A DISSERTATION REPORT ON

SENTIMENT ANALYSIS USING ORIGINAL AND REVERSED REVIEWS

SUBMITTED TO THE SAVITRIBAI PHULE PUNE UNIVERSITY, PUNE
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS
FOR THE AWARD OF THE DEGREE OF

MASTER OF ENGINEERING (Computer Engineering)

BY

Kaushik S. Hande Exam No:

Under The Guidance of

Prof. A. G. Phakatkar

DEPARTMENT OF COMPUTER ENGINEERING
PUNE INSTITUTE OF COMPUTER TECHNOLOGY
Sr. No. 27, Pune Satara Road, Dhankawadi
Pune 411043

PUNE INSTITUTE OF COMPUTER TECHNOLOGY

DEPARTMENT OF COMPUTER ENGINEERING

CERTIFICATE

This is to certify that the dissertation entitled

"SENTIMENT ANALYSIS USING ORIGINAL AND REVERSED REVIEWS"

Submitted by

Kaushik S. Hande Exam No:

is a bonafide work carried out by him under the supervision of Prof. A. G. Phakatkar and it is submitted towards the partial fulfillment of the requirement of Savitribai Phule Pune University, for the award of the degree of Master of Engineering (Computer Engineering)

Prof. A. G. Phakatkar	Dr. Rajesh Ingle	Dr. P. T. Kulkarni
Internal Guide	H.O.D	Principal
Dept. of Computer Engg.	Dept. of Computer Engg.	
PICT, Pune-43	PICT, Pune-43	PICT, Pune-43

ACKNOWLEDGEMENT

It gives me great pleasure in presenting the dissertation report for my dissertation on ***"SENTIMENT ANALYSIS USING ORIGINAL AND REVERSED REVIEWS"****.*

I would like to take this opportunity to thank my internal guide ***Dr. A. S. Ghotkar*** *for giving me all the help and guidance I needed. I am really grateful to them for their kind support throughout the analysis and design phase. Their valuable criticism and suggestions were very helpful.*

I am grateful to ***Dr. Rajesh B. Ingle****, Head of Computer Engineering Department, Pune Institute of Computer Technology for his indispensable support, priceless suggestions and for most valuable time lent as and when required.*

In the end my special thanks to ***Mr. Hrushikesh Kasar*** *for providing various resources such as well-equipped lab with all needed software platforms, continuous Internet connection, for my dissertation work.*

Kaushik S. Hande

(M.E. Computer Engg.)

List of Figures

List of Tables

ABSTRACT

Bag of words is used for modeling in machine learning algorithms. However, BOW is not able to handle negation well because of its fundamental deficiencies . Many ways are used to handle the problem of negation which results into polarity shift . They require either knowledge about language constructs or extra human interventions which eventually increases the complexity. In this paper, a data expansion technique, called dual sentiment analysis (DSA), is used to address the polarity shift problem due to negation in sentiment classification. Original and reversed training reviews are used for learning in a sentiment classifier and prediction is done on test reviews.

INDEX

CHAPTER 1

SYNOPSIS

1.1 DISSERTATION TITLE

SENTIMENT ANALYSIS USING ORIGINAL AND REVERSED REVIEWS

1.2 INTERNAL GUIDE

Prof. A. G. Phakatkar

1.3 PROBLEM STATEMENT

”To make use of the original and reversed review samples in pairs for training a statistical classifier and make predictions.”

1.4 OBJECTIVES

- To obtain reversed reviews from each corresponding original reviews.
- To train the classifiers using these reviews.
- To obtain the predictions of labels(positive review or negative review) for test data.

1.5 HYPOTHESIS

Polarity shift causes accuracy of classifier to decrease. We assume that original review and corresponding opposite review can be used together to increase the accuracy of review class label prediction and to avoid the problem caused due to polarity shift.

1.6 RELEVANT MATHEMATICS ASSOCIATED WITH DISSERTATION

1.6.1 Mathematical Model

$S = \{s, e, I, O, fmain|\phi\}$

where,

s = start state

e = end state

I = Inputs to the system

I = $\{x, x', y, y', D, D'\}$

where,

x = original sample

x' = reversed sample

$y \in \{0,1\}$ = The class label of the original sample

$y' = 1 - y$ = The class label of the reversed sample

$D = (x_i, y_i)_{i=1}^{n}$ = original training set

$D' = (x_i', y_i')_{i=1}^{n}$ = The reversed training set

O = Output

O = $\{ p(x), p(x'), p(x, x')\}$

where

p(x) = Prediction for the original sample

$p(x')$ = Prediction for the reversed sample

$p(x, x')$ = Dual prediction based on a pair of sample

$f_{main} = \{f_{reverse}, f_{classifier}\}$

$f_{reverse} =$ function for reversing the corresponding each review

$f_{classifier} =$ classifier for the prediction of class of review

1.6.2 Metrics for Performance Evaluation

Several statistical measures are used for performance evaluation -

- Accuracy-is the proximity of measurement results to the true value.

$$\frac{TP+TN}{TP+TN+FP+FN} \tag{1.1}$$

- Sensitivity- measures the proportion of positives that are correctly identified

$$\frac{TP}{TP+FN} \tag{1.2}$$

- Specificity- measures the proportion of negatives that are correctly identified

$$\frac{TN}{TN+FP} \tag{1.3}$$

- Positive predictive value- are the proportions of positive results in statistics and diagnostic tests

$$\frac{TP}{TP+FP} \tag{1.4}$$

- Negative predictive value- are the proportions of negative results in statistics and diagnostic tests

$$\frac{TN}{TN+FN} \tag{1.5}$$

CHAPTER 2

TECHNICAL KEYWORDS

2.1 AREA OF DISSERTATION

Natural language processing, machine learning, sentiment analysis, opinion mining.

2.2 ACM KEYWORDS

- A Information Systems
  - A.1 Information Retrievals
    - A.1.1 Retrieval tasks and goals
      - A.1.1.1 Sentiment analysis
      - A.1.1.2 Clustering and classification
- B Computing methodologies
  - B.1 Machine learning
    - B.1.1 Supervised learning by classification
      - B.1.1.1 Multinomial Naive Bayes
      - B.1.1.2 Random Forest
      - B.1.1.3 Support Vector Machines

CHAPTER 3

INTRODUCTION

3.1 DISSERTATION IDEA

Sentiment is an attitude, thought, or judgement prompted by feeling. Sentiment analysis is also known as opinion mining, it involves studing of peoples sentiments towards certain entities. Internet is a resourceful place with respect to sentiment information. From a perspective of a user, people are able to express their views through various social media, such as forums, micro-blogs, or online social networking sites.

With the advent of Web 2.0 techniques, users started prefering to share their opinions on the Web. These user-generated and sentiment-rich data are valuable to many applications like credibility analysis of news sites on the Web, recommendation system, business and government intelligence etc. At the same time, it brings urgent need for detecting overall sentiment inclinations of documents generated by users, which can be treated as a classification problem. Sentiment analysis includes several subtasks which have seen a great deal of attention in recent years:

1. To detect whether a given document is subjective or objective.
2. To Identify whether given subjective document express a positive opinion or a negative opinion.
3. To determine the sentiment strength of a document, such as strongly negative, weakly negative, neutral, weakly positive and strongly positive.

In this work we are focusing on second subtask.

Besides individuals on social media marketers also need to monitor all media for information related to their brands whether its for public relations activities, fraud violations, or competitive intelligence. Thus, aside from individuals, sentiment analysis is also the need of companies which are anxious to understand how their products and services are perceived by the public.

The dominating text representation method in both supervised and semi supervised sentiment classification is known as the bag-of-words (BOW) model, which is difficult to meet the requirements for understanding the review text and dealing with complex linguistic structures such as negation. For example, the BOW representations of two opposite reviews “It works well” and “It doesn’t work well” are considered to be very similar by most statistical learning algorithms.The two sentiment opposite texts are considered to be very similar by the BOW representation. This is exactly why standard machine learning algorithms often fail under the circumstance of polarity shift due to negation in the sentences of the review text.

Several approaches have been proposed in the literature to address the polarity shift problem .They require either knowledge about language constructs or extra human interventions which eventually increases the complexity in classification of sentiment. Such high-level dependency on external resources makes the systems difficult to be widely used in practice. There were also some efforts to address the polarity shift problem with the absence of extra annotations and linguistic knowledge. However, results are still far from satisfactory.

3.2 MOTIVATION OF DISSERTATION

Polarity shift is a kind of linguistic phenomenon which can reverse the sentiment polarity of the text. Negation is the most important type of polarity shift. For example, by adding a negation word dont to a positive text I like this book in front of the word like, the sentiment of the this book in front of the word like, the sentiment of the text will be reversed from positive to negative. However,the two sentiment-opposite texts are considered to be very similar by the BOW representation. This is the main reason why standard machine learning algorithms often fail under the circumstance of polarity shift.

CHAPTER 4

LITERATURE SURVEY

We studied the related work on sentiment analysis and polarity shift.

4.1 SENTIMENT ANALYSIS AND POLARITY SHIFT

According to the levels of granularity, tasks in sentiment analysis can be divided into four categorizations: document- level, sentence-level, phrase-level, and aspect-level sentiment analysis.

For document and sentence-level sentiment classification, there are two main types of methods in the literature: term-counting and machine learning methods. In term- counting methods, the overall orientation of a text is obtained by summing up the orientation scores of content words in the text, based on manually-collected or external lexical resources [38], [39]. In machine learning methods, sentiment classification is regarded as a statistical classification problem, where a text is represented by a bag-of-words; then, the supervised machine learning algorithms are applied as classifier [35]. Accordingly, the way to handle polarity shift also differs in the two types of methods.

The term-counting methods can be easily modified to include polarity shift. One common way is to directly reverse the sentiment of polarity-shifted words, and then sum up the sentiment score word by word [4], [16], [17], [37]. Compared with term counting methods, the machine learning methods are more widely discussed in the sentiment classification literatures. However, it is relatively hard to integrate the polarity shift information into the BOW model in such methods. For example, Das and Chen [6] proposed a method by simply attaching NOT to words in the scope of negation, so that in the text I dont like book, the word like becomes a new word like-NOT. Yet Pang et al. [35] reported that this method only has slightly negligible effects on improving the sentiment classification accuracy.

4.2 GAP IDENTIFICATION THROUGH LITERATURE SURVEY

The following table shows the literature survey about different techniques of sentiment analysis used for classification.

