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. Rushikesh Kasar*** *for providing various resources such as well-equipped lab with all needed software platforms, continuous Internet connection, for my dissertation work.*

Subhash K. Nevhal
(M.E. Computer Engg.)

List of Figures

List of Tables

ABSTRACT

Human Computer Interaction(HCI) is an active focus of research. Considering the motivation, need and challenges, we are proposing fingertip detection system with vision approach for Desktop handling operation.

The advantages of such system will be to avoid physical devices such as keyboard, mouse etc.

INDEX

CHAPTER 1

SYNOPSIS

DISSERTATION TITLE

SENTIMENT ANALYSIS USING ORIGINAL AND REVERSED REVIEWS

INTERNAL GUIDE

Prof. A. G. Phakatkar

PROBLEM STATEMENT

"To develop Vision based Human Computer Interaction System using Fingertip Tracking for Desktop Handling operation."

OBJECTIVES

- Design and implement fingertip detection algorithm.
- Design and implement fingertip tracking algorithm.
- Design and implement virtual keyboard.
- Execute Windows API for execution.
- Test and analyze the system.
-
-
-

HYPOTHESIS

Vision based Hand detection and tracking can determine cursor movements using operating system's system calls. Gestures determined by using computer vision techniques can be utilized to generate callback functions to implement various system functions. CamShift algorithm with Kalman filter can be used to determine accurate and exact location of cursor to give better Human Computer Interaction using virtual keyboard with web-cam.

RELEVANT MATHEMATICS ASSOCIATED WITH DISSERTATION

Mathematical Model

$S = \{s, e, I, F, F_s, F_t, F_f, O|\phi\}$

where,

s = start state

e = end state

F = Feature Vector for G

G = set of Gestures

$F_s =$ Fingertip detection function

$F_t =$ Fingertip tracking function

$F_f =$ Feature Extraction and Matching function

I = set of Inputs

I = {M,G}

where,

M = {RC,LC,MC..Drag}

RC = Right Click

LC = Left Click

MC = Middle Click

G = {VK,M,SG}

where

VK = Virtual Keyboard System

$SG = \{VK_0, VK_1, RCG, LCG, MCG, DragG, G1..G10\}$

where SG = Special Gestures

$VK_0 =$ Virtual Keyboard off

$VK_1 =$ Virtual Keyboard on

RCG = Right Click Gesture

LCG = Left Click Gesture

F = {KS,MS,GF}

where

KS = System Keyboard functions mapped to K

MS = System mouse functions mapped to M

GF = Gesture functions mapped to G

O = Output

O = {KSe,MSe,GFe,EMd}

where

KSe = API execution of Keyboard

MSe = API execution of Mouse

GFe = API execution of Gesture Functions

EMd = Display of Error Messages

Success: If $I_i = F_j, 1 < i <= |I|, 1 < j <= |F| ThenExecuteO_i$

Failure: Else No Execution (or print Error Messages)

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

AREA OF DISSERTATION

Computer Vision, Human Computer Interaction, Fingertip detection, Hand tracking and Segmentation, Hand Gesture Recognition.

ACM KEYWORDS

A Information Systems

- A.1 Retrieval tasks and goals
  - A.1.1 Sentiment analysis
  - A.1.2 Clustering and classification

B Information storage systems

- B.1 Storage architectures
  - B.1.1 Distributed storage

C Information systems applications

- C.1 Data Mining
  - C.1.1 Emotion Mining

D Computing methodologies

- D.1 Parallel computing methodologies
- D.2 Machine learning
  - D.2.1 Supervised learning by classification
    - D.2.1.1 Multinomial Naive Bayes
    - D.2.1.2 Random Forest
    - D.2.1.3 Support Vector Machines

E Security and privacy

- E.1 Cryptography
  - E.1.1 Kerberos Authentication Protocol

CHAPTER 3

INTRODUCTION

DISSERTATION IDEA

Today's organizations assess their external candidates, candidates they want to recruit and internal candidates or employees currently working, based on their compatibility to organization's culture and fitness in role associated to them. They use multiple tests that consists of several roles scenarios as well as capability questions for assessment. Assessing candidates behavior is equally important to determine mindset and thoughts to recent and trending topics. This can be done in two ways:

- Behavioral assessment test that comprises of some questions that is used to observe, describe and predict candidates behavior.
- Building a iterative behavioral profile by analyzing candidates profile on social media sites, their reaction to recent trending topics. Reactions can be mapped to emotion classes, which then give a more precise in-depth understanding of candidates behavior.

Assessment results along with inclusion of behavioral profiling of candidates is recorded by HR. This helps organizations in forming proper structure, planning, management, allocation of candidate to proper product development and operational efficiency.

Predictive analytics combined with proposed behavioral model gives organization essential information regarding internal candidates and helps in making key decisions like customized employment value proposition. Tweets from Twitter of candidates are collected, that becomes our input dataset for a particular candidate. Dataset persists in HDFS, thus provides information availability even if one of the connected node goes down. Only selective data from dataset is brought down to Alluxio, virtual distributed storage system.

Alluxio storage system stores in-memory data, so only limited data can be stored in the storage system. Therefore, Alluxio makes use of a optimized allocators and

evictors like Max Free Allocator or Greedy Allocator to move data between memory and underFS storage layer like HDFS. Candidate is classified to emotion classes based on sentiment analysiss results. We can see difference in the behavior of candidates as we are analyzing social media profile by months usage. t helps organizations to see patterns in behavior of candidate.

Our proposed model uses Naive Bayes Classifier for classification. Classifier's execution occurs as a Spark job which runs distributively on several connected commodity hardware configured nodes. Hadoop and Spark does not provide any authentication among its components while running spark job. Therefore, it is necessary to provide mutual authentication among server nodes, Hadoop components - Domain specific as well as infrastructure components. Mutual authentication is achieved by integrating Kerberos into Hadoop, Spark and Alluxio.

MOTIVATION OF DISSERTATION

In various organizations, candidates are assessed for criteria of suitability to organizations culture and their specific role. Behavior of a candidate plays a vital role in progress of organization. A candidate may excel at his role but do not have proper attitude towards his teammates, may degrade the quality and productivity of his entire team. Candidate may lie in behavioral assessment test but they do not hide their reactions on social media sites about current trending topics like political, scientific etc. By collecting those reactions, we can construct behavioral model of candidates and can predict future behavior.

CHAPTER 4

LITERATURE SURVEY

We studied the following related systems to discover their advantages, drawbacks and limitations and these are discussed below.

