[0]).polarity])
                if(TextBlob(tweets1[0]).polarity>0):
                  pos+=1
                elif(TextBlob(tweets1[0]).polarity<0):
                  neg+=1
                else:
                  neu+=1
              #print(tweets_senti)
              print("positive tweets ",pos," negative tweets ",neg," neutral
                  tweets ",neu)


def Error(error, message):
    # Error formatting
    print("[-] {}: {}".format(error, message))
    sys.exit(0)

def check():
    # Performs main argument checks so nothing unintended happens.
    if arg.u is not None:
        if arg.users:
            Error("Contradicting Args", "Please use --users in combination with
                -s.")
        if arg.verified:
            Error("Contradicting Args", "Please use --verified in combination
                with -s.")
        if arg.userid:
            Error("Contradicting Args", "--userid and -u cannot be used together
                .")
    if arg.tweets and arg.users:
        Error("Contradicting Args", "--users and --tweets cannot be used
            together.")
    if arg.csv and arg.o is None:
        Error("Error", "Please specify an output file (Example: -o file.csv")
    if arg.u is None and arg.s is None and arg.userid is None and arg.g is None:
        Error("Error", "Please specify a username, user id, search or geotag.")

def check_support(user_data,limit,verified):

#if __name__ == "__main__":
    ap = argparse.ArgumentParser(prog="tweep.py", usage="python3 %(prog)s [
        options]", description="tweep.py - An Advanced Twitter Scraping Tool")
```

```
    ap.add_argument("-u", help="User's Tweets you want to scrape.")
    ap.add_argument("-s", help="Search for Tweets containing this word or phrase.", default=user_data)
    ap.add_argument("-o", help="Save output to a file.")
    ap.add_argument("-g", help="Search for geocoded tweets.")
    ap.add_argument("--year", help="Filter Tweets before specified year.")
    ap.add_argument("--since", help="Filter Tweets sent since date (Example: 2017-12-27).")
    ap.add_argument("--fruit", help="Display 'low-hanging-fruit' Tweets.", action="store_true")
    ap.add_argument("--tweets", help="Display Tweets only.", action="store_true")
    ap.add_argument("--verified", help="Display Tweets only from verified users (Use with -s).", action="store_true", default=verified)
    ap.add_argument("--users", help="Display users only (Use with -s).", action="store_true")
    ap.add_argument("--csv", help="Write as .csv file.", action="store_true")
    ap.add_argument("--hashtags", help="Output hashtags in seperate column.", action="store_true")
    ap.add_argument("--userid", help="Twitter user id")
    ap.add_argument("--limit", help="Number of Tweets to pull (Increments of 20).", default=limit)
    ap.add_argument("--count", help="Display number Tweets scraped at the end of session.", action="store_true")
    ap.add_argument("--stats", help="Show number of replies, retweets, and likes", action="store_true")
    global arg
    #global var
    arg = ap.parse_args()
    #print(arg.verified)
    check()

    loop = asyncio.get_event_loop()
    loop.run_until_complete(main())
    polar=[]

check_support('Salman khan after black buck case',20,True)
```

6.4 Content Check Module

This module will basically check the content of the given data input.

```
'''
Content Check Module
'''
from textblob import TextBlob
from gingerit.gingerit import GingerIt
import re

class Linguistic:
    def __init__(self):
        self.grammar_mistakes_count = 0
        self.content_sentiment = {}
        self.quote_sentiment = {}
        self.caps_sentiment = {}

    def grammar_mistake(self, content):
        parser = GingerIt()
        result = parser.parse(content)
        return len(result['corrections'])

    def get_quotes_sentiment(self, content):
        quotes = re.findall(r'"([^"]*)"', content)
        quotes_string = '. '.join(quotes)

        self.quote_sentiment = self.sentiment_analysis(quotes_string)

    def get_caps_sentiment(self, content):
        caps = re.findall('\w+[A-Z]', content)
        caps_string = '. '.join(caps)
        #
        self.caps_string_length=len(''.join(caps))
        #
        self.caps_sentiment = self.sentiment_analysis(caps_string)

