OPINION MINING FROM BIG

DATA

Submitted by

ARYAN SINGH (13000113020)

ARITRA GHOSAL (13000113017)

ASHISH KUMAR (13000113022)

CHITRAJIT DEY (13000113027)

Under the guidance of Mr. TAPAN CHOWDHURY

Assistant Professor, Dept. of CSE

Submitted for the partial fulfillment for the degree of Bachelor of Technology in Computer Science and Engineering



Techno India

EM 4/1, Salt Lake, Sector – V, Kolkata – 700 091.

**CERTIFICATE**

This is to certify that the project report entitled “**Opinion Mining from big data**” preparedby **Aryan Singh (130000113020), Aritra Ghosal (13000113017), Ashish Kumar (13000113022) and Chitrajit Dey(13000113027)** ofB.Tech (Computer Science &Engg.), Final Year, has been done according to the regulations of the Degree of Bachelor of Technology in Computer Science & Engineering. The candidates have fulfilled the requirements for the submission of the project report.

It is to be understood that, the undersigned does not necessarily endorse any statement made, opinion expressed or conclusion drawn thereof, but approves the report only for the purpose for which it has been submitted.

|  |  |
| --- | --- |
| ……………………………………….. | … ……………………………………… |
| Mr. Tapan Chowdhury | Ms. Poulami Dutta |
| Assistant Professor, Dept. of CSE | In-Charge and Assistant Professor, Dept. |
|  | of CSE |

………………………………………

(Signature of the External Examiner

with Designation and Institute)

--------------------------------------------------------------------------------------------

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

--------------------------------------------------------------------------------------------

**ACKNOWLEDGEMENT**

We would like to express our sincere gratitude to **Mr. Tapan Chowdhury** of the department of Computer Science and Engineering, whose role as project guide was invaluable for the project. We are extremely thankful for the keen interest he took in advising us, for the books and reference materials provided for the moral support extended to us.

Last but not the least we convey our gratitude to all the teachers for providing us the technical skill that will always remain as our asset and to all non-teaching staff for the gracious hospitality they offered us.

Place: Techno India, Salt Lake

Date:25/05/2017

ARYAN SINGH (13000113020)

ARITRA GHOSAL (13000113017)

ASHISH KUMAR (13000113022)

CHITRAJIT DEY (13000113027)

Department of Computer Science and Engineering

Techno India, SaltLake

Kolkata – 700 091

West Bengal, India.

Contents

[1. INTRODUCTION](#page6) [1](#page6)

[1.1Briefing](#page6) [1](#page6)

[1.2 Problem Domain](#page6) [1](#page6)

[1.3 Related Studies](#page6) [1](#page6)

[1.4 Glossary](#page7) [2](#page7)

[2. PROBLEM DEFINITION](#page9) [4](#page9)

[2.1Scope](#page9) [4](#page9)

[2.2 Exclusions](#page9) [4](#page9)

[2.3 Assumptions](#page9) [4](#page9)

[2.3.1Hardware Failure](#page9) [4](#page9)

[2.3.2 Large Data Set](#page9) [4](#page9)

[2.3.3 Simple Coherency Model](#page9) [4](#page9)

[3. PROJECT PLANNING](#page10) [5](#page10)

[3.1 Software Life Cycle](#page10) [5](#page10)

[3.2 Scheduling](#page11) [6](#page11)

[3.3 Cost Analysis](#page19) [14](#page19)

[4. REQUIREMENT ANALYSIS](#page20) [15](#page20)

[4.1 Requirement Matrix](#page20) [15](#page20)

[4.2 Requirement Elaboration](#page21) [16](#page21)

[4.2.1 Big Data Fetching (Requirement ID: BDF-1)](#page21) [16](#page21)

[4.2.2 Data Reduction (Requirement ID: DRD-1)](#page21) [16](#page21)

[4.2.3 Dictionary Formation (Requirement ID: DF)](#page22) [17](#page22)

[4.2.4 Data Mining (Requirement ID: DM)](#page22) [17](#page22)

[4.2.5 Opinion Mining using game theory(Requirement ID: OM)](#page22) [17](#page22)

[4.2.6 Generation of Result Based on String Query (Requirement ID: GR)](#page22) [17](#page22)

[4.2.7 Visualization of Result (Requirement ID: VR-1)](#page22) [17](#page22)

[5. DESIGN](#page23) [18](#page23)

[5.1 Technical Environment](#page23) [18](#page23)

[5.2 Hierarchy of Modules](#page23) [18](#page23)

[5.3 Detailed Design](#page23) [18](#page23)

[5.3.1 Mapper Function](#page23) [18](#page23)

[5.3.2 Reducer Function](#page24) [19](#page24)

[5.4 Test Plan](#page25) [20](#page25)

[6. IMPLEMENTATION](#page26) [21](#page26)

[6.1 Implementation Details](#page26) [21](#page26)

[6.1.1 Big data fetching](#page26) [21](#page26)