AFFECTIVE TEXT MINING

Previous work focused on mining affective content from text documents using sentiment classification. However, study does not explores extraction of social emotions from multiple social media sites and building a behavioral profile of a candidate that quicksilver with consideration of new posts to sites. The most related direction to work is emotion prediction and classification. Affective text aims to add notes giving explanation to predefined list of emotions and/or polarity orientation (positive, negative and neutral). Alm et al [1] explored text-based emotion prediction using supervised learning approaches. Strapparava and Mihalcea [2] evaluated methods for the automatic identification of six emotions. Tokuhisa et al [3] proposed a two step model for emotion classification using emotion-provoking event instances. Yang et al [4] investigated the emotion classification of blogs using machine learning techniques.

EMOTION TERM MODEL

A method to model the word-emotion associations is emotion-term model [14], which uses Naive Bayes with the assumption that words are independently generated from social emotion labels. The model needs to be extended to account not only social emotion labels but querying entire text to generate more precise emotion-term relationship. Emotion-term model generates each word w_i of document d in two sampling steps.

- sample emotion e_i according to the emotion frequency count
- sample a word w_i given the emotion under the conditional probability P($w|e$)

The model parameters can be learned by maximum-likelihood estimation. In particular, the conditional probability of a word w given an emotion e can be estimated as follows:

$$P(w|e) = \frac{|(w,e)|}{\sum|(w,e)|} \tag{4.1}$$

where $w^i \varepsilon W$, $|(w,e)|$ is the co-occurrence count between $w \varepsilon W$ and emotion $e \varepsilon E$ for all documents. It is formally derived based on the word and emotion frequency counts

$$|(w,e)| = S + \sum_{d \varepsilon D} \delta_{d,w} . \Upsilon_{d,e} \tag{4.2}$$

where S is a small smoothing constant that is set to 1. For predicting emotion on a new document b, Bayes theorem can be applied under the term independence assumption

$$P(e|d) = \frac{P(d|e)P(e)}{P(d)} \alpha P(d|e)P(e) \tag{4.3}$$

where P(e) is the a priori probability of emotion e. It can again be calculated by maximum likelihood estimation from the emotion distribution of the entire collection.

AUTHENTICATION METHODS FOR HADOOP

Multiple clients submit their MapReduce jobs for processing. Before submitting any jobs, a client needs to get authenticated by Authentication Server. Somu et al [5] provides a method which is symmetric key based. It uses single authentication factor, supports only gate-level authentication and has more communication overheads. Rubika method of authentication [6] is also designed to support client authentication only. Wei et al [7] ensures the authenticity of messages sent from one MR-job component to another. There is need of mutual authentication between MR components and MR infrastructure components. J. Zhao et al [8] supports the authentication of a client to MR application and authentication between pair of domain specific MR components.

GAP IDENTIFICATION THROUGH LITERATURE SURVEY

The following table shows the literature survey about different techniques of analyzing sentiment from social media text and emotion classification.

Table 4.1: Literature Survey

No.	Reference	Techniques	Description
1	Hand Detection Techniques to hand gesture Recognition for Natural Human Computer Interaction	Hand Detection using Lab Color Space and Mean Shift Algorithm	Better results on skin color detection using HTS algorithm streams.
2	Hand Gesture Recognition for Indian Sign Language	HSV color model and General Camshift algorithm	The Gesture Recognition System takes the input hand gestures through in-built web camera.
3	A Novel Projector-Camera Interaction System with the Fingertip	Prediction Method and triangulation Zhang's method	Projector and binocular vision system based on two cameras is applied for detecting the depth of fingertips and touch operation.
4	Hand gesture based user interface for computer using a camera and Projector	The YCrCb color space with single Gaussian and Camshift tracking algorithm	A hand gesture based human computer interaction system comprising of a webcam and a pocket projector.
5	Real-Time Robust Hand Tracking Based on Camshift and Motion Velocity	Improved Camshift algorithm with KLT tracking.	Probability of Bayesian skin color is used to refine the velocity of hand motion.
6	Gesture Recognition in Ego-Centric Videos using Dense Trajectories and Hand Segmentation	Dense trajectories extracted around hand regions	Dense features are extracted around regions selected by a new hand segmentation technique.
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

GOALS

- Understanding of existing Cultural Framework and Role Scenarios.
- Designing a web platform that carries out functionality of the system.
- Understanding of Storage File Systems Apache's HDFS and Alluxio.
- Integrating Alluxio with Apache's HDFS and Apache's Spark.
- Understanding Sentiment Analysis and Emotion Mining Approaches.
- Classifying candidate's documents into emotion categories.

OBJECTIVES

Please refer Chapter 1, Section 1.7 on Page XX

STATEMENT OF SCOPE

- Analyzing candidate's tweets distributively on multiple node Spark cluster.
- Classify candidate's tweets into seven emotion categories and three polarities.
- Integrating Alluxio core services with Apache HDFS and Apache Spark.
- Validates candidate's emotion classification.
- Validates candidates comparison by emotion categories.

SOFTWARE CONTEXT

Apache Spark

Spark [9] provides fast cluster computing system. It's high level APIs in Java, Python, Scala, Python and R are used. It also uses Hadoop client's libraries for performing operation on data stored in HDFS. Spark system in cluster mode is deployed to take advantage of commodity hardware by using Hadoop YARN as cluster manager. Spark with Mahout APIs can be used for calculating term frequencies, generating TFIDF vectors, writing and reading from HDFS and Alluxio and classification. By using Spark, more records in less time is classified efficiently.

Apache Hadoop

Hadoop's HDFS [10] stores huge amount of training data in HDFS for distributed processing. Mahout APIs can access data stored in HDFS for processing. It provides scalable environment and fault-tolerant file systems.

Alluxio

It provides memory centric design and unified name space for different storage system. Alluxio [11] improves processing speed for big data applications while providing a common interface of data access. Haoyuan Li et al. [12] explains how Alluxio speeds up read and write throughput.

Kerberos

To provide a level of security among Hadoop components like Resource Manager etc. Kerberos [13] sets up a Key Distribution Center. KDC has three main components. Kerberos Database which stores users and services identity. Authentication server resides as a separate physical server. It issues tickets upon initial request. Ticket Granting Service responsible for providing service tickets. By using these three components, communication is secured between core components of Alluxio, Spark and Hadoop.

Laravel PHP

A PHP framework that provides useful tools to securely built a web application with an ease. It provides authentication, amazing ORM, painless routing and powerful queue library.

CHAPTER 6

DISSERTATION PLAN

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.

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.

DISSERTATION ESTIMATES

Reconciled Estimates

Cost Estimates

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

Time Estimates

Calendar time required: 11 months.

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

RISK MANAGEMENT

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

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?

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, when training Naive Bayes Model.
- Spark executors were getting lost.
- Incorrect candidate results.

Please refer Table 5.1, 5.2 and 5.3 for detail description.