    def sentiment_analysis(self, content):
        blob = TextBlob(content)
        sentiments = {'negative': 0, 'positive': 0, 'neutral': 0}
        for sentence in blob.sentences:
            if sentence.sentiment.polarity < 0:
                sentiments['negative'] += 1
            elif sentence.sentiment.polarity > 0:
                sentiments['positive'] += 1
            elif sentence.sentiment.polarity == 0:
                sentiments['neutral'] += 1
        return sentiments

    def get_linguistic_features(self, input_news_content):

        self.grammar_mistakes_count = self.grammar_mistake(input_news_content)
        self.content_sentiment = self.sentiment_analysis(input_news_content)
        self.get_quotes_sentiment(input_news_content)
        self.get_caps_sentiment(input_news_content)

        return [self.grammar_mistakes_count,self.caps_string_length, self.
            content_sentiment, self.quote_sentiment, self.caps_sentiment]
```

6.5 Classifier

In our system we have used two classifier ID3 classifier and KNN classifier.

```
'''
ID3
'''
import pandas as pd
import numpy as np

df = pd.read_csv('kyphosis.csv')

from sklearn.model_selection import train_test_split

X = df.drop('Kyphosis', axis=1)
y = df['Kyphosis']

X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=0.30)

from sklearn.tree import DecisionTreeClassifier

dtree = DecisionTreeClassifier()

dtree.fit(X_train,y_train)

predictions = dtree.predict(X_test)

from sklearn.metrics import classification_report,confusion_matrix

print(classification_report(y_test,predictions))

print(confusion_matrix(y_test,predictions))


#random forest

from sklearn.ensemble import RandomForestClassifier

rfc = RandomForestClassifier(n_estimators=100)

rfc.fit(X_train, y_train)

rfc_pred = rfc.predict(X_test)

print(confusion_matrix(y_test,rfc_pred))

print(classification_report(y_test,rfc_pred))
```

```
'''
Knn
'''
import pandas as pd
import numpy as np
import matplotlib.pyplot as plt
import numpy as np

df = pd.read_csv('kyphosis.csv')

```

```
from sklearn.preprocessing import StandardScaler

scaler = StandardScaler()

scaler.fit(df.drop('Kyphosis',axis=1))

scaled_features = scaler.transform(df.drop('Kyphosis',axis=1))

df_feat = pd.DataFrame(scaled_features, columns=df.columns[:-1])

from sklearn.model_selection import train_test_split

X_train, X_test, y_train, y_test = train_test_split(scaled_features,df['Kyphosis'],
                                                    test_size=0.30)
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=1)
knn.fit(X_train,y_train)
pred = knn.predict(X_test)

from sklearn.metrics import classification_report,confusion_matrix

print(confusion_matrix(y_test,pred))

print(classification_report(y_test,pred))

error_rate = []

# Will take some time
for i in range(1, 40):
    knn = KNeighborsClassifier(n_neighbors=i)
    knn.fit(X_train, y_train)
    pred_i = knn.predict(X_test)
    error_rate.append(np.mean(pred_i != y_test))


plt.figure(figsize=(10,6))
plt.plot(range(1,40),error_rate,color='blue', linestyle='dashed', marker='o',
         markerfacecolor='red', markersize=10)
plt.title('Error Rate vs. K Value')
plt.xlabel('K')
plt.ylabel('Error Rate')
plt.show()
knn = KNeighborsClassifier(n_neighbors=10)

knn.fit(X_train,y_train)