[6.1.2 Data Reduction](#page26) [21](#page26)

[6.1.3 Dictionary formation](#page26) [21](#page26)

[6.1.4 Opinion Mining](#page26) [21](#page26)

[6.1.5 Game Theory Implementation](#page26) [21](#page26)

[6.2 System installation manual](#page27) [22](#page27)

[6.3 System Uses Instruction](#page27) [22](#page27)

[7. CONCLUSION](#page28) [23](#page28)

[7.1Project Benefits](#page28) [23](#page28)

[7.2Future Scope](#page29) [24](#page29)

[8. REFERENCES](#page29) [24](#page29)

[APPENDIX A](#page30) [25](#page30)

[APPENDIX B](#page32) [27](#page32)

**List of figures:**

[**Figure 1: Iterative Waterfall Model**](#page10)[**5**](#page10)

[**Figure 2:Gantt-Chart-1**](#page11)[**6**](#page11)

[**Figure 3:Gantt-Chart-2**](#page12)[**7**](#page12)

[**Figure 4:Gantt-Chart-3**](#page13)[**8**](#page13)

[**Figure 5:Gantt-Chart-4**](#page14)[**9**](#page14)

[**Figure 6:Schedulling-1**](#page15)[**10**](#page15)

[**Figure 7:Schedulling-2**](#page16)[**11**](#page16)

[**Figure 8:Schedulling-3**](#page17)[**12**](#page17)

[**Figure 9:Schedulling-4**](#page18)[**13**](#page18)

[**Figure 10:Requirement Matrix**](#page20)[**15**](#page20)

[**Figure 11:Hierarchy of modules**](#page23)[**18**](#page23)

[**Figure 12:Class Diagram**](#page24)[**19**](#page24)

[**Figure 13:Use Case Diagram**](#page30)[**25**](#page30)

[**Figure 14:Activity Diagram**](#page31)[**26**](#page31)

[**Figure 15: Starting Daemons**](#page33)[**28**](#page33)

[**Figure 16:Running Daemons**](#page34)[**29**](#page34)

[**Figure 17:HDFS**](#page34)[**29**](#page34)

[**Figure 18:Map-reduce**](#page35)[**30**](#page35)

[**Figure 19:Opinion-Mining-1**](#page35)[**30**](#page35)

[**Figure 20:Opinion-Mining-2**](#page36)[**31**](#page36)

**List of Tables**:

[**Table 1: Problem domain**](#page6)[**1**](#page6)

[**Table 2:Technical Environment**](#page23)[**18**](#page23)

[**Table 3:Test Plan**](#page25)[**20**](#page25)

1. INTRODUCTION

1.1Briefing

Big data is large amount of data, which requires new technologies and architectures so that it becomes possible to extract value from it by capturing and analysis process. Opinion mining, also sometimes technically referred as sentimental analysis, is the field of study that analyses people’s opinions, sentiments, evaluations, appraisals, attitudes, and emotions towards entities such as products, services, organizations, individuals, issues, events, topics, and their attributes. Our Project will perform opinion mining on unstructured and raw data (big data) and present it in the form of statistical charts so that one can easily understand whatever meaningful information the data want to convey.

1.2 Problem Domain

*Table 1: Problem domain*

|  |  |
| --- | --- |
| **Hardware / Software** | **Specifications** |
| **Hard disk** | **100 GB** |
| **RAM** | **4GB** |
| **Processor** | **Intel Core i3(Dual Core)** |
| **Operating System** | **Linux Ubuntu 14.04, Windows 10** |
| **Software Requirements** | **Hadoop-2.7.3, Python 2,Hive-2.1.1,** |
|  | **Apache-Flume-1.6,Ms-Excel 2016** |
|  |  |

1.3 Related Studies

Sentiment analysis has become one of the mainstream researches in social network analysis. Its impact can be seen in many practical applications, ranging from public opinion analysis to marketing of public praise and information prediction.

With the development of CMC(Computer Meditated Communication), a large number of people are writing reviews online. However, getting an overall sense of the reviews can daunt and time consuming. To solve these problems, recent years have witnessed a growing interest in affective computing, whose objective is to find opinions, feelings, and attitude expressed in text, rather than facts. In the literature, affective computing also goes under various names, such as opinion mining [2,3], sentiment extraction [4-6], etc. Its related work may come from both computer science and linguistics, and its immediate applications may involve data mining, market intelligence, and customer relationship management. Despite the increased focus on affective computing of static text, there has been limited emotional evolution analysis of complex interactive text. In the real world, the attitude toward a product or person may evolve over time. For example, a product may improve its features and functionalities with upgrades, and a person may change

1

his/her point of view during online discussion. Such changes may be reflected in newly available reviews. Accordingly, a sentiment analysis system must adjust itself to capture the changes effectively, and even provide some reasonable predictions.