Table 4.1: Literature Survey

No.	Reference	Techniques	Description
1	Dual Sentiment Analysis: Considering Two sides of one review	Support vector machine (SVM), Naive bayes, Logistic Regression	Dual training and Dual Prediction technique is used.
2	Thumbs up?Sentiment Classification using Machine learning algorithms	Learning algorithms and n-gram model	Classify the dataset using different machine.
3	Classification of sentiment reviews using N-gram machine learning approach	Support Vector Machine Naive Bayes	Converting text reviews into numeric matrices using countvectorizer and TF-IDF
4	Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews	Unsupervised learning algorithm for classifying a review	A specific unsupervised learning technique based on the mutual information
5	Automatic Opinion polarity Classification of movie	Naive Bayes And Markov Model (MM)	Accessed overall opinion polarity(OvOp) concept using machine learning algorithm
6	Dual Training and dual prediction for polarity classification	SVM and Naive Bayes	Dual training and dual prediction (DTDP)
7	A 3D Hand Tracking Design for Gesture Control in Complex Environments	3D hand tracking design	It segments hands out of entire image and also facilitates depth estimation of tracked hands in real-time by dual camera systems.
8	Hand tracking and Gesture Recognition	Kalman filter and derived Scale Invariant Feature Transform (SIFT).	It presents a method for tracking and recognizing hand gestures by extracting unique invariant features from gestures.

No.	Reference	Techniques	Description
9	Hand Position Tracking Using a Depth Image from a RGB-d Camera	RGB image based on the skin color Hand Tracking	The algorithms can be used for natural user interfaces, the guidance of the end effector of an industrial robot and hand segmentation.
10	Indian Sign Language Recognition: Database Creation, Hand Tracking and Segmentation	YcbCr based skin color model	This algorithm works on motion tracking, edge detection and skin color detection.

Table 4.2: Literature Survey

CHAPTER 5

PROBLEM DEFINITION AND SCOPE

5.1 GOALS

- Understanding existing sentiment analysis approaches.
- Study corpus based, lexical based and semantic based techniques.
- Understanding unigram, bigram, trigram and combination of them for modeling purpose.
- Training the model with naive bayes, support vector machine, maximum entropy.
- Applying this learned model to the test dataset.
- Evaluating the results generated by classifiers.

5.2 OBJECTIVES

Please refer Chapter 1, Section 1.7 on Page 2

5.3 STATEMENT OF SCOPE

- Preprocessing the reviews
- Classify reviews into two polarities.
- Evaluate the classification accuracy by each classifier.

5.4 SOFTWARE CONTEXT

5.4.1 Scikit-learn

Scikit-learn (formerly scikits.learn) is a free software machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, random forests, gradient boosting, k-means and DBSCAN, and is designed to interoperate with the Python numerical and scientific libraries NumPy and SciPy.

5.4.2 NumPY

NumPy is the fundamental package for scientific computing with Python. It contains among other things:

- A powerful N-dimensional array object
- Sophisticated (broadcasting) functions
- Tools for integrating C/C++ and Fortran code
- Useful linear algebra, Fourier transform, and random number capabilities

Besides its obvious scientific uses, NumPy can also be used as an efficient multi-dimensional container of generic data. Arbitrary data-types can be defined. This allows NumPy to seamlessly and speedily integrate with a wide variety of databases.

5.4.3 Natural language toolkit

NLTK is a leading platform for building Python programs to work with human language data. It provides easy-to-use interfaces to over 50 corpora and lexical resources such as WordNet, along with a suite of text processing libraries for classification, tokenization, stemming, tagging, parsing, and semantic reasoning, wrappers for industrial-strength NLP libraries, and an active discussion forum.

5.4.4 Matplotlib

Matplotlib is a Python 2D plotting library which produces publication quality figures in a variety of hardcopy formats and interactive environments across platforms. Matplotlib can be used in Python scripts, the Python and IPython shell, the jupyter notebook, web application servers, and four graphical user interface toolkits. You can generate plots, histograms, power spectra, bar charts, errorcharts, scatterplots, etc., with just a few lines of code.

CHAPTER 6

DISSERTATION PLAN

6.1 PURPOSE OF THE DOCUMENT

This document specifies and estimates various risks associated with this project and states how they are handled. It also states the project plan in terms of task and their dependency.

6.2 TECHNICAL CONSTRAINTS

- To build a classification module that distributes data and execution among spark executors.
- To fetch candidate's tweets in a CSV format and store it in Alluxio.

6.3 DISSERTATION ESTIMATES

6.3.1 Reconciled Estimates

6.3.1.1 Cost Estimates

No cost is required for tools and software as open source softwares are used.

6.3.1.2 Time Estimates

Calendar time required: 11 months.

6.3.1.3 Dissertation Resources

- People : Single Person
- Hardware resources used are mentioned in Chapter 6, Section 6.5 on Page XX
- Software resources used are mentioned in Chapter 6, Section 6.6 on Page XX

6.4 RISK MANAGEMENT

This section discusses dissertation risks and the approach to managing them.

6.4.1 Risk Identification

For risks identification, review of scope document, requirement specifications and schedule is done. Answers to questionnaire revealed some risks. Following risk identification questionnaire has been referred.

- Are requirements fully understood by the software engineering team and its customers?
- Have customers been involved fully in the definition of requirements?
- Do end-users have realistic expectations?
- Does the software engineering team have the right mix of skills?
- Are project requirements stable?
- Is the number of people on the project team adequate to do the job?
- Do all customer/user constituencies agree on the importance of the project and on the requirements for the system/product to be built?

6.4.2 Risk Analysis

The risks for the dissertation are analyzed within the constraints of time and quality. Risk can be as follows:

- Out of memory error, while creating model and training the model.
- Review text contains unrecognized characters.
- Out of memory error, while predicting on the test dataset.

Please refer Table 6.1, 6.2 and 6.3 for detail description.

Table 6.1: Risk Table

ID	**Risk Description**	**Probability**	**Impact**		
			Schedule	**Quality**	**Overall**
1	Out of Memory	High	Low	High	High
2	Unrecognized characters	Low	Medium	High	Medium
3	Out of Memory	Low	Medium	High	High

Table 6.2: Risk Probability Definitions

Probability	Value	Description
High	Probability of the occurrence is	>75%
Medium	Probability of the occurrence is	26% - 74%
Low	Probability of the occurrence is	25%

Table 6.3: Risk Impact Definitions

Impact	Value	Description
Very High	>10%	Schedule impact or Unacceptable quality
High	5%-10%	Schedule impact or Some parts of the project have low quality
Low	<5%	Schedule impact or Barely noticeable degradation in quality Low Impact on schedule or Quality can be incorporated

6.4.3 Overview of Risk Mitigation, Monitoring and Management

Please refer Table 6.4, 6.5 and 6.6 for detail description.

Table 6.4: Risk 1

Risk ID	1
Risk Description	Out of memory error, when training Model
Category	Configuration
Source	Software Requirement Specification Document
Probability	High
Impact	High
Response	Mitigate
Strategy	Changing number of features resolves this issue.
Risk Status	Occurred and Resolved

Table 6.5: Risk 2

Risk ID	2
Risk Description	Unreconized characters
Category	Configuration
Source	Software Requirement Specification Document
Probability	Low
Impact	Low
Response	Mitigate
Strategy	Convert all characters into unicode format
Risk Status	Occurred and Resolved

Table 6.6: Risk 3

Risk ID	3
Risk Description	Out of memory error, when predicting
Category	Development Environment
Source	Software Requirement Specification Document
Probability	Low
Impact	Low
Response	Mitigate
Strategy	Using sparse matrix to represent text
Risk Status	Occurred and Resolved

6.5 STAFF ORGANIZATION

6.5.1 Team Structure

- Internal Guide : Prof. A. G. Phakatkar
- Student : Kaushik S. Hande

6.5.2 Management Reporting and Communication

The progress of dissertation is reported once in a month.

6.5.3 Dissertation Task Set

Major tasks in the Dissertation stages are -

Task 1 : Requirement

1. Define problem statement
2. Identify scope, requirements
3. Related mathematical model

Task 2 : Design

1. Identifying of key objects, functional relation
2. UML diagrams and functional dependency graph
3. System design

Task 3 : Implementation

1. GUI Implementation
2. MongoDB Implementation
3. Cloud Installation

Task 4 : Testing

1. Unit testing
2. Integration testing
3. System testing

Task 5 : Integration and Maintenance

1. Integration
2. Maintenance

Please refer figure 6.1 Task Network.

6.5.4 Task Network

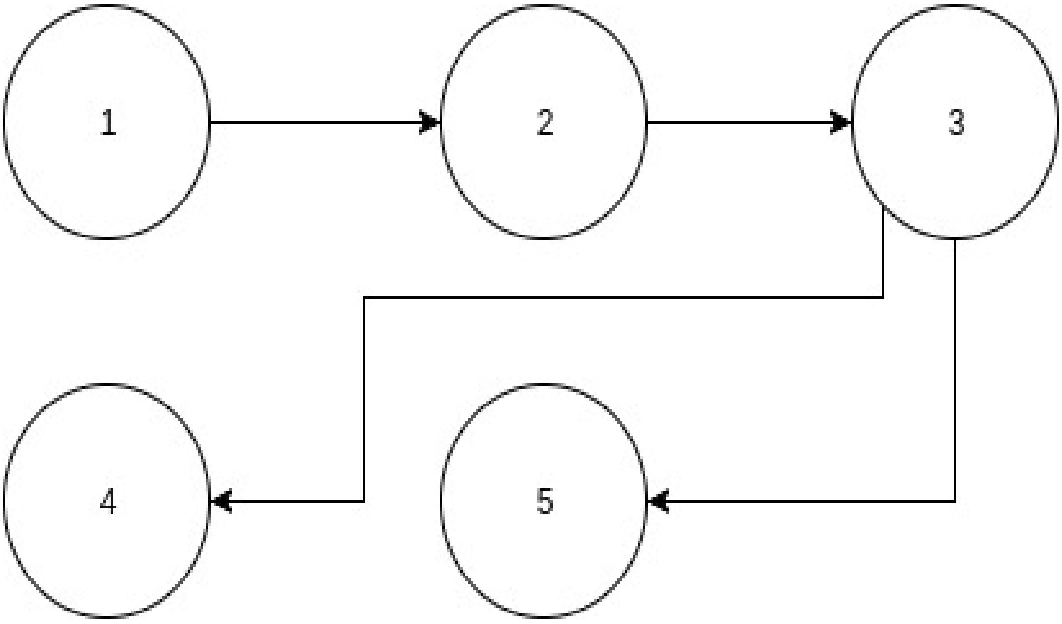

Figure 6.1: Task Network

6.5.5 Timeline Chart

Please refer Annexure B, Table B.1 on Page 66 for all Dissertation Tasks.