Table 6.1: Risk Table

ID	**Risk Description**	**Probability**	**Impact**		
			Schedule	**Quality**	**Overall**
1	Out of Memory	Low	Low	High	High
2	Executors Lost	Medium	Medium	High	Medium
3	Incorrect Results	Medium	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

Overview of Risk Mitigation, Monitoring and Management

Please refer Table 5.4, 5.5 and 5.6 for detail description.

Table 6.4: Risk 1

Risk ID	1
Risk Description	Out of memory error, when training Naive Bayes Model
Category	Configuration
Source	Software Requirement Specification Document
Probability	Low
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	Spark executors were getting lost
Category	Configuration
Source	Software Requirement Specification Document
Probability	Medium
Impact	Medium
Response	Mitigate
Strategy	Increasing heart beat interval and network timeout resolve this issue
Risk Status	Occurred and Resolved

Table 6.6: Risk 3

Risk ID	3
Risk Description	Incorrect Candidate Results
Category	Development Environment
Source	Software Requirement Specification Document
Probability	Medium
Impact	High
Response	Mitigate
Strategy	Debugging Candidate Results.
Risk Status	Occurred and Resolved

STAFF ORGANIZATION

Team Structure

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

Management Reporting and Communication

The progress of dissertation is reported twice in a month to internal guide and discussed with external guide once in a month.

DISSERTATION SCHEDULE

Dissertation Task Set

Major tasks in the Dissertation stages are -

- Configure Alluxio, Apache's HDFS and Apache's Spark.
- Fetch candidate's tweets from Twitter.
- Understand and build a document classifier.
- Profile candidates based on analysis done on candidate's tweets.
- Compare candidate's emotional and polarity categories.

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

Task Network

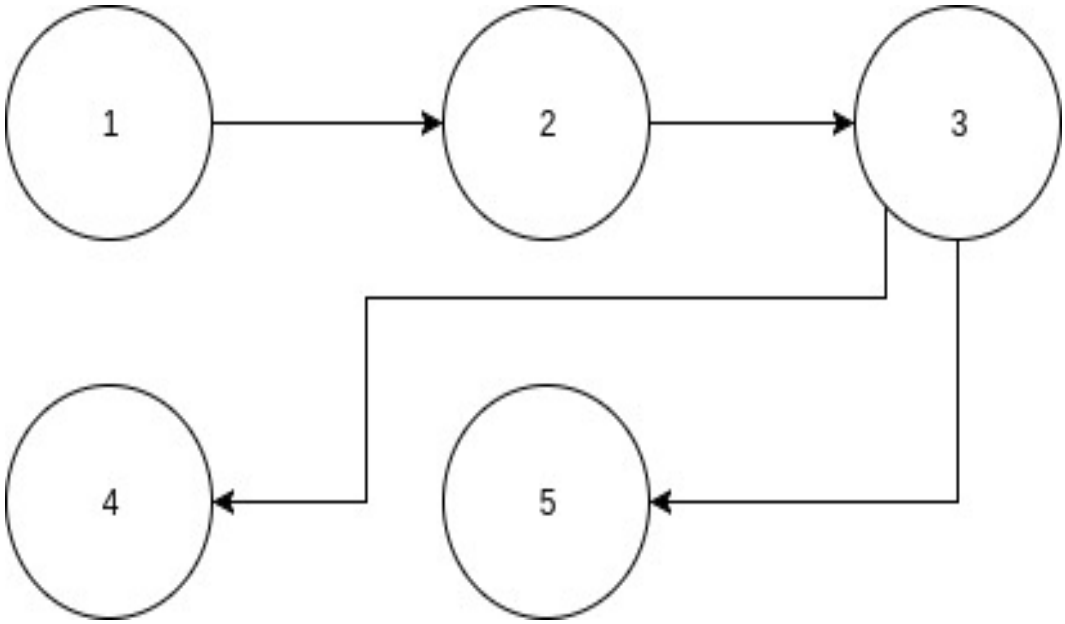

Figure 6.1: Task Network

Timeline Chart

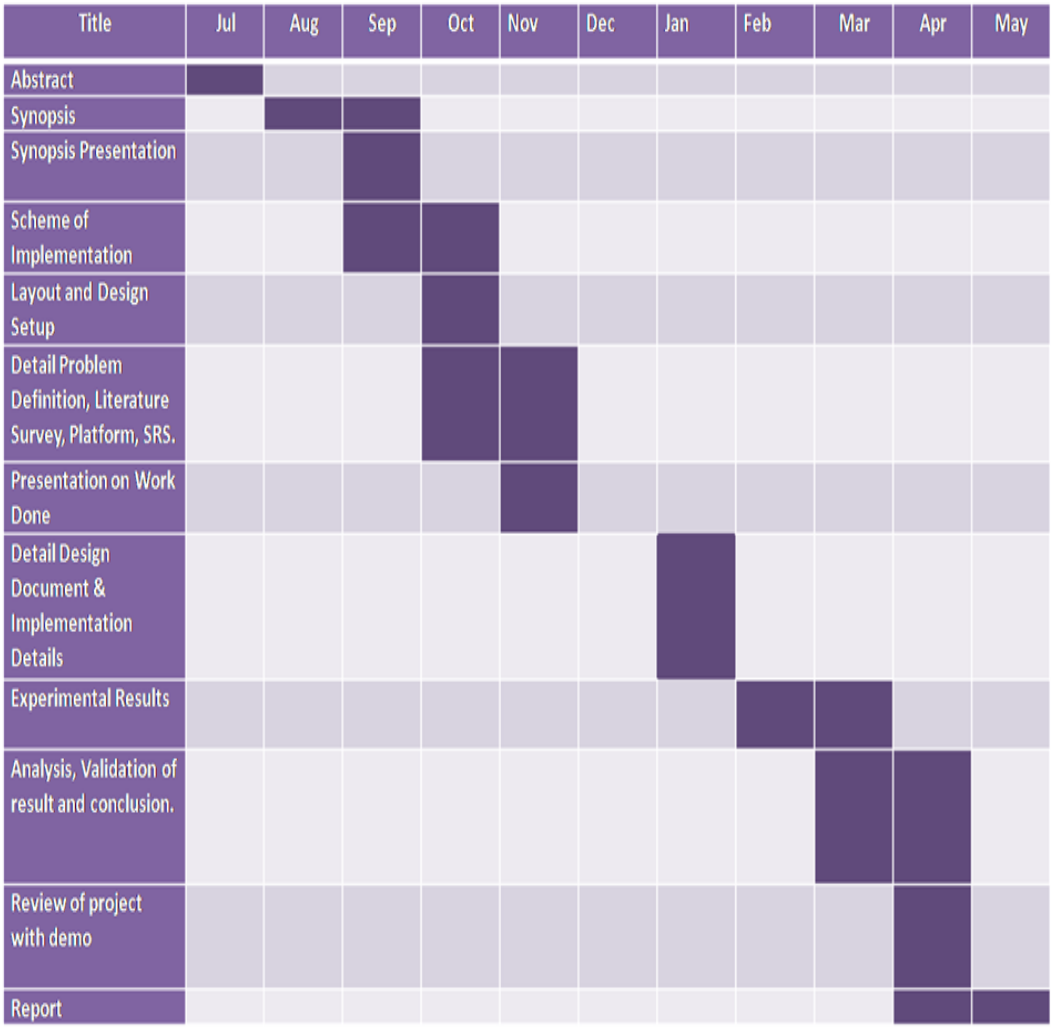

Figure 6.2: Timeline Chart

CHAPTER 7

SOFTWARE REQUIREMENT SPECIFICATION

INTRODUCTION

The aim of this document is to specify the software requirements for building a iterative behavioral model by analyzing candidate's tweets.