First, an efficient affective computing framework is proposed to capture the underlying emotions of Chinese online reviews. It can accurately calculate the semantic orientation of the entire review, without requiring expensive manual labeling of seed words. As users' attitudes might influence with each other, predicting their future emotional behaviors that only relying on the emotional values of historical reviews is very one-sided. Therefore, we propose a game theory based emotional evolution prediction algorithm combining the affective computing, in which the mixed Nash-equilibrium strategies are calculated as the future emotional behavior of interactive users. Then, experimental results on the large-scaled review dataset are provided to demonstrate the effectiveness and accurateness of our approaches.

1.4 Glossary

1. **Opinion mining**: It can be defined as a sub-discipline of computational linguistics thatfocuses on extracting people’s opinion from the web. The recent expansion of the web encourages users to contribute and express themselves via blogs, videos, social networking sites, etc. All these platforms provide a huge amount of valuable information that we are interested to analyse.
2. **Big data:** Big data is a buzzword, or catch-phrase, used to describe a massive volume ofboth structured and unstructured data that is so large it is difficult to process using traditional database and software techniques.
3. **Sentiment Analysis:** Sentiment analysis, on the other hand, is about determining thesubjectivity, polarity (positive or negative) and polarity strength (weakly positive, mildly positive, strongly positive, etc.) of a piece of text.
4. **Apache Hadoop:** Apache Hadoop is an open-source software framework written in Javafor distributed storage and distributed processing of very large data sets on computer clusters built from commodity hardware.
5. **Hadoop:** Hadoop is an open-source software framework for storing data and runningapplications on clusters of data.
6. **Map-Reduce:** Map-Reduce is a programming model and an associated implementationfor processing and generating large data sets with a parallel, distributed algorithm on a cluster.
7. **Name-Node**: The Name-Node is the centrepiece of an HDFS file system. It keeps thedirectory tree of all files in the file system, and tracks where across the cluster the file data is kept.

2

1. **Data-Node:** A Data-Node stores data in the HDFS. A functional file system has morethan one Data-Node, with data replicated across them.
2. **Job-Tracker:** The Job-Tracker is the service within Hadoop that farms out Map-Reducetasks to specific nodes in the cluster, ideally the nodes that have the data, or at least are in the same rack.
3. **Task-Tracker:** A Task-Tracker is a node in the cluster that accepts tasks - Map, Reduceand Shuffle operations - from a Job-Tracker.
4. **Replication Factor:** Replication Factor tells about data replication on multiple nodes bythe way we can achieve high fault tolerant and high availability.
5. **HDFS:** The Hadoop Distributed File System (HDFS) is a sub-project of the ApacheHadoop project. This Apache Software Foundation project is designed to provide a fault-tolerant file system designed to run on commodity hardware.

3

2. PROBLEM DEFINITION

2.1Scope

Having a right product is important and equally important is to present it before the right customer (one who actually needs it or is interested in it). The product should put on positive feeling of ownership among the individuals. And such feelings are clearly expressed in opinion mining polls.[1] The critical part out here is to correlate as what the product offers and what are the actual wants of a consumer.

Many experts consider social media as a missed opportunity for better policy debate. At the same time, the sheer amount of raw data is also an opportunity to better make sense of opinions.[2]Argument mapping software helps organizing in a logical way these policy statements, by explicating the logical links between them.

2.2 Exclusions

For simple and structured data, databases already exist. Hence working upon unstructured data is primary focus of this project.

2.3 Assumptions

2.3.1Hardware Failure

Hardware failure is the norm rather than the exception. An HDFS instance may consist of hundreds or thousands of server machines, each storing part of the file system’s data. The fact that there are a large number of components and that each component has a non-trivial probability of failure means that some component of HDFS is always non-functional. Therefore, detection of faults and quick, automatic recovery from them is a core architectural goal of HDFS.

2.3.2 Large Data Set

Applications that run on HDFS have large data sets. A typical file in HDFS is gigabytes to terabytes in size. Thus, HDFS is tuned to support large files. It should provide high aggregate data bandwidth and scale to hundreds of nodes in a single cluster. It should support tens of millions of files in a single instance.

2.3.3 Simple Coherency Model

HDFS applications need a write-once-read-many access model for files. A file once created, written, and closed need not be changed. This assumption simplifies data coherency issues and enables high throughput data access. A Map-Reduce application or a web crawler application fits perfectly with this model. There is a plan to support appending-writes to files in the future.