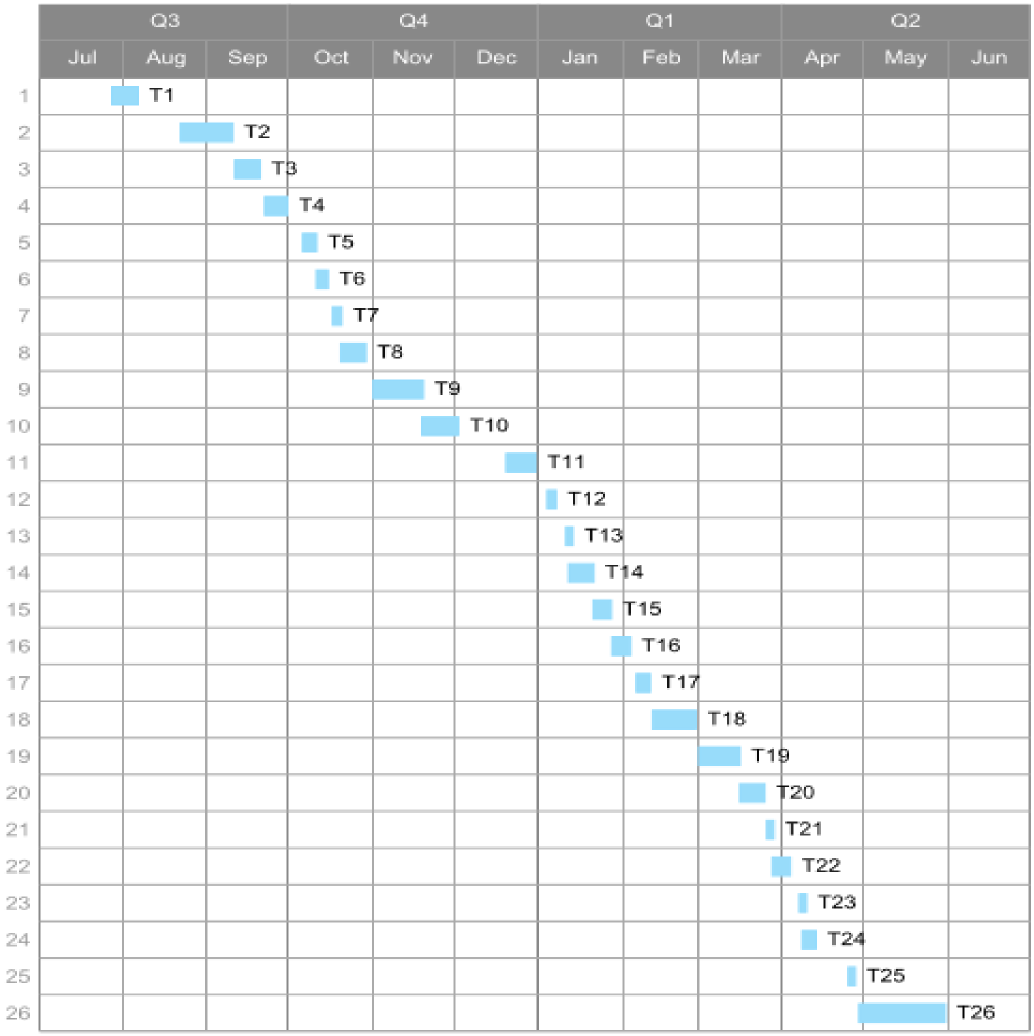

Figure 6.2: Timeline Chart

CHAPTER 7

SOFTWARE REQUIREMENT SPECIFICATION

7.1 INTRODUCTION

The aim of this document is to specify the software requirements for classification of movie reviews.

7.2 PURPOSE AND SCOPE OF THE DOCUMENT

The purpose of the document is to enlist various software requirements to build the system. This document has functional and non-functional requirements for the software being developed.

7.3 OVERVIEW OF RESPONSIBILITIES OF DEVELOPER

The responsibilities of a developer includes gathering of information about the classification libraries, that can be used to design and develop the system to categorize movie reviews. The developers responsibilities include:

- Planning for dissertation (Scheduling)
- Designing of system (High Level Design Document)
- Coding of system (Implementation)
- Testing of system (Test Cases)

7.4 PRODUCT OVERVIEW

System builds classifier models for classification of reviews. Different functionality of the system are :

- Candidate Registration - It shows a registration page that candidate uses to registers for one of the organizations.
- Candidate Data Downloading - It allows a Product Admin to download candidate's tweets from Twitter.
- Candidate Profiling - Profiles are formed for specific candidate after analyzing tweets in the form of column graph. It shows percentage of emotional and

polarity categories scores. Scores are shown year-wise, month-wise and day-wise.

- Candidate Comparison - It allows Product Admin to compare two candidates with respect to their emotional and polarity scores achieved. Comparison graphs of candidates are shown year-wise, month-wise and day-wise.
- Managing Candidate's Details and Data - It allows Product Admin to delete candidate's data or details or both.

7.5 HARDWARE RESOURCES USED

7.5.1 Software Requirements

- Python 2.7.6
- Rstudio Version 0.99.893
- R version 3.3.2
- Operating Systems:
  - Windows XP, 7, 8, 10
  - Linux(Any flavor)
  - Mac OS

7.5.2 Hardware Requirements

- Intel(R) Core(TM) i3 CPU @ 2.90GHz or later, width : 64 bits
- Memory : 4 GB DDR3 or more
- Capacity : 1697MHz or more
- Cores : 4 or more
- PCI Express Gigabit Ethernet Controller, Size: 100Mbit/s, Capacity: 1Gbit/s, Width: 64 bits
- Hard Disk : 500 GB (EXT4 Primary/Logical Partition)

7.6 FUNCTIONALITY

- Download movie reviews from imDb dataset.
- Import the moview review dataset into python environment using csv package.
- Convert the text reviews into matrix form.
- Remove the stopwords from reviews.
- Show positive and negative polarity score for test reviews.
- Compare classifiers for accuracy of classification.

7.7 INPUT

- Dataset that consists of movie reviews and their corresponding labels.
- List of stopwords which play no role in classification.

7.8 OUTPUT

- Classification of each test review into positive or negative.
- Percentage of accuracy achieved in classification.
- Comparison of accuracies obtained by each classifier.

7.9 MAJOR CONSTRAINTS

- To store movie reviews as input in csv file format.
- To execute classifiers in configured environment.
- To train the model for polarity classification.

7.10 APPLICATIONS

- Businesses and organisations which require consumer opinions to do with products they market and services they produce.

- Individuals who make decisions to purchase products or services based upon word of mouth or on-line reviews, or to find public opinion, e.g. concerning politics or local issues.
- On-line advertising where in social media, an organisation may place an advertisement in response to a favourable review of a product, or a rival product could be advertised upon receipt of a bad review
- Opinion retrieval for general searches of opinions
- HR Analytics.

7.11 USAGE SCENARIO

A use case represents a particular functionality of a system. Hence, use case diagram is used to describe the relationships among the functionalities and their internal/external actors. This section provides various usage scenarios for the system to be developed.

7.11.1 User Profiles

Actors of the system are Candidate, Product Administrator, Storage System, Database System and Web Interface.

- **Candidate** : Actor registers for a specific organization giving twitter URL and other details to Database using Web Interface.
- **Product Administrator** : Actor manages registers candidates, downloads tweets of a candidate, manages several behavioral assessment tests, profiles and compares candidates.
- **Storage System** : Actor stores tweets of candidates downloaded by Product Administrator.
- **Database System** : Actor stores emotional, polarity scores and other details of candidates.
- **Web Interface** : Actor displays candidate's emotional and polarity graphs according to year, month and day. It also allows Product Admin to download

candidate’s data and manage behavioral assessment tests.

7.11.2 Use Cases

Table 7.1 gives Use Cases for system to be developed.

Table 7.1: Use Cases

Sr. No.	Use Case	Descriptions	Actors	Assumptions
1	Candidate Registration	Candidate has to registers for a specific organization giving necessary details and saved to Database.	Candidate, Database System	Provided details are correct
2	Candidate Data Download	Product Admin fetches candidate's details from database, Extract screen name from Twitter URL, downloads candidate's data and store it in storage system.	Candidate, Database System, Storage System, Product Admin	Data is downloaded properly.
3	Candidate Comparison	Product Admin chooses two candidates for comparison based on emotional and polarity values.	Candidate, Database System, Product Admin, Web Interface	Comparison between two candidates are shown in the form of graph.
4	Candidate Profiling	Candidate are profiled based on set of rules defined by Product Admin.	Candidate, Database System, Product Admin, Web Interface	Profile results are displayed in the form of column graph.
5	System	Overall system description	Candidate, Product Admin, Database System, Web Interface, Storage System	System is functional