print(type(X_test))
from numpy import array
#a = array([[1.46067113, -0.65203532,  1.34045062]])
#a.reshape(-1,1)
pred = knn.predict(X_test)
print(pred)
print('WITH K=23')
print('\n')
print(confusion_matrix(y_test,pred))
print('\n')
print(classification_report(y_test,pred))
```

Chapter 7

System Testing

The system testing is done as per as following criterias

7.1 Test Cases and Test Results

Test ID	Test Case Title	Test Condition	System Behavior	Expected Result
T01	Login	Should be registered user	User should login into our system	User should login into our system
T02	Search data	Data must be in correct form	Display the probability	Display the probability
T03	Search non existing data	Data must be in correct form	Output should be null	Output should be null

7.2 Sample of a Test Case

Title: Login Page – Authenticate Successfully on our system.
Description: A registered user should be able to successfully login at our system.

Precondition: The user must already be registered with an name, email address and password.

Assumption: A supported browser is being used.

Test Steps:

1. Navigate to register.
2. Register on our system
3. In the 'email' field, enter the email of the registered user.

4. Enter the password of the registered user
5. Click ‘Login’

Expected Result: After the user has been login successfully the user will get its API key.

Actual Result: After user has login a random and unique key will be generated for further authentication of user. As expected the user will get its API key.