4

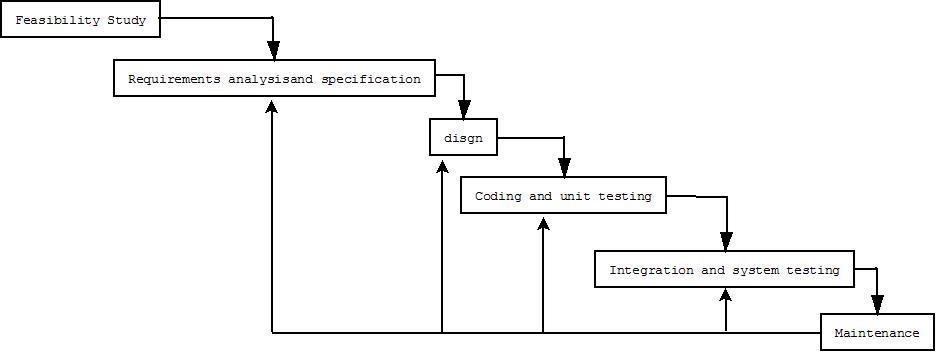
3. PROJECT PLANNING

3.1 Software Life Cycle

The Software life cycle model that we have used here is “Iterative Waterfall Model”. The different phases of this model are,

* Requirements Definition
* System and Software Design
* Implementation and Unit Testing
* Integration and System Testing
* Operation and Maintenance

The phases are represented diagrammatically below.