7.11.3 Use Case Views

7.11.3.1 Candidate Registration

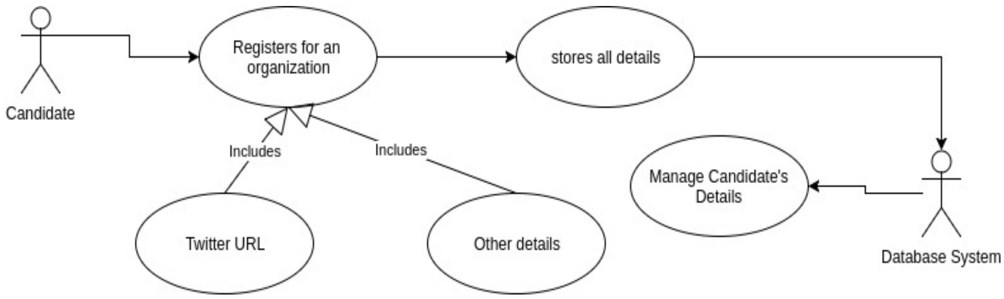

Figure 7.1: Use Case : Candidate Registration

7.11.3.2 Candidate Data Download

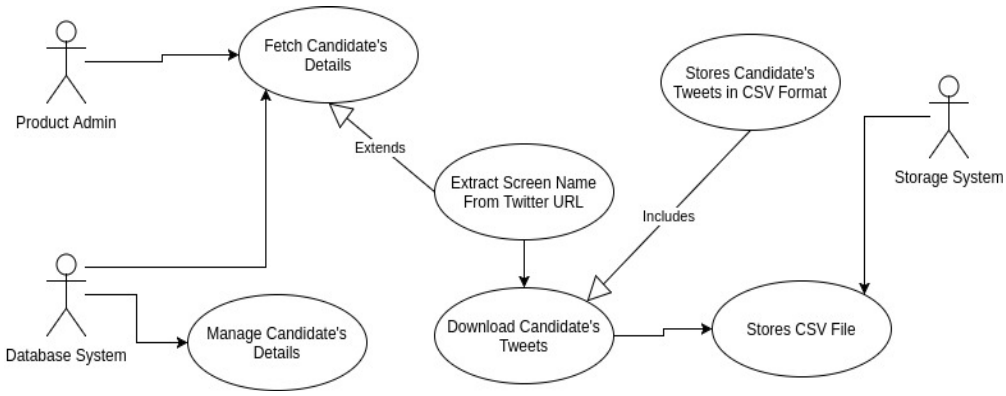

Figure 7.2: Use Case : Candidate Data Download

7.11.3.3 Candidate Comparison

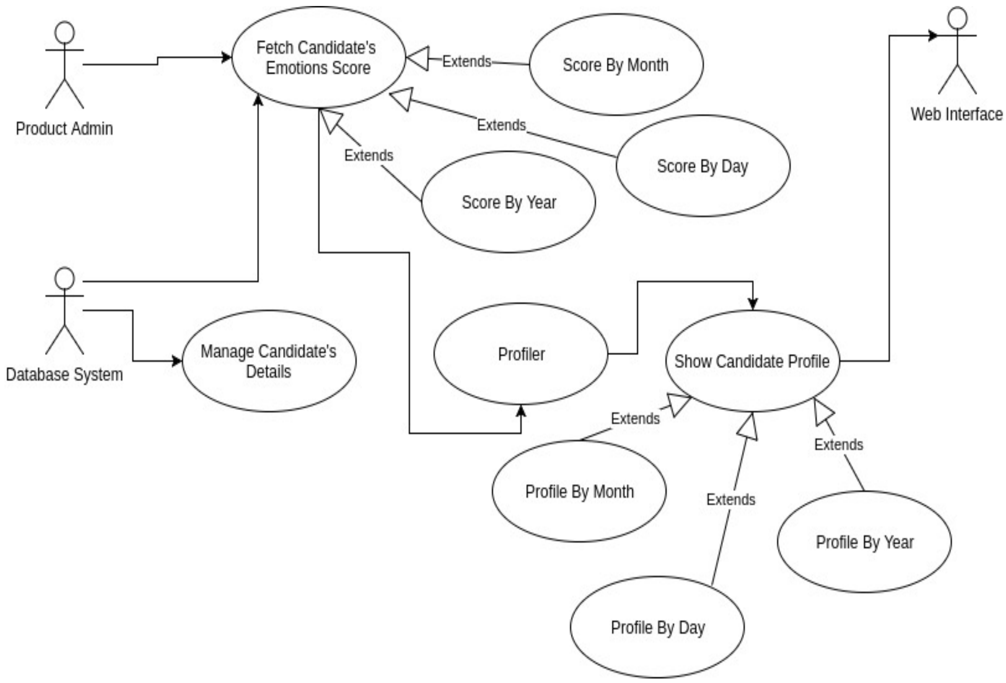

Figure 7.3: Use Case : Candidate Comparison

7.11.3.4 Candidate Profiling

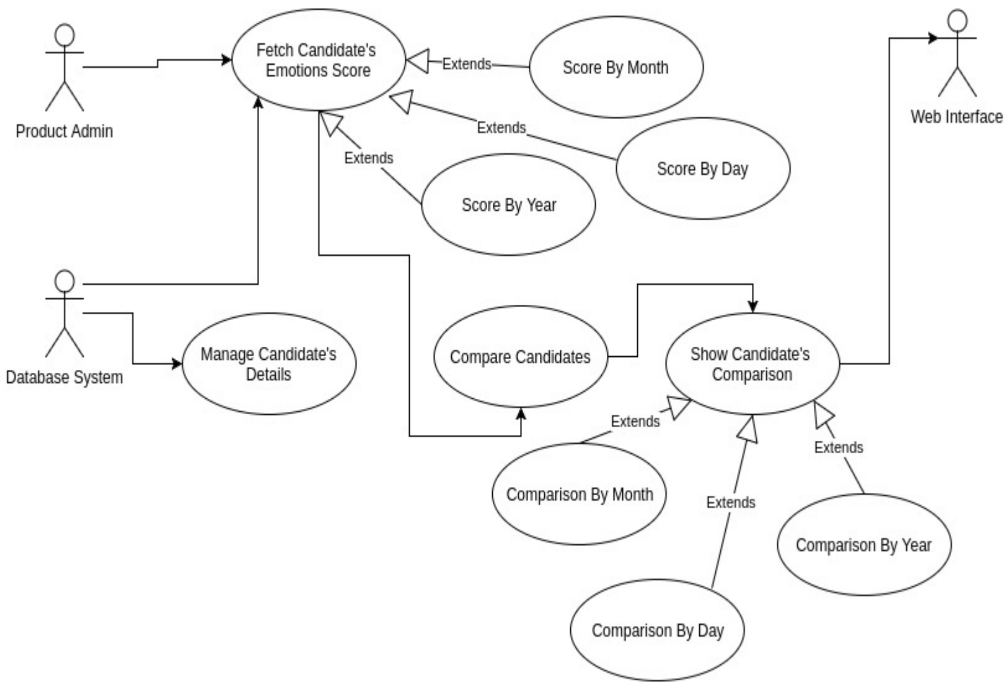

Figure 7.4: Use Case : Candidate Profiling

7.11.3.5 System Use Case

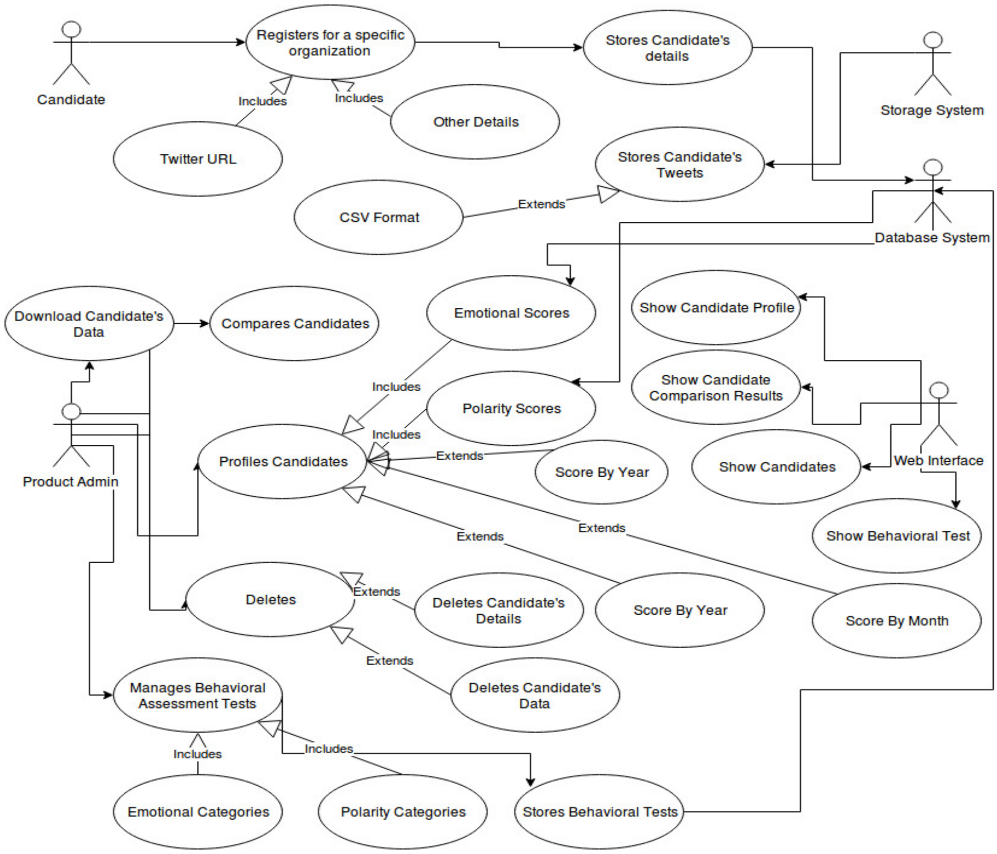

Figure 7.5: Use Case : Overall System

7.12 BEHAVIORAL MODEL AND DESCRIPTION

This section contains details about events and associated behaviour of the system which is shown using diagram below.

7.12.1 Activity Diagram

Activity diagram is a flow chart to represent the flow form one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched or concurrent. The purpose of activity diagrams is to capture the dynamic behaviour of the system.

Description : As shown in figure 7.6, Product Administrator downloads candidate's tweets from Twitter for analysis. Tweets are stored in a CSV format in Alluxio data storage. For analysis to take place, candidate's download status is check. If it is true, load candidate's CSV file and proceed with analysis else download candidate's tweets. For document classification, initially model existence is checked. If it exists, then load model for document classification else proceed with training phase. The training phase consists of document preprocessing, feature extraction and saving model in Alluxio. Classifier uses this trained model for document classification. Candidates are profiled and compared based on their document classified into emotional and polarity categories.

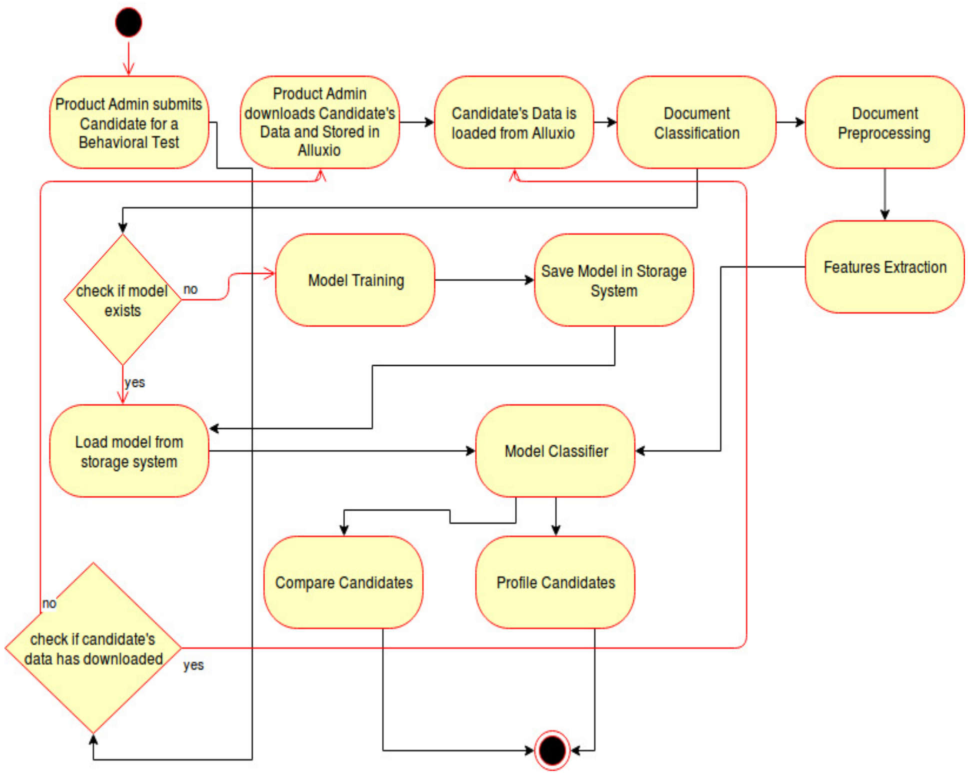

Figure 7.6: Activity Diagram

7.13 FUNCTIONAL MODEL AND DESCRIPTION

This section describes data flow diagrams (DFD) of the proposed system. There are three types of DFDs explained in the section. These diagrams explain the system in brief.