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.

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 candidate's tweets. The developers responsibilities include:

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

PRODUCT OVERVIEW

System builds a iterative behavioral model by fetching candidate's tweets and classifies them into one of the emotional categories and polarity categories. 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.

HARDWARE RESOURCES USED

4-Node Cluster with following configuration -

- Intel(R) Core(TM) i5 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)

SOFTWARE RESOURCES USED

- OpenCV 3.0.0 or later
- Python 2.7.6 or later
- Ubuntu 14.04 or later
-
-

FUNCTIONALITY

- Fetch candidate's tweets from Twitter using Twitter API.

- Show behavioral profile of candidates.
  - Show percentage of emotional and polarity categories for specific candidate year-wise, month-wise and day-wise.
  - Show profile deviation of specific candidate.
  - Show positive, negative and offensive polarity score for specific candidates.
- Show percentage of relevance to particular user defined requirements.
- Compare two candidates for set of emotional and polarity categories.

INPUT

- Dataset that consists of Twitter's tweets.
- User specific requirements for finding relevance in candidate's profile.
- Candidate's emotional and polarity categories score for comparison and profile generation.

OUTPUT

Candidate's behavioral profile that shows:

- Percentage of emotion and polarity categories
- Relevance to particular user defined requirements.
- Percentage of profile deviation.
- Candidate's comparison graph.

MAJOR CONSTRAINTS

- To store candidate's data as an input in CSV format.
- To store candidate's CSV file in Alluxio.
- To form Spark Standalone Cluster.
- To execute Spark Classifier job in configured environment.
- To train the model for emotional and polarity categories and store it in Alluxio.

APPLICATIONS

- External candidates behavioral assessments visiting for recruitment drive or invited by managers, organized by multiple organizations.
- Internal candidates profile evaluation of multiple organizations.
- HR Analytics.

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.

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.