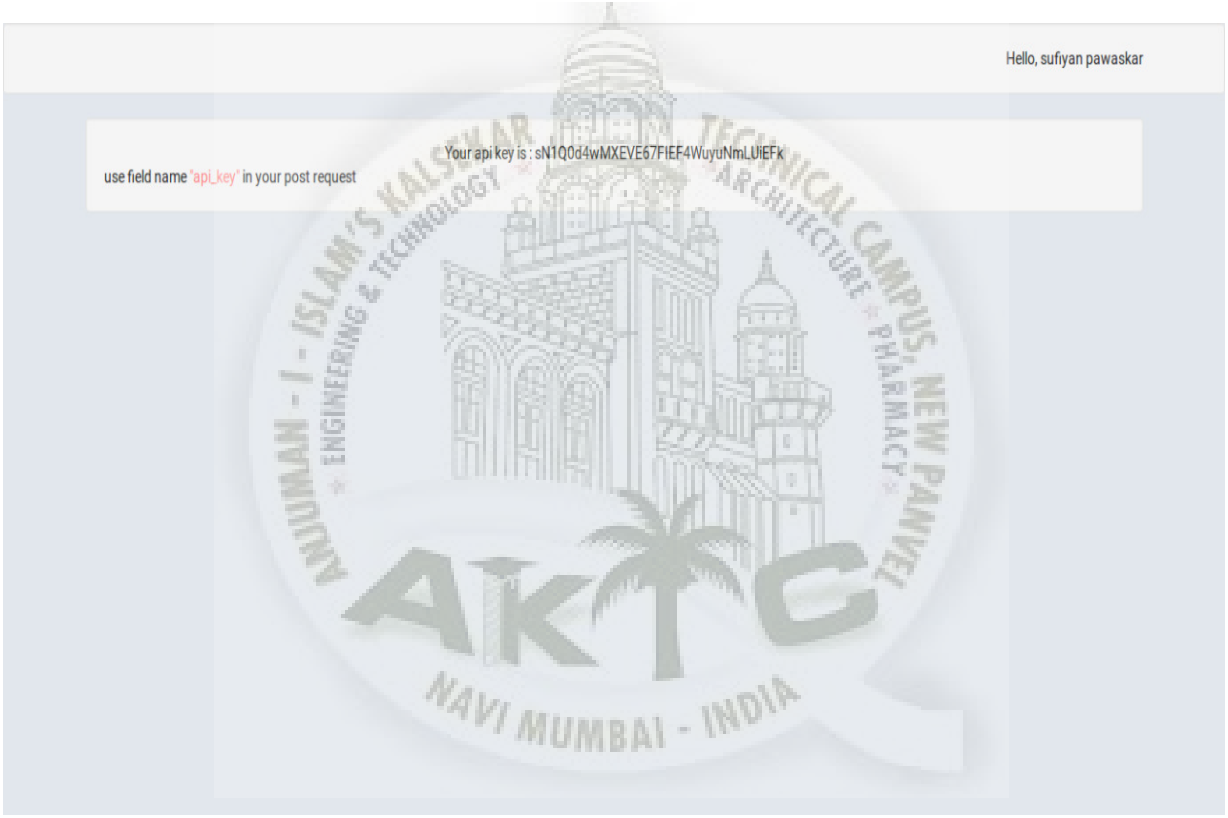

Figure 7.1: Key generation after login

7.2.1 Software Quality Attributes

1. Availability:The system should not be down,whenever the user use the system the specific data should be available to the user.
2. Correctness:As per as the user search correct data should be shown to user.

3. Maintainability:The administrator of the system should maintain the system.
4. Reliability: The system should be reliable for producing correct output so that user can reliable on system.
5. Extensibility: The system is capable to be modified by changing some modules or by adding some features to the existing system.

Chapter 8

Screenshots of Project

8.1 Registration of client

Figure 8.1: Register

8.2 Login of client

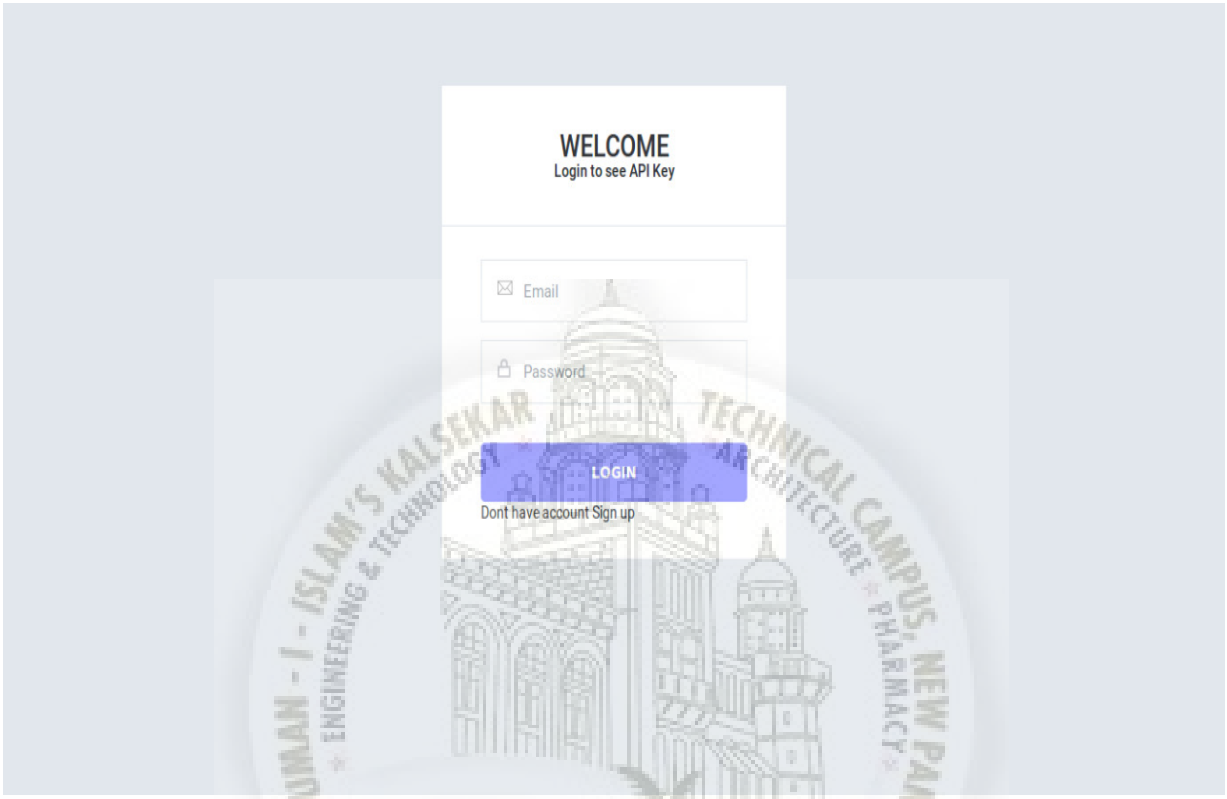

Figure 8.2: Login

8.3 Api Key Generation for client

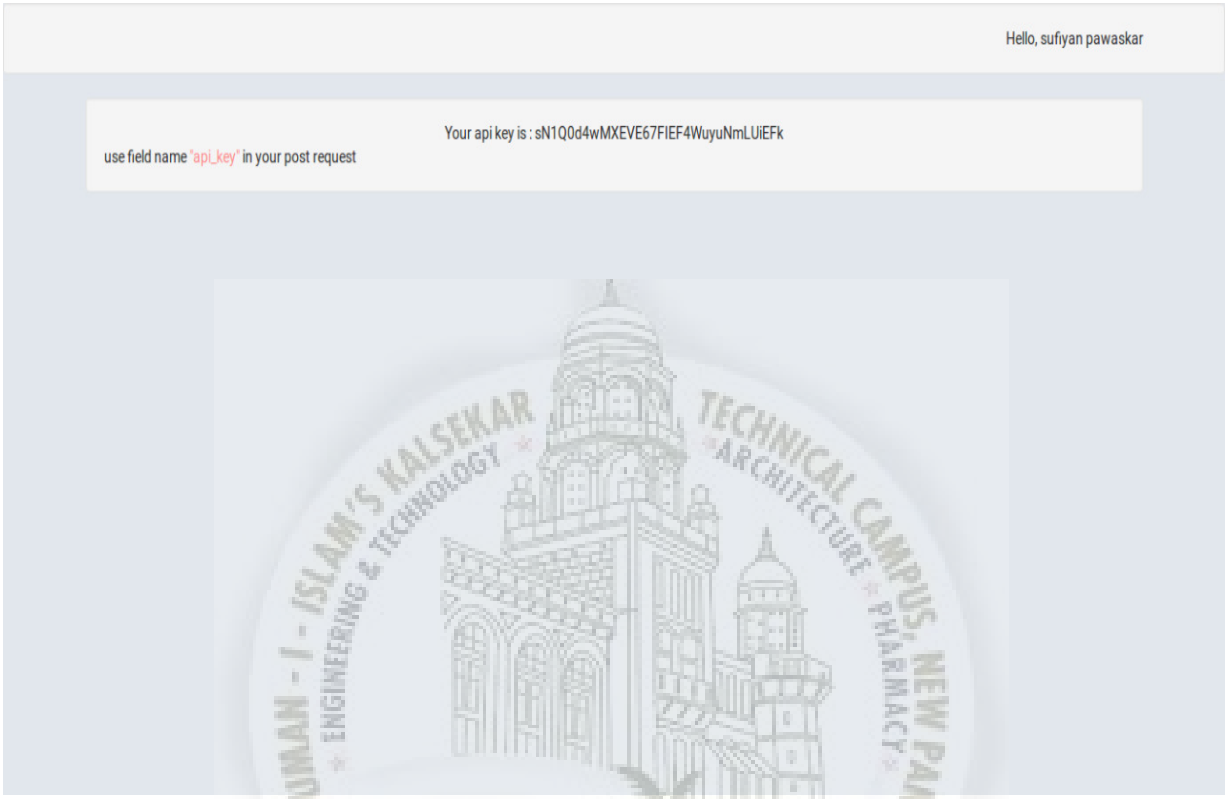

Figure 8.3: Key generation after login

8.4 Output for the search data

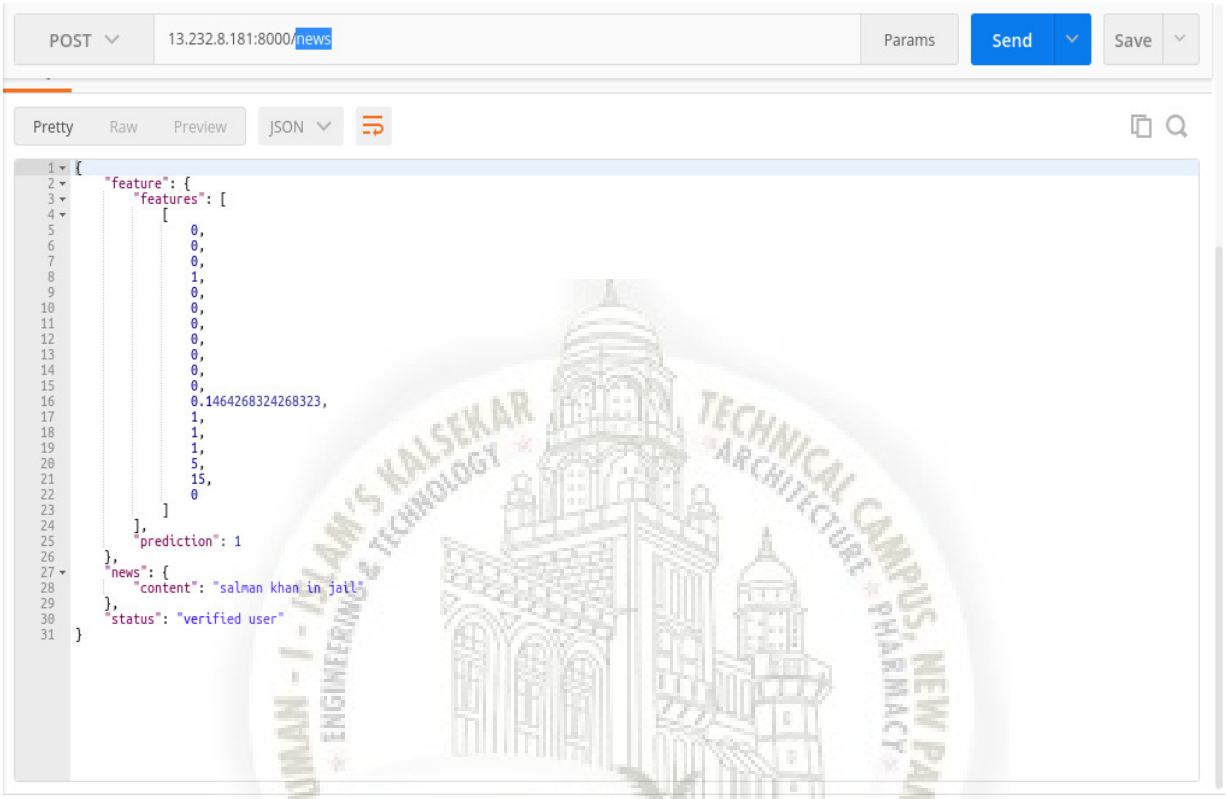

Figure 8.4: Output

Chapter 9

Conclusion and Future Scope

9.1 Conclusion

Information in internet should be correct and reliable. But currently our society lacks a system that can check whether if some information implied in news is true or not. Developing such system will make a difference in society and will help to maintain peace.

9.2 Future Scope