*Figure 1: Iterative Waterfall Model*

5

3.2 Scheduling



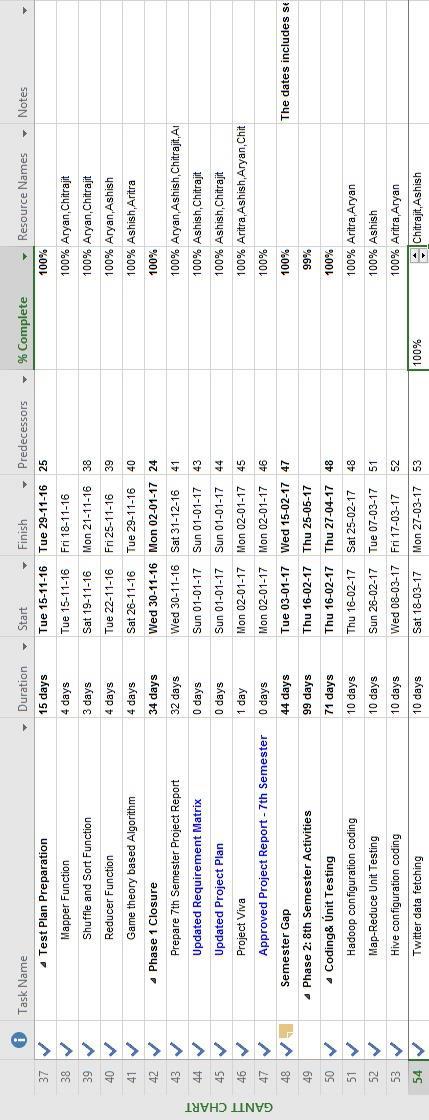
*Figure 2:Gantt-Chart-1*

6



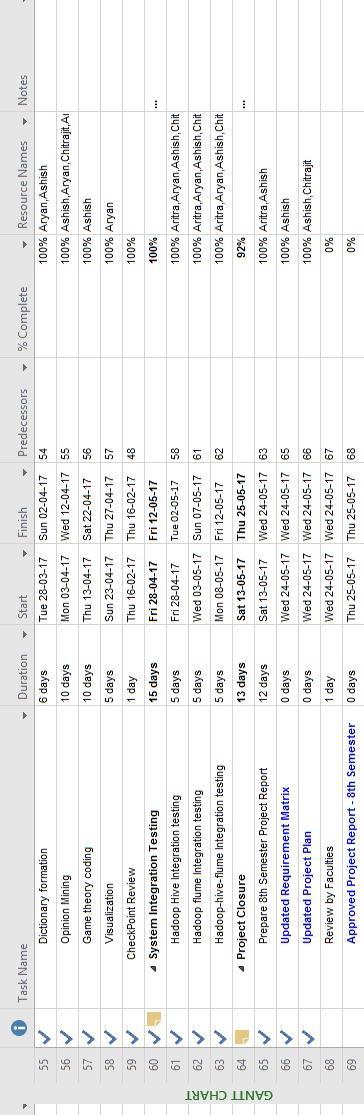
*Figure 3:Gantt-Chart-2*

7



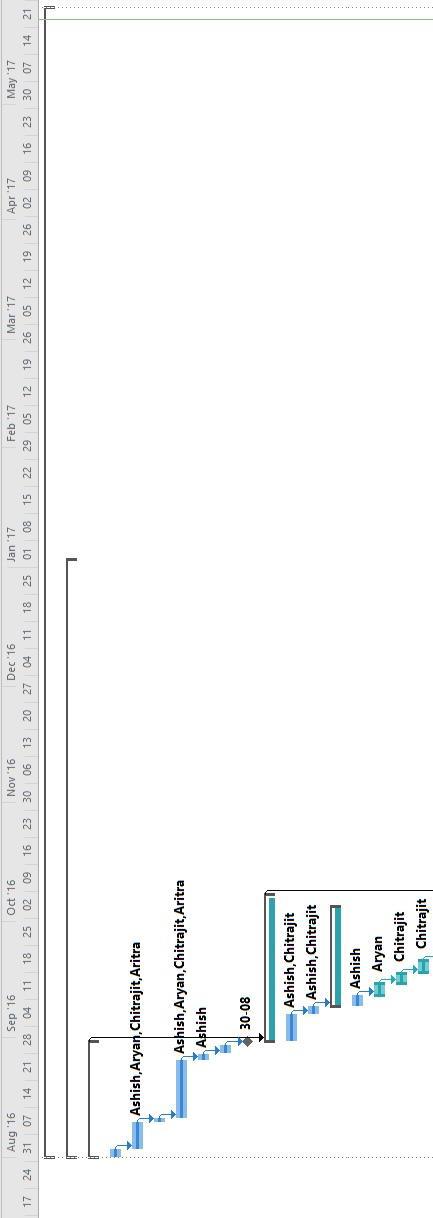
*Figure 4:Gantt-Chart-3*

8



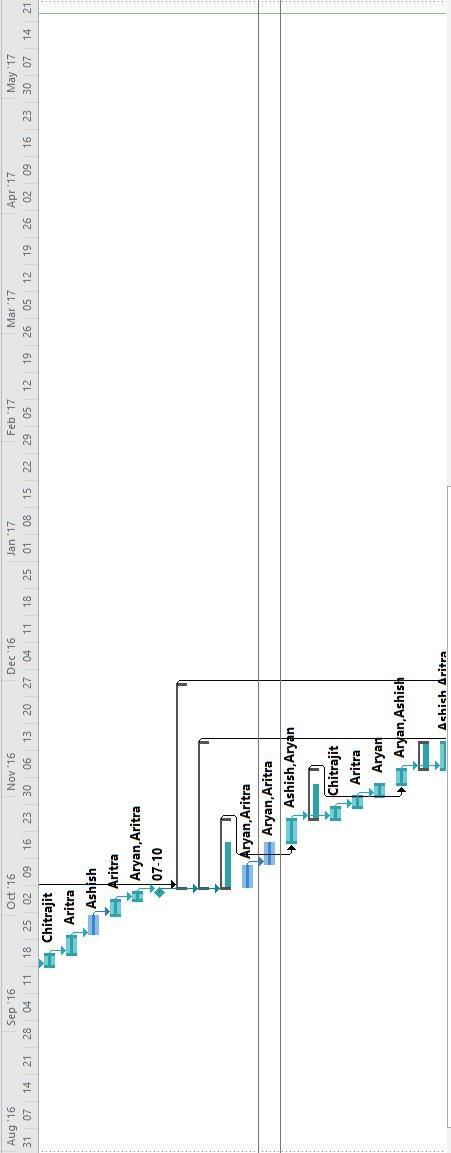
*Figure 5:Gantt-Chart-4*

9



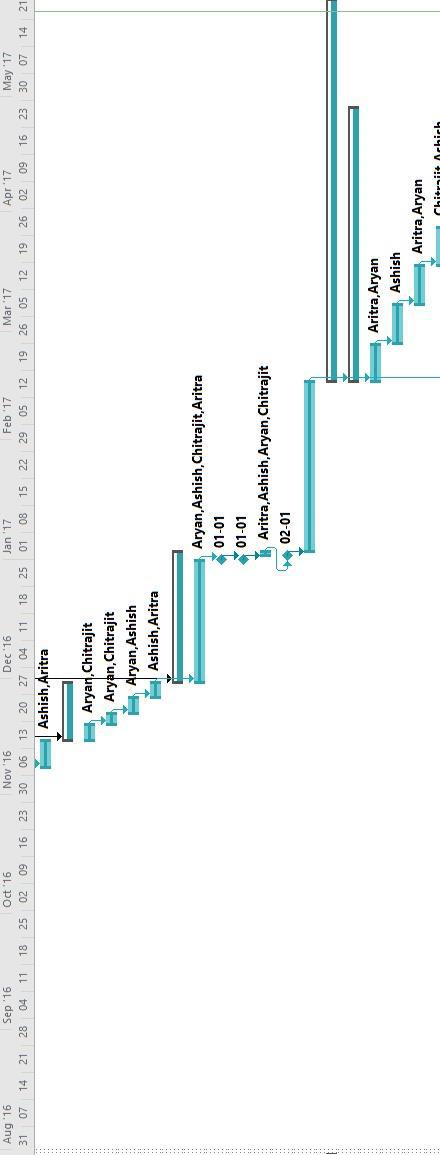
*Figure 6:Schedulling-1*

10



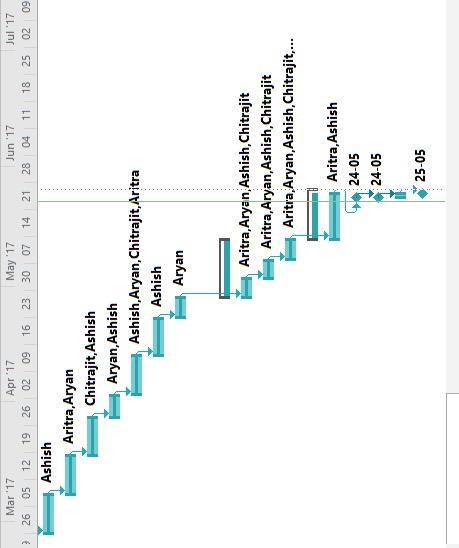
*Figure 7:Schedulling-2*

11



*Figure 8:Schedulling-3*

12



*Figure 9:Schedulling-4*

13

3.3 Cost Analysis

We have used the basic COCOMO model for cost estimation.