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

Use Case Views

Candidate Registration

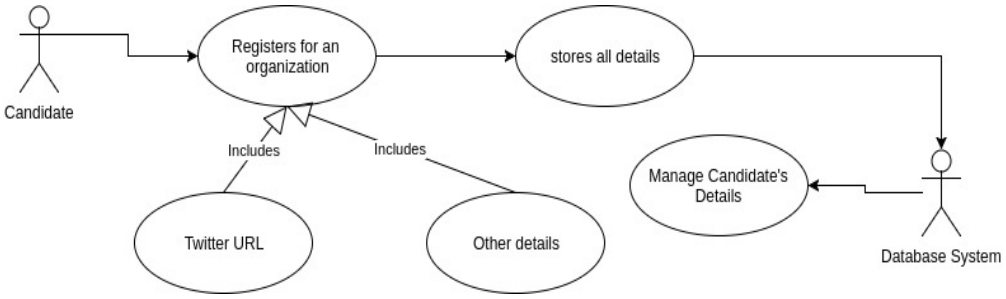

Figure 7.1: Use Case : Candidate Registration

Candidate Data Download

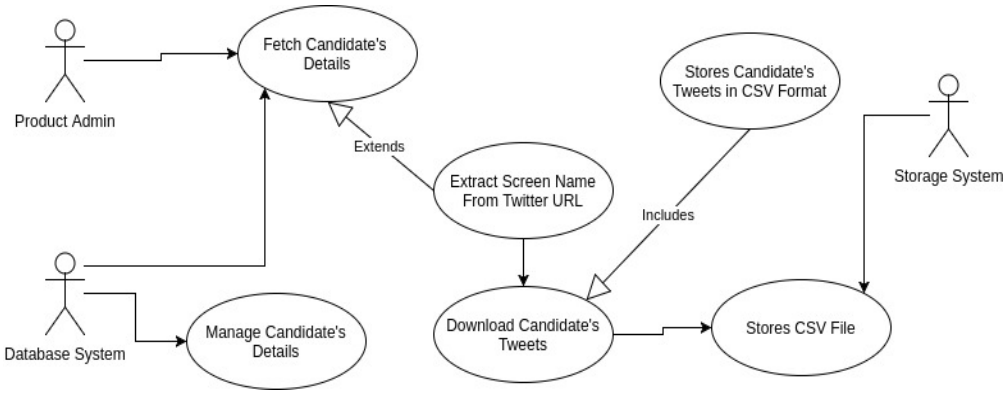

Figure 7.2: Use Case : Candidate Data Download

Candidate Comparison

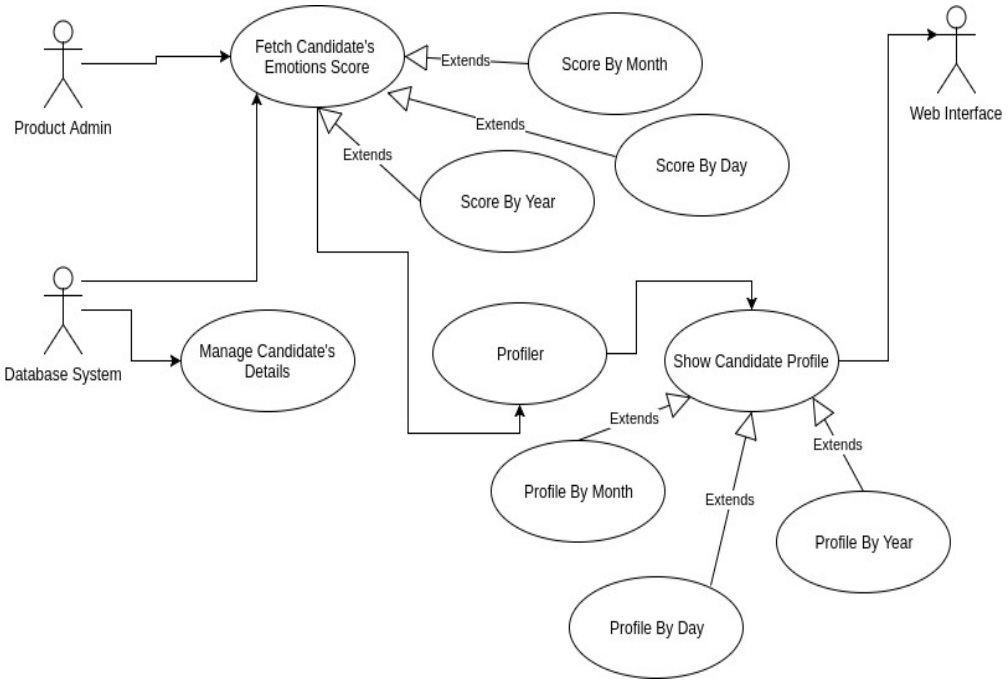

Figure 7.3: Use Case : Candidate Comparison

Candidate Profiling

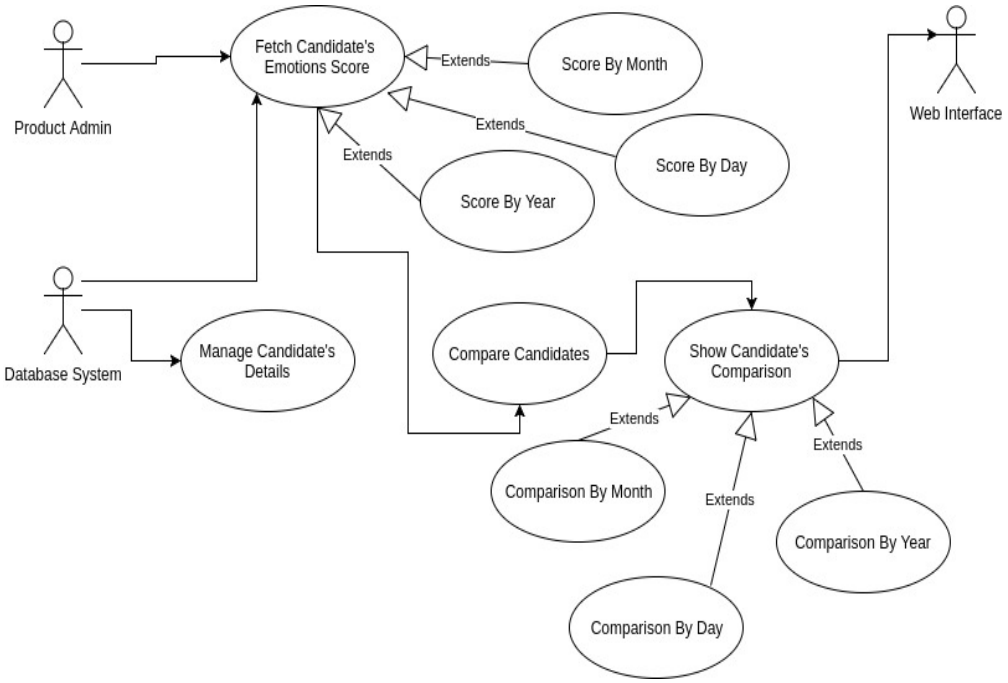

Figure 7.4: Use Case : Candidate Profiling

System Use Case

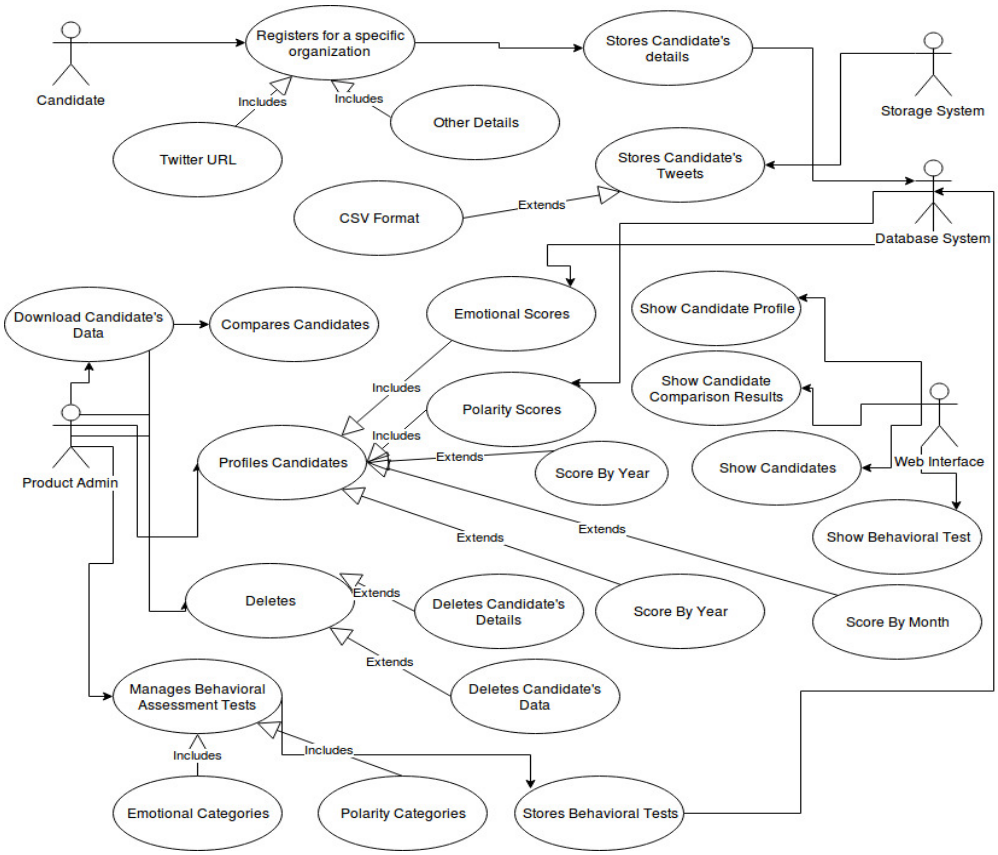

Figure 7.5: Use Case : Overall System

BEHAVIORAL MODEL AND DESCRIPTION

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

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.