Currently, We are targeting only global news, In future targeting the local news can be implemented.

Also geographical area wise analysis can be implemented that would make the results more efficient.

Also, our system can be implemented on social media platform, so that sharing of any uncertain news can be evaluated.

Only text can be put as input in our system, in future taking input in the form of image and videos can make our system more wide in scope of news.

References

[1] *Automatic deception detection: Methods for finding fake news*; N. J. Conroy, V. L. Rubin, and Y. Chen,Proceedings of the Association for Information Science and Technology, 2015

[2] *News Aggregation and Validation System*; Josef Moucachein, 2017

[3] *Social Media and Fake News in the 2016 Election*; Hunt Alcott, 2017

[4] *Deception Detection for News: Three Types of Fakes*; Victoria L. Rubin, Yimin Chen and Niall J. Conroy,University of Western Ontario,2015

Achievements

1. Conferences

   (a) *News Validation System*; Sufiyan Pawaskar, Nooralam Shaikh, Pradnyesh Rane,Prof.Tabrez Khan ,Conference on Recent Trends in Computer Engineering , February and 2018 of attend(Venue : Thakur College of Engineering.)

2. Project Competitions

   (a) *News Validation System*;Sufiyan Pawaskar, Nooralam Shaikh, Pradnyesh Rane,Prof.Tabrez Khan,4th National Level Project Exhiition Cum Poster Presentation , 9th march and 2018 of attend(Venue :Universal College of Engineering) **Secured 2nd position.**