7.13.1 Data Flow Diagram

7.13.1.1 Level 0 Data Flow Diagram

In the level 0 DFD as shown in figure 7.7, Candidates registers into Behavioral Assessment System. System performs analysis and generates reports for a registered candidate. They are displayed to Product Admin.

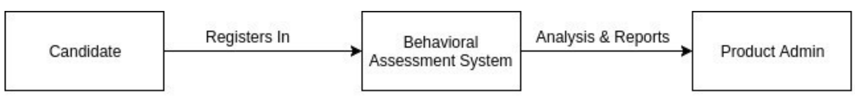

Figure 7.7: Level 0 DFD

7.13.1.2 Level 1 Data Flow Diagram

In the level 1 DFD as shown in figure 7.8, Candidate's tweets are fetched from Twitter by Product Admin using Web Interface. Tweets are stored in Alluxio for storage. They are retrieved by Web Application modules for analysis and report generation.

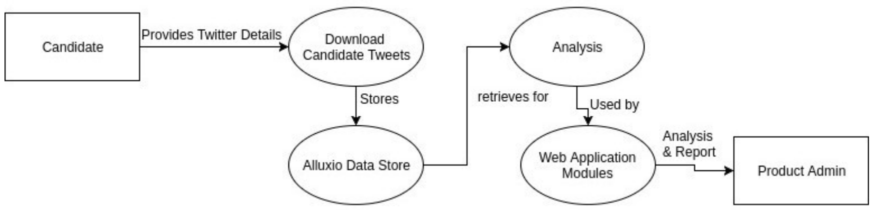

Figure 7.8: Level 1 DFD

7.13.1.3 Level 2 Data Flow Diagram

In the level 2 DFD as shown in figure 7.9, Candidate's tweets are retrieved from Alluxio Data Storage and preprocessed. Features are extracted for Classification. It classifies tweets of a candidate to emotional and polarity categories. Candidate are profiled and compared based on these categories. Profile and comparison results are displayed to Product Admin using Web Interface.

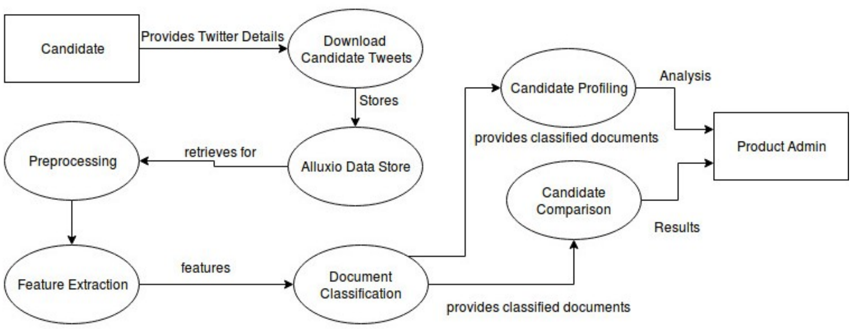

Figure 7.9: Level 2 DFD

7.14 NON-FUNCTIONAL REQUIREMENTS

7.14.1 Availability

Required libraries must be installed and loaded in the python environment with the required configurations. Dataset must be downloaded from specified url [17].

7.14.2 Scalability

The system should be scalable to classify reviews even if the training and test data are increased. System can comfortably handle reviews dataset upto 25000 reviews.

7.14.3 Performance

The system must be interactive and delays involved must be less. There should be no immediate delays for every action and response of the system. Training time increases as the training data increases. It takes 4 to 5 seconds in training the dataset. Training increases further when bigram and trigram models are used.

7.14.4 Usability

The system should be easy to handle and process requests efficiently. System's functions are designed to use with ease and provide results. Results are presented in the form of graphs and are easy to comprehend.

7.14.5 Reliability

The system should efficiently analyze movie reviews entirely and give correct classification result. It should be reliable to perform classification effectively on any review dataset.

7.14.6 Maintainability and Changeability

The system is made up of different independent modules that can be modified to correct faults, improve performance or other attributes, or adapt to a changed environment. System can be improved for new features and will be able to include new requirements.

CHAPTER 8

DETAILED DESIGN DOCUMENT

8.1 INTRODUCTION

This document specifies the design that is used to fetch candidate tweets in CSV format, classifies individual tweets into emotion and polarity categories, profiles candidates based on predefined rules and predict future behavior of candidate.

8.2 BEHAVIORAL MODELING

Candidate's tweets are classified into emotional and polarity categories. Over the course of candidate's social media presence, profile deviates that forms a iterative behavioral model.

- **Emotional Categories**
  - Anger
  - Disgust
  - Joy
  - Love
  - Fear
  - Sadness
  - Surprise
- **Polarity Categories**
  - Positive
  - Negative
  - Offensive

Candidate's tweets are collected and classified based on these categories. They are profiled and compared based on emotional and polarity categories.

8.3 ARCHITECTURAL DESIGN

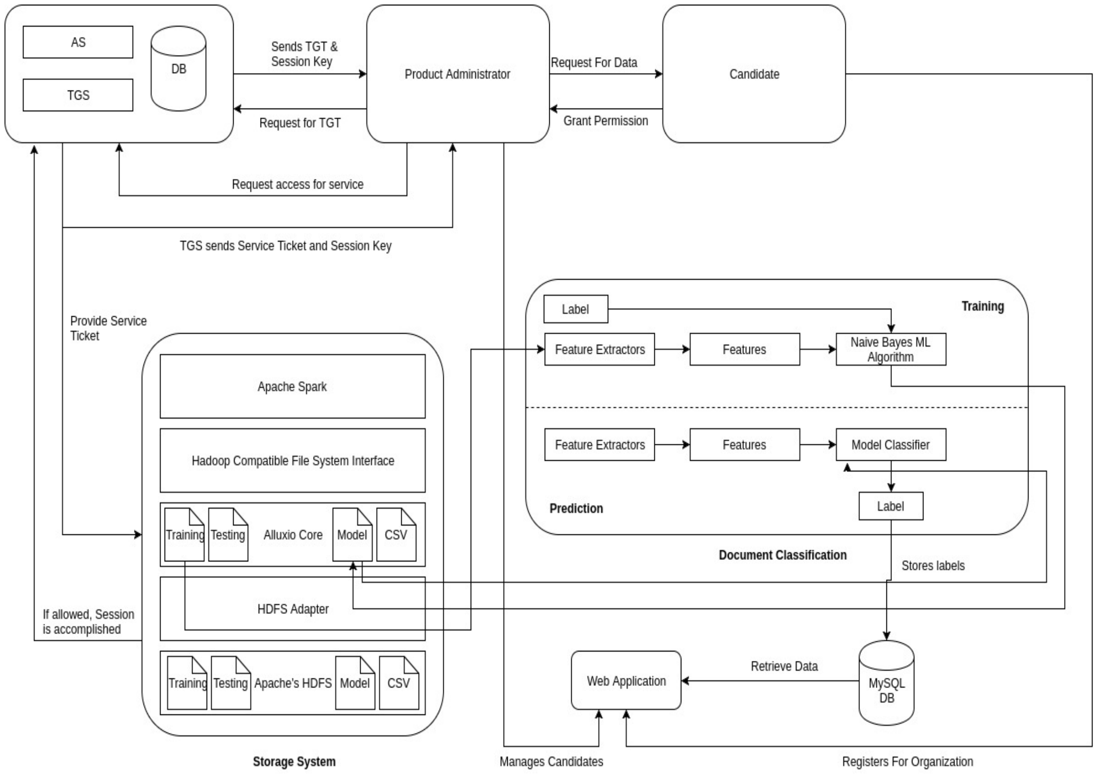

Figure 8.1: Proposed System Architecture

Figure 7.1 shows architectural design of proposed system. Following are important components in the system :

- Storage System : Alluxio, memory centric storage system is used as main storage system which stores in-memory data. Apache's HDFS is used as an underFS storage system. Candidate's CSV files, trained models, testing and training dataset is stored in Alluxio.
- Computation Framework : Apache's Spark is used for computations. Document classifier is built in Spark. It access candidate's CSV, trained model files from Alluxio. Spark's Mlib is used for training Naive Bayes model which is stored in Alluxio after training.
- Document Classifier : First, Naive Bayes model is trained by training dataset of emotional and polarity categories and saved in Alluxio. After training, prediction stage occurs. It access candidate's CSV file from Alluxio and classifies document on each candidate's tweet. Predicted labels on each tweet is stored

in MySQL database with year, month and day.

- Candidate registers for specific organization using web application, providing twitter details. Product Administrator access web application to download candidate's tweets from Twitter, manages candidates, profiles and compares candidates.
- Kerberos is used for mutual authentication between storage system components and document classifier. It consists of Authentication Server, Database and Ticket Granting Service. Product Administrator requests Key Distribution Center for a valid ticket before submitting Spark job. If no valid ticket found, operation is not permitted. With only valid ticket, candidate's tweets are analyzed.

8.4 CLASS DESIGN

It is a static diagram that represents the static view of an application. It is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. It describes the attributes and operations of a class and also the constraints imposed on the system.

Description : In figure 8.2, modules and their relationships are shown. Document classifier used for classifying candidate's tweets takes user identifier of candidate and candidate's CSV location in Alluxio as an input. Product Administrator fetches candidate's tweets from Twitter in a CSV format and store the file in Alluxio. Candidate's CSV file location is saved into database for further processing. After document classification, candidates can be profiled and compared based on their emotional and polarity categories. For both modules, results are displayed by year, month and day.