Estimated KLOC= 3

As our project is Embedded, a1= 3.6, a2=1.20, b1= 2.5 and b2= 0.32

Effort = 3.6 ∗ (3)1.20

=13.45 PM.

Tdev = 2.5 ∗ (10.81)^0.32 Months

= 5.74 Months

As our project is an academic project, so we are not considering the cost. If we commercialize our project, then software cost need to be added because we need to pay for the software. Time of development does not include semester break and team formation periods.

14

4. REQUIREMENT ANALYSIS

4.1 Requirement Matrix



*Figure 10:Requirement Matrix*

15

4.2 Requirement Elaboration

4.2.1 Big Data Fetching (Requirement ID: BDF-1)

4.2.1.1 Fetching big data from internet sources (Requirement ID: BDF-1.1)

Python script can be used to fetch unstructured big data (of the order of 50-100 Gb) from internet sources (web scrapping).

4.2.1.2 Data Cleaning (Requirement ID: BDF-1.2)

Data cleansing, data cleaning, or data scrubbing is the process of detecting and correcting (or removing) corrupt or inaccurate records from a record set, table, or database and refers to identifying incomplete, incorrect, inaccurate or irrelevant parts of the data and then replacing, modifying, or deleting the dirty or coarse data.

4.2.2 Data Reduction (Requirement ID: DRD-1)

4.2.2.1 Splitting of data into different commodity hardware (Requirement ID: DRD-1.1)

The large dataset is split into several smaller units and is distributed among various nodes for parallel and independent processing.

4.2.2.2 Mapping as Key Value Pairs (Requirement ID: DRD-1.2)

A Map is an object that maps keys to values. A map cannot contain duplicate keys: Each key can map to at most one value. It models the mathematical function abstraction. Each element of a map is accessed(or, is indexed) by a unique key, and so they are known as **key-value pairs**.

4.2.2.3 Shuffling and Sorting (Requirement ID: DRD-1.3)

After the first map tasks have completed, the nodes may still be performing several more map tasks each. But they also begin exchanging the intermediate outputs from the map tasks to where they are required by the reducers. This process of moving map outputs to the reducers is known as **shuffling**.

Each reduce task is responsible for reducing the values associated with several intermediate keys. Hadoop automatically sorts the set of intermediate keys on a single node before they are presented to the Reducer.

4.2.2.4 Generate Output Using Reducer Function (Requirement ID: DRD-1.4)

A Reducer instance is created for each reduce task. This receives a key as well as an iterator over all the values associated with the key. The iterator returns the values associated with a key in an undefined order.

16

4.2.3 Dictionary Formation (Requirement ID: DF)

Systems, which process natural language, require a reliable source of information about words. Not only must their lexical subsystems handle large known words; they must also cope with coinages. These subsystems combine very idiosyncratic lexical information, stored in a dictionary, with systematic information derived from word structure.

4.2.4 Data Mining (Requirement ID: DM)

Data mining (sometime called data or knowledge discovery) is the process of analyzing data from different perspectives and summarizing it into useful information - information that can be used to increase revenue, cuts costs, or both. Data mining software is one of a number of analytical tools for analyzing data. It allows users to analyze data from many different dimensions or angles, categorize it, and summarize the relationships identified. Technically, data mining is the process of finding correlations or patterns among dozens of fields in large relational databases.

4.2.5 Opinion Mining using game theory(Requirement ID: OM)

Opinion Mining can be defined as a sub-discipline of computational linguistics that focuses on extracting people’s opinion from the web. The recent expansion of the web encourages users to contribute and express themselves via blogs, videos, social networking sites, etc.

4.2.6 Generation of Result Based on String Query (Requirement ID: GR)

Result will be generated based on string query.

4.2.7 Visualization of Result (Requirement ID: VR-1)

4.2.7.1 Text format result visualization (Requirement ID: VR-1.1)

The result will be visualized as text format for every comments whether it is positive, negative or neutral comment.

4.2.7.2 Statistical format result visualization (Requirement ID: R-1.2)

The result can be visualized as statistical format in forms of charts or graphs.