Laxmi Singh Charitable Trust's (Regd.)

THAKUR TRUSTS

THAKUR COLLEGE OF ENGINEERING & TECHNOLOGY

(Approved by AICTE, Govt. of Maharashtra & Affiliated to University of Mumbai)

Thakur Village, Kandivali (E), Mumbai - 400101.

ISO 9001 : 2015 Certified
NBA and NAAC Accredited

Certificate of Appreciation

This is to certify that Dr./Mr./Ms. Sufiyan Pawaskar has presented/published a Short length paper with the title News Validation System in the Conference on Recent Trends in Computer Engineering (CRTCE 2018) organized during February, 23rd & 24th, 2018 at Thakur College of Engineering and Technology.

(Dr. Sheetal Rathi)
Convenor

(Dr. R. R. Sedamkar)
Technical Chair

(Dr. B. K. Mishra)
Principal & Programme Chair

MULTICON-W 2018

9th National & International Conferences & Workshops

Platinum Partners

Diamond Partners

Technical Partners

Collaborations

Other Partners

Universal College of Engineering

DTE Code: 3460

(Permanently Unaided | Approved by AICTE, DTE & Affiliated to University of Mumbai)

Near Bhajansons and Pinyadham, Kaman Bhiwandi Road, Vasai

in association with **I.E.T.E. - I.S.F., C.S.I. & I.S.A.**

4th National Level Project Exhibition Cum Poster Presentation

Certificate of Merit

This is to certify that Ms./Mr. SUFIYAN PAWASKAR

of A·I·K·T·C

College has won ~~*First Prize*~~ */ 1st Runner up /* ~~*2nd Runner up*~~ */* ~~*Consolation*~~ *Prize in the*

"4th National Level Project Exhibition Cum Poster Presentation" *2018.*