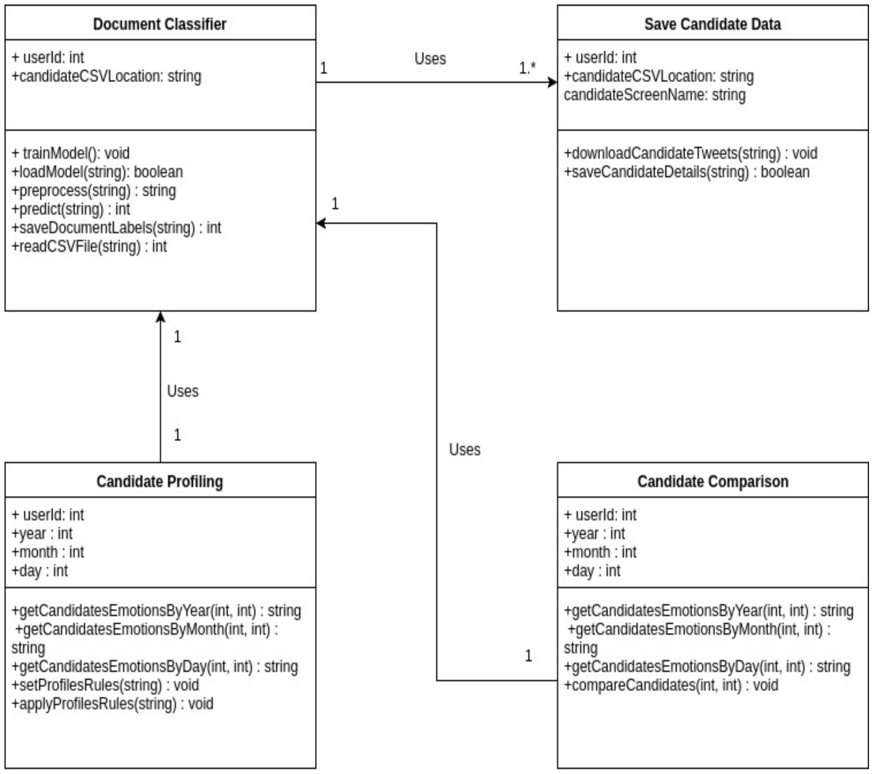

Figure 8.2: Class Diagram

8.5 COMPONENT DESIGN

It is used to model the physical aspects of a system. It is also used to visualize the organization and relationships among components in a system. It does not describe the functionality of the system but it describes the components used to make those functionalities.

Description : Figure 8.3 describes primary components of the system. A web application provides candidate's tweets to be fetched and candidates are profiled and compared functionality to Product Administrator. Product Admins are authenticated first before using any of the functionality. To download tweets of a specific candidate, he/she must register for that organization. Tweets are stored in Alluxio data storage in CSV format. Data storage is accessed by Spark application for fetching candidate's tweets for preprocessing and document classification.

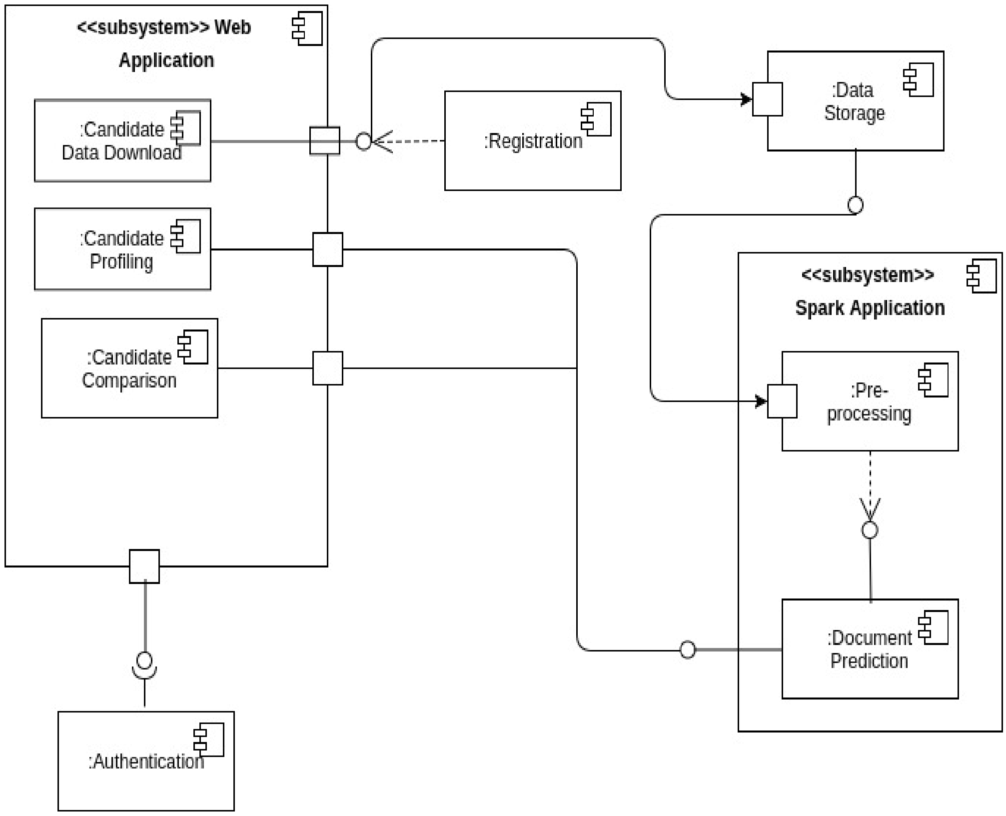

Figure 8.3: Component Diagram

8.6 DEPLOYMENT DESIGN

It is used visualize the topology of the physical components of a system, where the software components are deployed. It describes the static deployment view of a system, consisting of nodes and their relationships.

Description : In figure 8.4, Web application designed in Laravel PHP runs on Apache Tomcat Server. Process component of Symphony is used to run spark application. 4-Node cluster is formed for spark application execution. It requires data from Alluxio data storage system. Alluxio is specifically compiled for Spark for data access. Web application retrieves candidate's emotional and polarity scores from database.

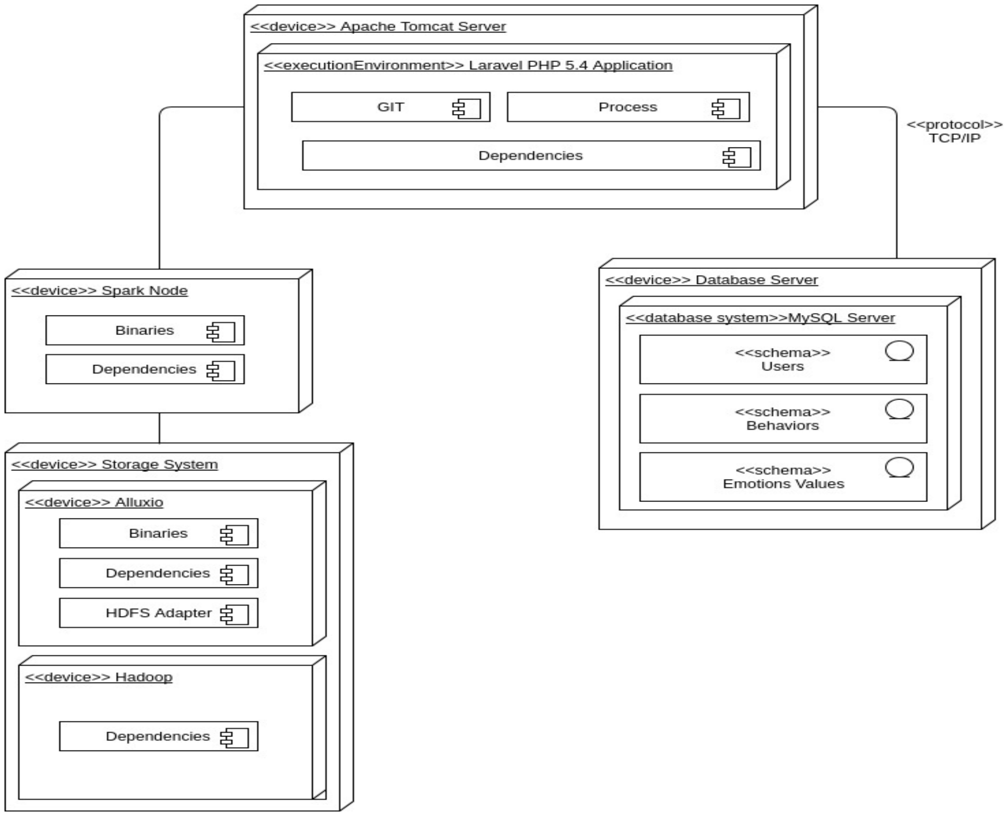

Figure 8.4: Deployment Diagram

CHAPTER 9

IMPLEMENTATION DETAILS

9.1 INTRODUCTION

This section describes implementation of the system, required libraries and dependencies needed for components of the system and use of implementation strategy.

9.2 ALGORITHM

9.2.1 Document Classification

Input : Candidate's CSV file and Unique Identifier.

Output : Candidate's tweets classified into emotional and polarity categories.

```
Initialize numFeatures to 10000
Initialize emotionalTrainingDataPath to /emotion-training
Initialize polarityTrainingDataPath to /polarity-training
Initialize emotionModel to /emotion-model
Initialize polarityModel to /polarity-model
Input candidateCSVLocation
Input candidateUserId

If emotionModel exists
    load emotionModel
Else
    load Category Data from emotionalTrainingDataPath
    transform Category Data using Hashing Transform for numFeatures
    attach label to Category Data
    save emotionModel to /emotion-model
    load emotionModel from /emotion-model
If polarityModel exists
    load polarityModel
Else
    load Category Data from polarityTrainingDataPath
    transform Category Data using Hashing Transform for numFeatures
```

attach label to Category Data

save polarityModel to /polarity-model

load polarityModel from /polarity-model

Read CSV file from candidateCSVLocation

While CSV file contains some rows

Read Columns from CSV file

Column 0 contains 'Date'

Column 1 contains 'Actual Tweet'

Split Column 0 into year, month and day

Transform Column 1 using Hashing Transform for numFeatures

Classify Column 1 with respect to emotionModel and get emotionLabel

Save emotionLabel into database with year, month and day

Classify Column 1 with respect to polarityModel and get polarityLabel

Save polarityLabel into database with year, month and day

9.2.2 Candidate Profiling

Input : Candidate's User Identifier.