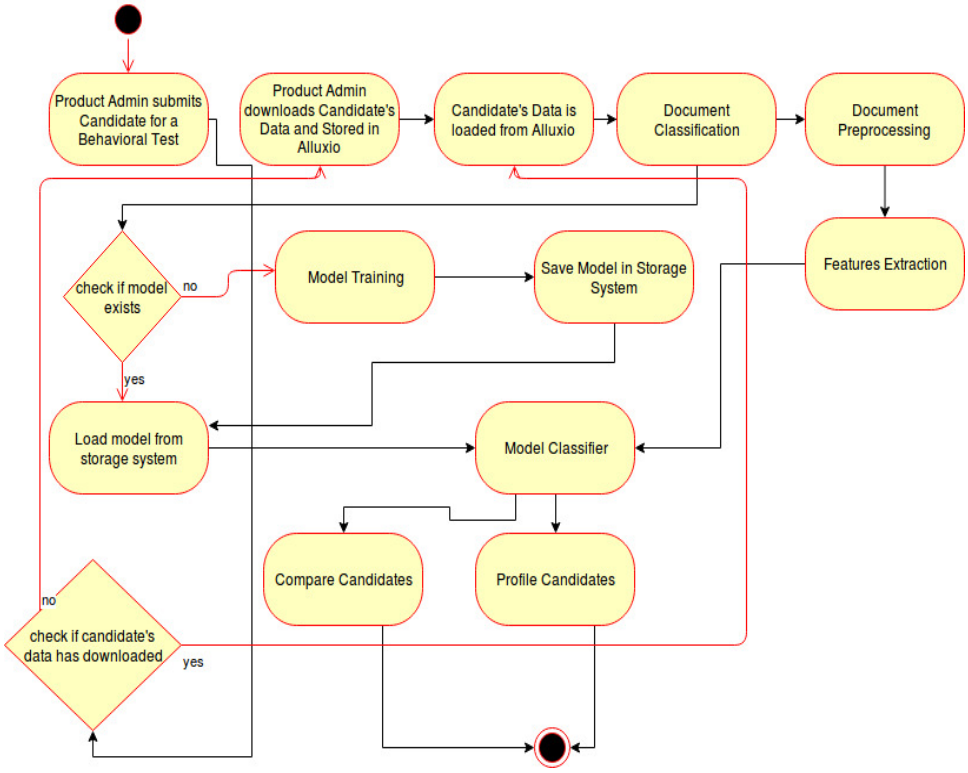

Figure 7.6: Activity Diagram

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.

Data Flow Diagram

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.

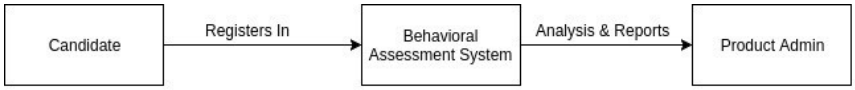

Figure 7.7: Level 0 DFD

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.

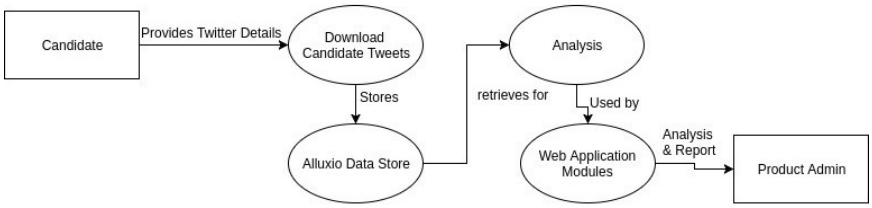

Figure 7.8: Level 1 DFD

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.

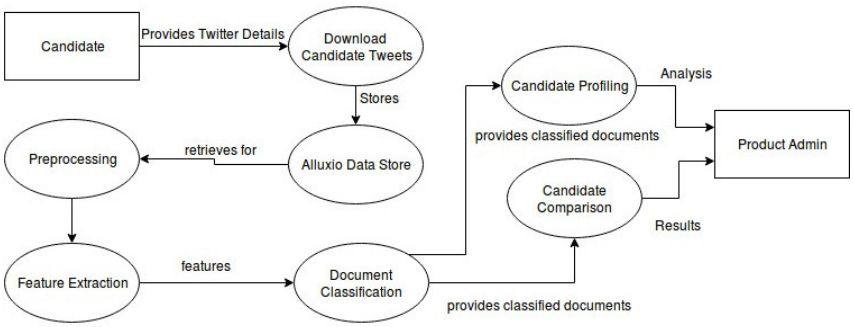

Figure 7.9: Level 2 DFD

NON-FUNCTIONAL REQUIREMENTS

Availability

Internet connection is needed as Bootstrap, J Query libraries are fetched from their respective Content Delivery Network. It is also required to fetch candidate's tweets from Twitter.

Scalability

The system should be scalable to connect more number of Spark nodes if candidate's data is increased. Apache Spark is configured to connect more nodes if need arises.

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. In web application, flash success and error messages generate delay of 1 second. Execution request sent from web application to spark application should involve minimal delay. Execution time of Spark application depends upon processing of candidate's tweets stored in CSV format. It should take less time as multiple nodes are connected for processing.

Security

A product administrator should be able to securely login to web application and submit candidate's tweets for analysis. Hadoop, Alluxio and Spark components must interact with each other in a secure manner.

Usability

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

Reliability

The system should efficiently analyze candidate's tweets entirely and stores all the classified documents to database. It should be reliable to perform web application requests, receive responses and perform actions based on responses without fail. Kerberos Authentication is implemented for mutual authentication between processing components of computational framework and storage systems to secure candidate's information.

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

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.

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.