Date: 9th March 2018

Dr. Ajoy Kumar
(Principal)

Dr. J. B. Patil
(Campus Director)

Universal College of Engineering

DTE Code: 3460

(Permanently Unaided | Approved by AICTE, DTE & Affiliated to University of Mumbai)

Near Bhajansons and Pinyadham, Kaman Bhiwandi Road, Vasai

in association with **I.E.T.E. - I.S.F., C.S.I. & I.S.A.**

4th National Level Project Exhibition Cum Poster Presentation

Certificate of Merit

This is to certify that Ms./Mr. SHAIKH MOHD NOORALAM

of A.I. KALSEKAR TECHNICAL CAMPUS

College has won ~~First Prize~~ / 1st Runner up / ~~2nd Runner up~~ / ~~Consolation Prize~~ in the

"4th National Level Project Exhibition Cum Poster Presentation" *2018.*

Date: 9th March 2018

Dr. Ajoy Kumar
(Principal)

Dr. J. B. Patil
(Campus Director)

Universal College of Engineering

DTE Code: 3460

(Permanently Unaided | Approved by AICTE, DTE & Affiliated to University of Mumbai)

Near Bhajansons and Pinyadham, Kaman Bhiwandi Road, Vasai

in association with I.E.T.E. - I.S.F. C.S.I. & I.S.A.

4th National Level Project Exhibition Cum Poster Presentation

Certificate of Merit

This is to certify that Ms./Mr. PRADNYESH RANE

of A·I·K·T·C

College has won ~~First Prize~~ / 1st Runner up / ~~2nd Runner up~~ / ~~Consolation~~ Prize in the

"4th National Level Project Exhibition Cum Poster Presentation" *2018.*

Date: 9th March 2018

Dr. Ajoy Kumar
(Principal)

Dr. L.B. Patil
(Campus Director)