Output : Candidate Profiles

Input Candidate's User Identifier.

Retrieve years from Database based on UserId

Retrieve months from Database based on UserId

Retrieve days from Database based on UserId

Calculate Percentage of Categories for all years

Retrieve emotionsValues from Database based on UserId for all years

Retrieve polarityValues from Database based on UserId for all years

While candidate has emotion category and value in a year

Calculate Total number of documents in a year

Calculate Total number of document for a specific category

Calculate Percentage for a specific category

While candidate has polarity category and value in a year

Calculate Total number of documents in a year

Calculate Total number of document for a specific category for a year

Calculate Percentage for a specific category for a year

Calculate Percentage of Categories for all months

Retrieve emotionsValues from Database based on UserId for all months group by years

Retrieve polarityValues from Database based on UserId for all months group by years

While candidate has emotion category and value in a month for specific year

Calculate Total number of documents in a month

Calculate Total number of document for a specific category

Calculate Percentage for a specific category for a month

While candidate has polarity category and value in a month for specific year

Calculate Total number of documents in a month

Calculate Total number of document for a specific category for a year

Calculate Percentage for a specific category for a month

Calculate Percentage of Categories for all days

Retrieve emotionsValues from Database based on UserId for all days in a month for a specific year

Retrieve polarityValues from Database based on UserId for all days in a month for a specific year

While candidate has emotion category and value for all days in a month for specific year

Calculate Total number of documents in a day

Calculate Total number of document for a specific category

Calculate Percentage for a specific category for a day

While candidate has polarity category and value for all days in a month for specific year

Calculate Total number of documents in a day

Calculate Total number of document for a specific category

Calculate Percentage for a specific category for a day

9.3 MODULES

9.3.1 Candidate's Tweets Fetched From Twitter

- This modules retrieves candidate's tweets from Twitter. Input to module is candidate's screen name.
- Every user in twitter has unique screen name which can be used to retrieve his/her tweets using Twitter APIs and OAuth.
- OAuth 2 is an authorization framework that enables application to obtain limited access to user accounts on an HTTP service such as Facebook and Twitter.
- A Twitter application is created, python module connects to Twitter application using consumer secret key and consumer key. Twitter application also generates access key and access secret key which decides validity of tweets access. Tweets are fetch and store it in CSV format.
- The CSV file of every candidate is pushed to Alluxio storage system. Python's Tweepy library is used for implementation of this module.
- A Product Administrator uses candidate data downloading functionality to trigger this module giving candidate's screen name as an input.

9.3.2 Document Classification

- This module takes candidate's CSV location in Alluxio and unique identifier as an input.

- This module is written in Scala and executes as an spark job on multi-node spark cluster. This means, spark job uses resources of multiple connected nodes for faster processing.
- Alluxio is memory centric distributed storage system that provides candidate's CSV file to spark job. Spark's Machine Learning library is used for implementation of Naive Bayes Classifier.
- Initially, model is trained by training dataset that consists of emotional categories of 21429 records and polarity categories of 8797 records. Both models are saved to and loaded from Alluxio.
- Every candidate's tweet is classified into one of the emotional and polarity categories. A database insertion operation push classified documents along with user identifier to MySQL.

9.3.3 Web Application

Web application is designed and built in Laravel PHP 5.4. High charts is used for showing candidate results in the form of column graphs. There are different modules in web application -

- Candidate Profiling : Module shows each candidate's emotional and polarity categories percentage year-wise, month-wise and day-wise.
- Candidate Comparison : Two candidates are compared by emotional and polarity categories percentage. Results are shown year-wise, month-wise and day-wise.
- Candidate Data Downloading Functionality : It enables product administrator to download tweets of a certain candidate. It uses Process component of Symphony to execute python script that fetches tweets from Twitter.
- Storage Analyzer : It shows storage space used by candidate's CSV files, training and testing dataset and saved trained models in Alluxio. Alluxio's local file system commands are used to retrieve space occupied. Registration It allows a candidate to register himself/herself to a certain organization.

- Assessment creation and deletion : Assessments are created, deleted and updated by product administrator.

9.4 DATASET

Dataset comprises of tweets from Twitter. It has to be collected for every candidate that needs to be assessed for behavioural assessment. There is significant latency and load on server in fetching such information of candidate using Twitter API.

After the dataset is collected, it needs to be stored in underFS storage layer. Movement of huge dataset to storage layer requires additional I/O writes and communication overhead. But, by using proposed system, writes operation to storage layer is significantly lesser.

For document classification, a training and testing dataset is required. Training records for emotional and polarity categories are mentioned in Table 9.1 and 9.2.

Table 9.1: Emotional Training Dataset

Emotion Category	Training Records
Anger	1572
Joy	8276
Disgust	761
Love	216
Sadness	3853
Surprise	3912
Fear	2839
Total	**21429**

Table 9.2: Polarity Training Dataset

Polarity	Training Records
Positive	2007
Negative	4783
Disgust	2007
Total	**8797**

9.5 SNAPSHOTS

CHAPTER 10

TEST SPECIFICATION

10.1 INTRODUCTION

This document explains the test plan and testing strategy for modules in a system. Following modules needs to be tested -

- Fetching candidate's tweets from Twitter and store it in CSV format.
- Classification of candidates in emotional and polarity categories.
- Web application that sends requests, collect responses and act based on responses.

10.1.1 Goals and Objectives

- To validate candidate's tweets in a CSV after fetching it from Twitter.
- To check accuracy of different document classifiers.
- To validate predicted document labels of candidate's tweets.
- To validate candidate's profiles and comparison results.
- To validate execution distribution among multiple connected nodes.

10.1.2 Statement of Scope

Testing will be done on individual modules of a system. Testing will be carried out for several different candidates. Finally, system as a whole is tested for correctness of results.

10.1.3 Major Constraints

Testing is done manually. For testing the accuracy of document classifiers, testing dataset is formed and used. Total number of multiple connected nodes is limited to four nodes.

10.2 TEST PLAN

10.2.1 Modules to be Tested

- Candidate's tweets fetched from Twitter module.

- Candidate's tweets classification module.
- Web Application.

10.2.2 Testing Strategy

10.2.2.1 Unit Testing

Unit testing has been done for all the individual modules. While doing unit testing different parameter has been considered and according to input to the system different output is recorded. After recording output of unit testing different solution are applied to pass the test. This is carried out as white box testing.

- To test candidate's tweets are fetched and stored in a CSV format.
- To test classification module's effectiveness and prediction of document labels.
- To test different classifier's accuracy.
- To test candidate profiles and comparison's results. Result must be shown in the form of graph.
- To test candidate's documents analysis occurs distributively, using resources of multiple connected nodes.

10.2.2.2 Integration Testing

Once unit testing is complete for individual modules, all the modules are integrated together and tested for functional correctness. While doing integration testing, developer has kept in mind few constraints which need to be achieved in order to get desired results.

- Web application requests needs to be accepted by classification module and response back with predicted document labels for candidate's tweets.
- Web application uses candidate's data download functionality to fetch candidate's tweets from Twitter and store it in Alluxio.
- Web Application retrieves candidate's analysis results and forms a column b ar graph.

- Candidates are compared for seven emotional categories and results are shown as a graph.

10.2.2.3 Validation Testing

Validation testing is carried out to test the entire work flow and input validation. This is carried out as a black box testing. In this project validation testing has been conducted on different modules.

- To validate candidate's tweets in a CSV after fetching it from Twitter.
- To validate predicted document labels of candidate's tweets.
- To validate candidate's profiles and comparison results.
- To validate execution distribution among multiple connected nodes.

10.2.2.4 System Testing

- To test if all GUI elements are shown properly in a web interface.
- To test if spark application is executed atomically using web interface.
- To test if spark application is able to access data stored in Alluxio.
- To test if Kerberos is integrated into Spark and Alluxio.

10.2.2.5 GUI Testing

Front End of the system is designed as web application, runs on local server. Data visualization is provided by High charts.
Following modules needs to be tested in a web application -

- Candidate's data downloading functionality
- Assessment creation and deletion
- Storage Analyzer
- Candidate's profiles
- Candidates comparison

10.2.2.6 High Order Testing

It includes carrying out performance testing by checking complete running time of document classification on multinode spark cluster.

10.2.3 Test Procedure

10.2.3.1 Unit Testing

Table 10.1: Unit Test Cases

Sr. No.	**Test Case Name**	**Test Case Objective**	**Test Case Input**	**Test Case Result**
1	Log In	Product Admin should be able to log in for correct organization only.	Username & Password	Pass
2	View Candidates	Product Admin should be able to view all candidates registered for organization.	Candidate's Details	Pass
3	Create Behavioral Assessment Test	Product Admin should be able to create customized behavioral test.	Emotional and polarity categories's values	Pass
4	Candidate Registration	Candidate should be able to register for specific organization.	Candidate's Details	Pass
5	Manage Candidate Data	Product Admin should be able to delete candidate's records and tweets stored in CSV format	Candidate's User Identifier	Pass

10.2.3.2 Integration Testing

Table 10.2: Integration Test Cases

Sr. No.	Test Case Name	Test Case Objective	Test Case Input	Test Case Result
1	Candidate Profiling	Product Admin submits candidate's tweets for profiling. Emotional and polarity scores must be generated after analysis.	Candidate's Unique Identifier	Pass
2	Candidate Comparison	Product Admin selects two candidates for comparison. Comparison occurs based on emotional and polarity scores.	Unique User identifier of both candidates.	Pass

10.2.3.3 Validation Testing

Table 10.3: Validation Test Cases

Sr. No.	Test Case Name	Test Case Objective	Test Case Input	Test Case Result
1	Candidate Data Validation	Candidate's details are first validated against rules set.	Candidate's details.	Pass
2	Candidates listing for Behavioral Assessment Tests	Candidates whose tweets are fetched should only appear in list for test.	Candidate's Twitter Data Download Status	Pass
3	Candidate listing for Profiling and Comparison	Candidates whose tweets have been analyzed should only appear in list for profiling and comparison.	Candidate's Behavioral Assessment Completion Status	Pass

10.2.3.4 System Testing

Table 10.4: System Test Cases

Sr. No.	Test Case Name	Test Case Objective	Test Case Input	Test Case Result
1	GUI elements displayed properly.	To show all GUI elements in Web Interface properly	Views	Pass
2	Executing Spark Application	Web application executes Spark application for tweets classification into emotional and polarity categories.	Candidate's Unique User Identifier	Pass
3	Alluxio data access	Spark application accesses tweets stored in CSV format from Alluxio.	Candidate's Unique User Identifier	Pass
4	Kerberos Authentication	Product Admin must be authenticated first and should have valid ticket, before submitting candidate's tweets for analysis. submitting	Product Admin's Keytab	Pass