ARCHITECTURAL DESIGN

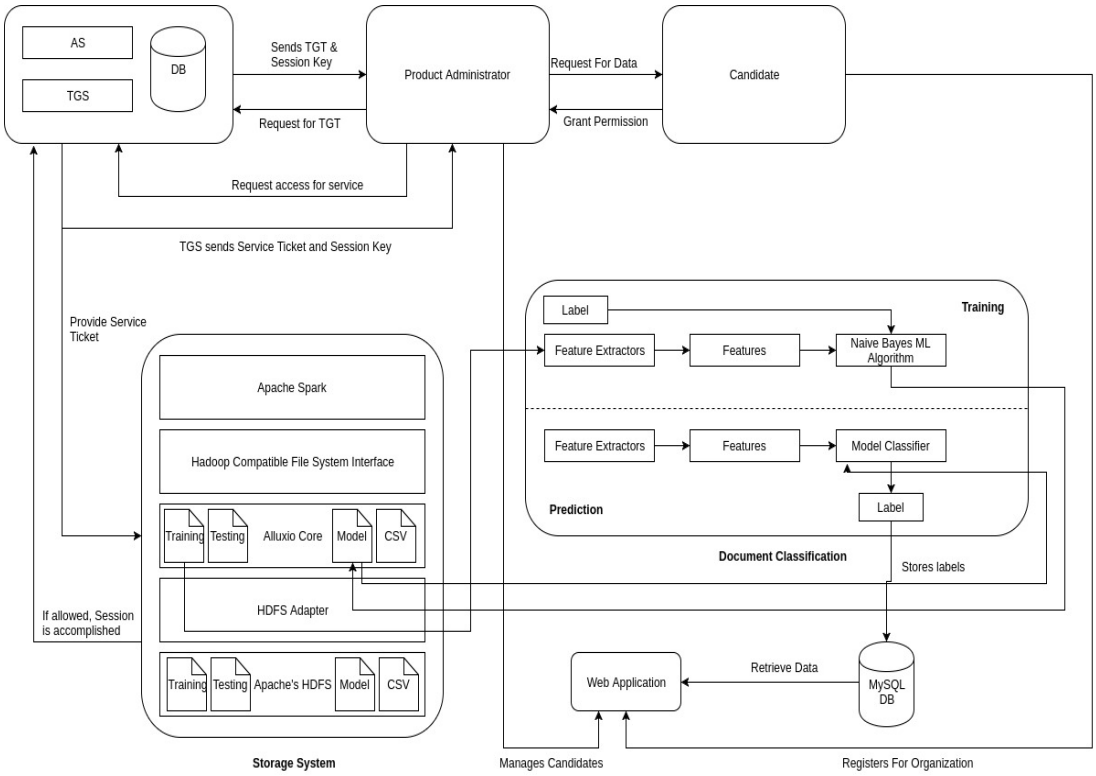

Figure 8.1: Proposed System Architecture

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.

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.

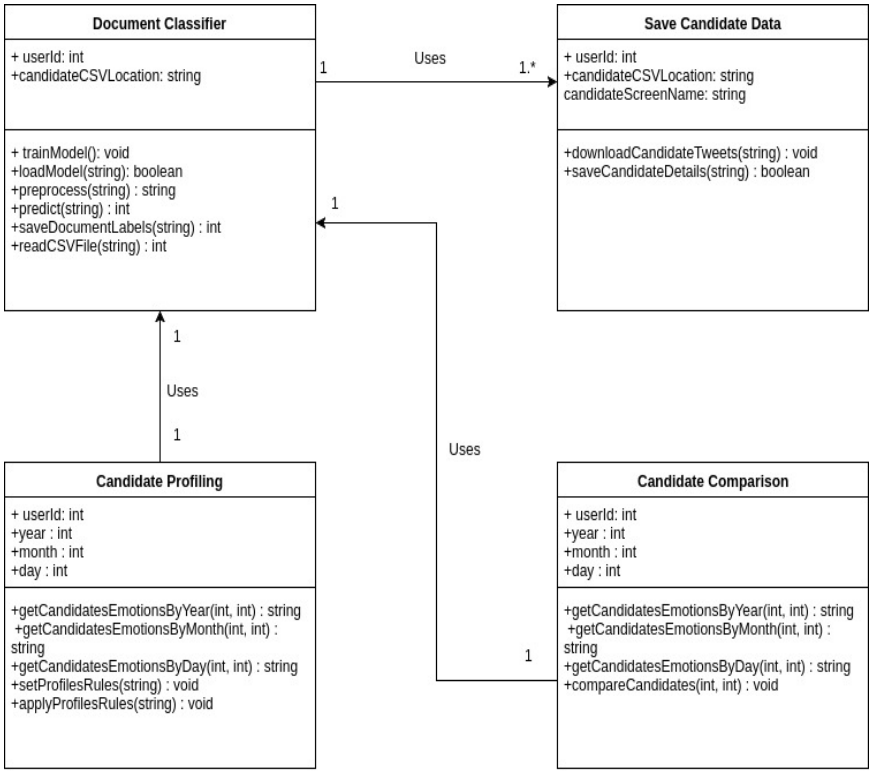

Figure 8.2: Class Diagram

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.

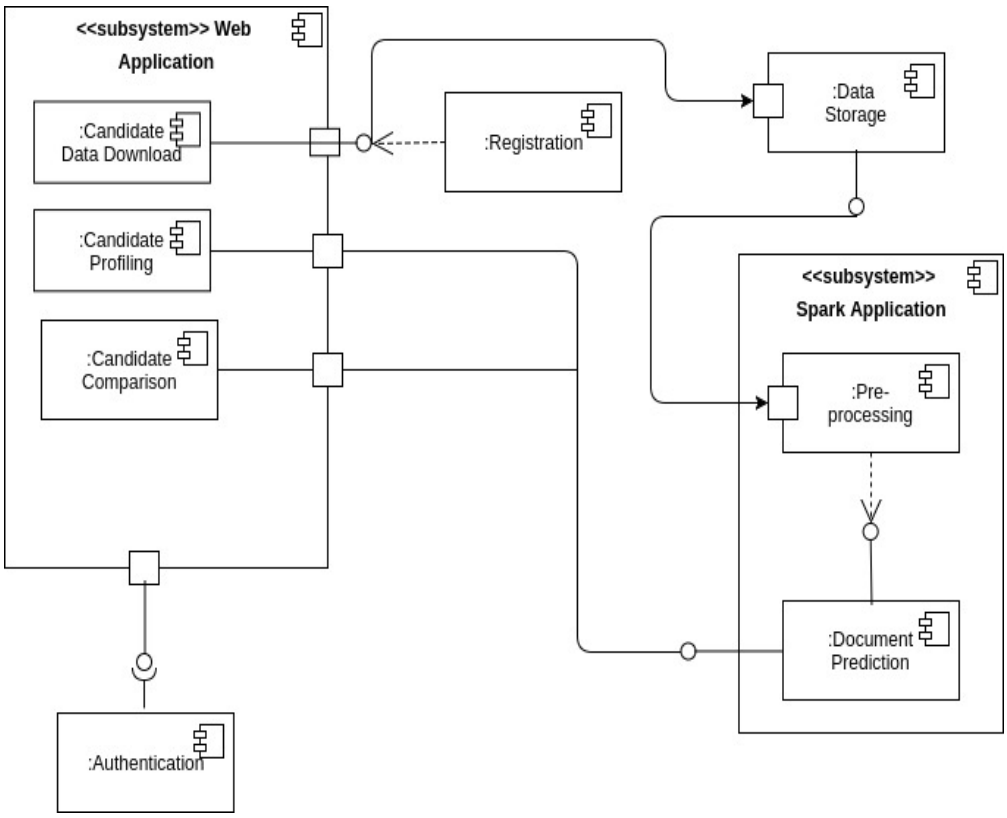

Figure 8.3: Component Diagram

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.