CHAPTER 11

DATA TABLES AND DISCUSSIONS

11.1 KERBEROS SUB-SYSTEM ANALYSIS

A client may submit only one MR job or multiple jobs at a same time. The number of communication rounds and total number of protocol messages generated for different numbers of MR-Request-Component can be calculated. As number of jobs submission increases so does the communication overhead. There are three different use cases-

- One client can submit one job for submission.
  - Total number of components requesting access to MR Resource for this case-

$$1(C)+n(Comp) \tag{11.1}$$

  - Total number of communication rounds from Authentication Server to MR-Request-Component-

$$2(R)+2n(R) \tag{11.2}$$

  - Total number of communication rounds from MR-Request-Component to MR-Resource-Component are-

$$1(R)+n(R) \tag{11.3}$$

  - Total number of messages sent are-

$$6+6n \tag{11.4}$$

- One client can submit multiple jobs for submission.
  - Total number of components requesting access to MR Resource for this case-

$$1(C)+z\times n(Comp) \tag{11.5}$$

  - Total number of communication rounds from Authentication Server to

MR-Request-Component-

$$2(R) + z \times 2n(R) \tag{11.6}$$

- Total number of communication rounds from MR-Request-Component to MR-Resource-Component are-

$$1(R) + z \times n(R) \tag{11.7}$$

- Total number of messages sent are-

$$6 + 6n \times z \tag{11.8}$$

- Multiple client can submit multiple jobs for submission.
  - Total number of components requesting access to MR Resource for this case-

$$y(C) + y \times z \times n(Comp) \tag{11.9}$$

  - Total number of communication rounds from Authentication Server to MR-Request-Component-

$$y \times 2(R) + y \times z \times 2n(R) \tag{11.10}$$

  - Total number of communication rounds from MR-Request-Component to MR-Resource-Component are-

$$1(R) + y \times z \times n(R) \tag{11.11}$$

  - Total number of messages sent are-

$$6y + 6y \times n \times z \tag{11.12}$$

One Client (C) has on Job(J) with n Components (Comp) per job. One Client has z jobs with n Components (Comp) per job. y Clients, each client has z jobs with n

Component (Comp) per job.

For a N number of MR components requests access to MR-Resource-Component per job, number of communication rounds and messages for an authentication process can be calculated as,

- For communication round from Authentication Server to MR-Request-Component Communication

$$2N \times R(2N \times Request + 2N \times Response) \tag{11.13}$$

- For communication round from MR-Request-Component to MR-Resource-Component

$$N \times R(N \times Request + N \times Response) \tag{11.14}$$

11.2 ALLUXIO STORAGE SYSTEM PERFORMANCE ANALYSIS

For writes throughput, Alluxio outperforms MemHDFS by 110x and for reads throughput, 2x greater than MemHDFS. It's read throughput is higher than write throughput. This happens due to machine configured with optimized memory hardware, leaving more bandwidth for reads.

It also improves the end-to-end latency of a realistic work flow by 4x. Introducing check-pointing algorithm guarantees recovery cost and resource allocation strategies for re-computation under resource schedulers.

Alluxio architecture consists of two layers-lineage and persistence. The lineage layer provides high throughput I/O and tracks the sequence of jobs that have created a particular data output. The persistence layer persists data onto storage without the lineage concept.

CHAPTER 12

CONCLUSION

We proposed a hand gesture based human computer interaction system that provides a natural way to interact with computer.The hand is first segmented by using skin color information and then tracked using 'Camshift' tracker with Kalman filter, then fingertips are located on the contour of the segmented hand and single gestures drawn from fingertips are recognized. For pointing, click, right click, zoom, drag and window closing various gestures have been allocated.

CHAPTER 13

FUTURE ENHANCEMENTS

The research can be extended to explores the relationship between behaviors and psychological theories to determine candidate's language style or social tendencies. The system only fetches 3200 tweets of a candidate for analysis. For more precise profile deviation, more tweets should be fetched for Behavioral Analytics.

CHAPTER 14

REFERENCES

[1] R. Xia, F. Xu, C. Zong, Q. Li, Y. Qi and T. Li, “ Dual sentiment analysis: Considering two sides of one review, ” in IEEE transactions on knowledge and data engineering, vol. 27, no. 8, pp. 2120 - 2133, Aug. 2015.

[2] S. Das and M. Chen, “ Yahoo! for Amazon: Sentiment extraction from small talk on the web, ” *Management science* , Vol.53, Issue no.9, pp.1375-1388, 2007.

[3] Pang, L. Lee, and S. Vaithyanathan, “ Thumbs up?: Sentiment classification using machine learning techniques, ” *Proceedings of the ACL-02 conference on Empirical methods in natural language processing,* pp. 79-86, 2002.

[4] B. Pang and L. Lee, “ Opinion mining and sentiment analysis, ” *Foundations and Trends in Information Retrieval,* vol. 2, no. 1-2, pp. 1-135, 2008.

[5] R. Xia, T. Wang, X. Hu, S. Li, and C. Zong, “ Dual Training and Dual Prediction for Polarity Classification,” *Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL - 02),* pp. 521-525, 2013.

[6] P. Turney, “ Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews, ” Proceedings of the Annual Meeting of the Association for Computational Linguistics (ACL), pp. 417-424, 2002.

[7] M. Li and C. Huang, “ Sentiment classification considering negation and contrast transition, ” *Proceedings of the Pacific Asia Conference on Language, Information and Computation (PACLIC),* pp. 307-316, 2009.

[8] Li, S. Lee, Y. Chen, C. Huang and G. Zhou, “ Sentiment Classification and Polarity Shifting, ” *Proceedings of the International Conference on Computational Linguistics (COLING),* pp. 635-643, 2010.

[9] D. Turney and Michael L. Littman, “ Un-supervised learning of semantic orientation from a hundred-billion-word corpus. ”*Technical Report EGB-1094, National Research Council Canada,* arXiv preprint cs/0212012, 2002.

[10] Yuan Wang, Zhaohui Li, Jie Liu, Zhicheng He, Yalou Huang and Dong Li, " Word Vector Modeling for Sentiment Analysis of Product Reviews " *Natural Language Processing and Chinese Computing 2014,* pp. 168-180, 2014.

[11] Xia, Rui and Wang, Cheng and Dai, Xinyu and Li, Tao, " Co-training for Semi-supervised Sentiment Classification Based on Dual-view Bags-of-words Representation " *Association for Computational Linguistics (ACL 1),* pages 10541063, 2015.

[12] Na, J.C., Sui, H., Khoo, C., Chan, S., and Zhou, Y., " Effectiveness of Simple Linguistic Processing in Automatic Sentiment Classification of Product Reviews "*Proceedings of the Eighth International ISKO Conference* pp. 49-54, 2004.

[13] Rui Xia, Feng Xu, Jianfei Yu, Yong Qi and Erik Cambria, " Polarity shift detection, elimination and ensemble: A three-stage model for document-level sentiment analysis " *Information Processing & Management 52, no. 1,* pp. 36 - 45, 2016

[14] Rui Xia, Chengqing Zong and Shoushan Li, " Ensemble of feature sets and classification algorithms for sentiment classification "*Information Sciences 181, no. 6* pp. 11381152, 2011

[15] Anderson Uilian Kauer and Viviane P. Moreira, " Information retrieval for sentiment polarity prediction, " *Expert Systems With Applications 61,* pp. 282 - 289, 2016

[16] Yuming Lin, Jingwei Zhang, Xiaoling Wang and Aoying Zhou, " An Information Theoretic Approach to Sentiment Polarity Classification "*Proceedings of the 2nd joint WICOW/AIRWeb workshop on web quality, ACM Lyon France* , pp. 35 - 40, 2012

[17] http://boston.lti.cs.cmu.edu/classes/95-865-K/HW/HW3/movie-pang02.zip

ANNEXURE A

PAPERS PUBLISHED

A.1 PAPER TITLE

Sentiment Analysis using Machine Learning Algorithms: A Survey

A.1.1 IJIRCCE Certification


ISSN (Online) 2278–1021
ISSN (Print) 2319–5940

IJARCCE

INTERNATIONAL JOURNAL OF ADVANCED RESEARCH IN COMPUTER AND COMMUNICATION ENGINEERING

ISO 3297:2007 Certified

Impact Factor 5.947

IJARCCE

CERTIFICATE OF PUBLICATION

SAGAR S. PATIL

Student, Department of Computer Engineering, Pune Institute of Computer Technology, Pune, India

Published a research paper entitled

Sentiment Analysis, Emotion Mining & Authentication Methods in Hadoop: A Survey of Approaches

in IJARCCE, Volume 6, Issue 3, March 2017

DOI 10.17148/IJARCCE.2017.6369

Certificate No: 10.17148-1/2-A
www.ijarcce.com

Tejass Publisheers
ORGANIZATION

Editor-in-Chief
IJARCCE

Figure A.1: IJARCCE Certificate

A.2 PAPER TITLE

Sentiment Analysis using Original and Reversed Reviews

A.2.1 cPGCON Certificate

A.2.2 cPGCON Review

ANNEXURE B

DISSERTATION PLANNER

Table B.1: Dissertation Task Set

Task Title	**Dissertation Task**
T1	Study of Domain - Machine Learning and Natural Language Processing
T2	Identification of problem in existing systems
T3	Review of Literature
T4	Building Mathematical Model
T5	Report On Scheme of Implementation
T6	Identification of Prerequisites and Installation
T7	Configuring python and python package installer pip in the system
T8	Study of various machine learning algorithms and its implementation in python
T9	Studying libraries in python required for implementation
T10	Downloading and extracting reviews from IMDb movie datasets provided for resear
T11	Removing stopwords, punctuation marks, numbers etc.
T12	Report Preparation
T13	Dissertation Project Stage I Presentation
T14	Document Preprocessing
T15	Creating Bag of words model from movie reviews.
T16	Spliting the dataset into training and test dataset
T17	Train machine learning classifiers using bag of words model.
T18	Create unigram, bigram, trigram variations of model
T19	Train machine learning classifiers using this model.
T20	Cpgcon Paper Presentation
T21	Predictive Model Construction
T22	Model Testing
T23	Experimental results, Analysis and Validation of results
T24	Project Review with Demonstration
T25	Report Validation and Submission, Report Submission