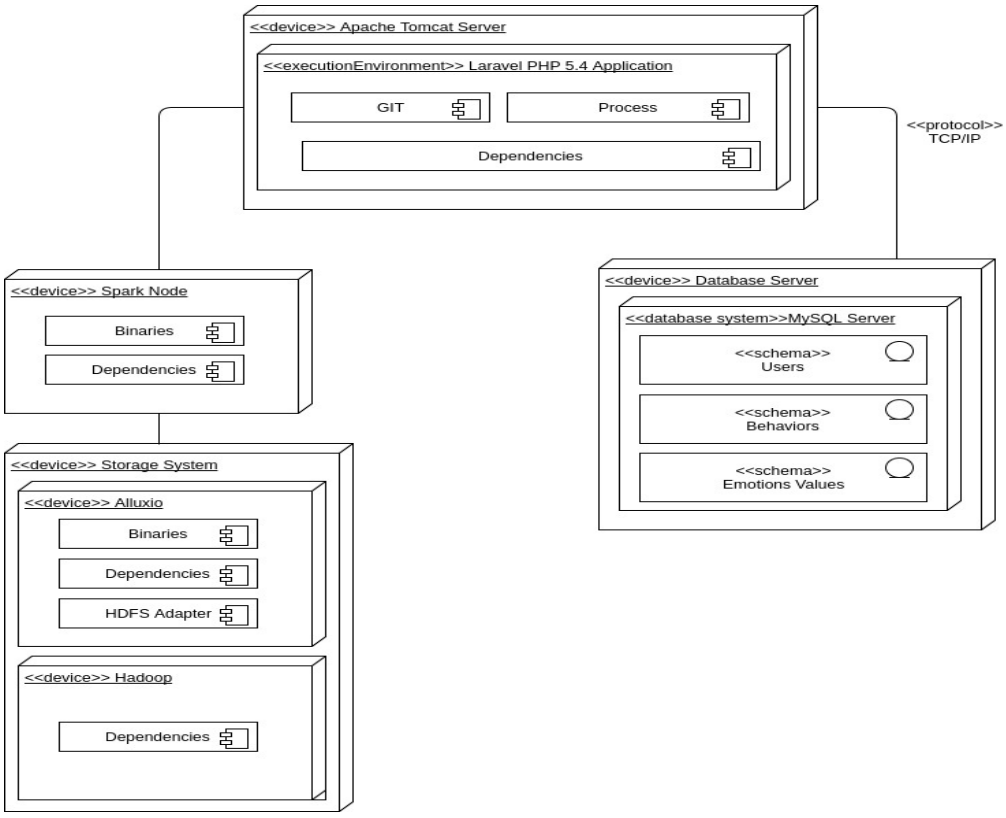

Figure 8.4: Deployment Diagram

CHAPTER 9

IMPLEMENTATION DETAILS

INTRODUCTION

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

ALGORITHM

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

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

MODULES

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.

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.

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.

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**

SNAPSHOTS

CHAPTER 10

TEST SPECIFICATION

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.

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.

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.

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.

TEST PLAN

Modules to be Tested

- Candidate's tweets fetched from Twitter module.

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

Testing Strategy

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.

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.

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.

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.

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

High Order Testing

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

Test Procedure

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

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

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

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

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

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

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] A.Ghotkar and K.Kharate,Hand Segmentation Techniques to Hand Gesture Recognition for Natural Human Computer Interaction, in International Journal of Human Computer Interaction (IJHCI), Volume(3), Issue(1), 2012: 15-250.

[2] A.Ghotkar, R.Khatal,S.Khupase,S.Asati and M.Hadap, Hand Gesture Recognition for Indian Sign Language, International Conference on Computer Communication and Informatics (ICCCI-2012), Coimbatore, India : 2012.

[3] Q.Wang, J.Cheng and J.Pang,A Novel Projector-Camera Interaction System with the Fingertip, Journal of Image and Graphics Vol.1,No.2,2013:80-84.

[4] S. Akhalaq, H. Shah, A. Ahmed, I. Mohmood and K. Khurshid,Hand Gesture based User Interface for Computer using a Camera and Projector, IEEE International Conference on Signal and Image Processing Applications (ICSIPA2011), 2011: 168-173.

[5] C.Chen, M.Zang, K.Qiu and Z. Pan,Real-Time Robust Hand Tracking based on Camshift and Motion Velocity, IEEE International conference on Digital Home, 2014: 20-24.

[6] L.Baraldi, F.Paci, G.Sera and L.Benini, Gesture Recognition in Ego-centric Videos using Dense Trajectories and Hand Segmentation, IEEE Conference on Computer Vision and Pattern Recognition Workshop, 2014 : 702-707.

[7] P.Chien,Y.Miao and J.Guo,A 3D Hand Tracking and Gesture Control in Complex Environments, IEEE Conference, 2015.

[8] A.Dhote and S. Badwaik,Hand Tracking and Gesture Recognition, IEEE International Conference on Pervasive Computing (ICPC): 2015.

[9] P.Nanivadekar and V.Kulkarni,Indian Sign Language Recognition: Database Creation, Hand Tracking and Segmentation, International Conference on Circuits, Systems, Communication and Information Technology Appliances (CSCITA), 2014: 358-363.

[10] J.Y.Lee and S.I. Yoo,An Elliptical Boundry Model for Skin Color Detection, in Proc.The International Conference on Imaging Science, Systems and Technology 2002.

[11] A.Salhi and A.Y. Jammoussi,Object tracking system using Camshift, Meanshift and Kalman filter, International Journal of Electrical, Computer, Energetic, Electronic and Communication Engineering Vol:6, No:4, 2012.

[12] D.Leonardo, M. Lizarazo, J. Antonio and T. Borja,Hand Position Tracking Using a Depth Image from a RGB-d Camera, IEEE International Conference,2015: 1680-1687.

ANNEXURE A

PAPERS PUBLISHED

PAPER TITLE

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

IJARCCE Certification

Figure A.1: IJARCCE Certificate

PAPER TITLE

Behavioral Assessment of Internal and External Candidates of Multiple Organizations using Sentiment Analysis and Emotion Mining

cPGCON Certificate

cPGCON Review

ANNEXURE B

DISSERTATION PLANNER

Table B.1: Dissertation Task Set

Task Title	**Dissertation Task**
T1	Study of Domain - Big Data Analytics
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 Single Node Hadoop Cluster
T8	Study of Distributed Computing and Configuring Alluxio
T9	Kerberos Authentication Protocol Configuration for Hadoop and Alluxio.
T10	Implementation Of Rubika Method of Authentication.
T11	Preparing Web Interface For Proposed System
T12	Report Preparation
T13	Dissertation Project Stage I Presentation
T14	Document Preprocessing
T15	Naive Bayes Classifier Implementation
T16	Candidate Behavioral Assessment Interface Implementation
T17	Candidate Assessment Flow Implementation
T18	Behavioral Model Construction For Candidates
T19	Behavioral Analytics Implementation and Report Generation
T20	Apache Spark and Apache Spark on Alluxio Benchmarking for Naive Bayes Classifier
T21	CPGCon Paper Presentation
T22	Predictive Model Construction
T23	Model Testing
T24	Experimental results, Analysis and Validation of results
T25	Project Review with Demonstration
T26	Report Validation and Submission, Report Submission