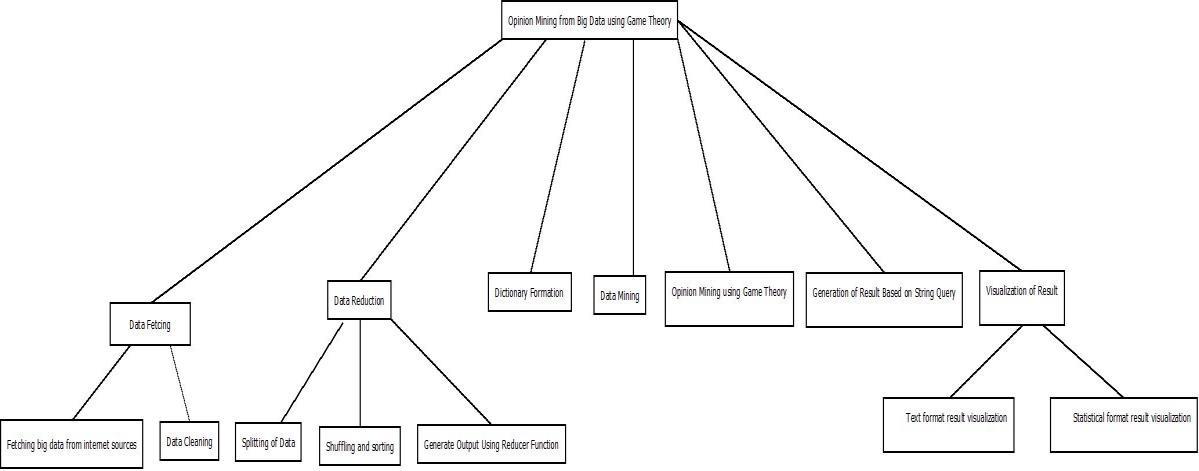
17

5. DESIGN

5.1 Technical Environment

*Table 2:Technical Environment*

|  |  |  |
| --- | --- | --- |
|  | **Hardware / Software** | **Specifications** |
|  | **Hard disk** | **100 GB** |
|  | **RAM** | **4GB** |
|  | **Processor** | **Intel Core i3(Dual Core)** |
|  | **Operating System** | **Linux Ubuntu, Windows 10** |
|  | **Software Requirements** | **Hadoop-2.7.3,Python2,Apache-Flume-** |
|  |  | **1.6,Apache-Hive-2.2.1,Ms-Excel 2016** |
|  |  |  |
| 5.2 Hierarchy of Modules | |  |



*Figure 11:Hierarchy of modules*

5.3 Detailed Design

5.3.1 Mapper Function

Algorithm 1:

Map( String key, String value)

18

For each word w in value

EmitIntermediate(w,”1”)

5.3.2 Reducer Function

Algorithm 2:

Reducer( String Key, Iterator value)

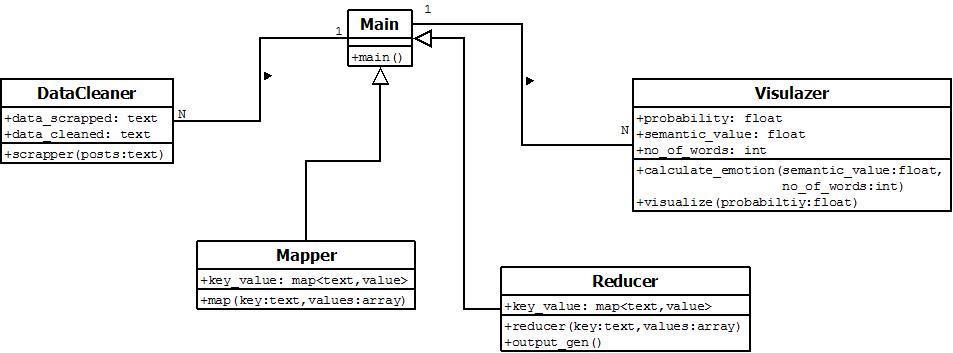
For each v in values

Do

Result+=parseint(v)

Emit(AsString(result))

The class diagram is show below:



*Figure 12:Class Diagram*

TheUse Case Diagram and Sequence Diagram is given in APPENDIX A

19

5.4 Test Plan

*Table 3:Test Plan*

|  |  |  |  |  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| **Serial** |  |  | **Testing Component details** | | | | |  |  |  |  |  |  |
| **No.** | **Test ID** | **Description** | |  |  | **Test Input** | | **Desired Result** | | | | |  |
| 1 | T\_BDF-1.1 | Fetching | Big | Data | from | Power | / | Partial or no data is | | | | | |
|  |  | internet sources | |  |  | Network Failure | | fetched | |  |  |  |  |
|  |  |  |  |  |  | Up & Running | | All data is fetched | | | | | |
|  |  |  |  |  |  |  |  | successfully | | | |  |  |
| 2 | T\_BDF-1.2 | Data Cleaning | |  |  | Unstructured | | Structured Data | | | | |  |
|  |  |  |  |  |  | Data |  |  |  |  |  |  |  |
|  |  |  |  | | |  |  |  | | | | | |
| 3 | T\_DRD-1.1 | Splitting | of data between | | | Link | Failure | Part of data sent on | | | | | |
|  |  | different |  | commodity | | between |  | the | other | | side | | of |
|  |  | hardware |  |  |  | different |  | broken | | link | | is | not |
|  |  |  |  |  |  | computers | | sent. |  |  |  |  |  |
|  |  |  |  |  |  | (nodes). |  |  |  |  |  |  |  |
|  |  |  |  |  |  | Up & Running | | All | split | | data | | are |
|  |  |  |  |  |  |  |  | sent | to | | various | | |
|  |  |  |  |  |  |  |  | nodes as desired | | | | |  |
| 4 | T\_DRD-1.2 | Mapping as key Value Pairs | | | | A key | value | A new | | key | | value | |
|  |  |  |  |  |  | pair is provided. | | pair is generated. | | | | | |
| 5 | T\_DRD-1.3 | Shuffling and Sorting | | |  | Key Value pair | | Same | | key | | value | |
|  |  |  |  |  |  | from | mapper | pair | is | routed | | | to |
|  |  |  |  |  |  | phase |  | same reducer | | | |  |  |
| 6 | T\_DRD-1.4 | Generate | Output | | using | Group | of key | Aggregated Output | | | | | |
|  |  | reducer function | |  |  | value pairs | |  |  |  |  |  |  |
| 7 | T\_DF | Dictionary Formation | | |  | Tokens | of | Sentimental | | | |  |  |
|  |  |  |  |  |  | sentences | | polarity of tokens | | | | | |
| 8 | T\_DM | Data Mining | |  |  | Data Set |  | Some | |  |  | Visual | |
|  |  |  |  |  |  |  |  | Statistics to analyze | | | | | |
|  |  |  |  |  |  |  |  | data |  |  |  |  |  |
| 9 | T\_OM | Opinion | Mining | | Using | Full sentences | | Total | | Sentimental | | | |
|  |  | Game Theory | |  |  |  |  | polarity | |  |  |  |  |
| 10 | T\_GR | Generalization | | of | String | Full Tweet | | Sentiment value of | | | | | |
|  |  | Based on Query String | | | |  |  | the total tweet | | | | |  |
| 11 | T\_VR-1.1 | Text | Format | | Result | Sentiment | | Visualization | | | |  | in |
|  |  | Visualization | |  |  | Value |  | textual format | | | | |  |
| 12 | T\_VR-1.2 | Statistical Format | | | Result | Sentiment | | Visualization | | | |  | in |
|  |  | Visualization | |  |  | Value |  | form |  | of |  | charts | |
|  |  |  |  |  |  |  |  | generally generated | | | | | |
|  |  |  |  |  |  |  |  | custom | |  |  | using | |
|  |  |  |  |  |  |  |  | required software | | | | | |

20

6. IMPLEMENTATION

6.1 Implementation Details

6.1.1 Big data fetching

We are fetching data by using Apache flume. We have created a twitter application named sentiment in apps.twitter.com. It gives me four different keys/ passwords. We are configuring flume with these keys to connect with the twitter. This requires a stable internet connection. The configuration file codes can be found in **APPENDIX B**.

6.1.2 Data Reduction

We are getting data from twitter in JSON form, which is unstructured. In order to format data so that it can be used for sentimental analysis, we are creating a table in hive which will store all details of a tweets. There is a jar file called serede which will parse JSON format data into a structured data. This way we are getting reduced data. Hive will perform map-reduce operations for that. Hive is configured in such a way so that it can work with Hadoop and HDFS. Table creation and other codes can be found out in

**APPENDIX B**.

6.1.3 Dictionary formation

We are using an open source dictionary. This dictionary was initially in a tsv file i.e. tab separated file. For our purpose, we require our dictionary to be in Hive. So we are creating a schema for dictionary. We inserted the data from tab separated file to the schema formed in the Hive. The table creation code can be found in **APPENDIX B.**

6.1.4 Opinion Mining

We are splitting the tweets inserted in the hive tables in sentences. i.e. if a tweet contains multiple sentences, we are separating each sentence. Then, each sentence is divided into words. Each words is then assigned a value from {-1,0,1} depending upon if word is positive or negative or neutral. For this purpose, we are using dictionary made in the previous step. After each word is assigned a sentiment value, we are aggregating back the whole tweet to get the sentiment of the whole tweet and storing the result in the new table in hive. This will be done by some map-reduce operations using have query. This way we are getting our final result. Sample example can be found in **APPENDIX B.**

6.1.5 Game Theory Implementation

We have done the same opinion mining using text-blob library of python. We are comparing the performance of both the methods using game theory principles. We are using concepts of nash equilibrium for that. A 2 player game is designed where our both

21

approaches are the two players. The result of game is the probability of performances of both the approaches. The method having more probability is considered as good. The Game is explained in **APPENDIX B.**

6.2 System installation manual

Step-1 Installing Java

Java is the primary requirement for running hadoop on any system. Command for installation is:

Java version “1.7.0\_101”

Java (TM) SE Runtime Environment ( build 1.7.0\_101)

Java HotSpot (TM) 64-bit Server VM ( build 1.7.0\_101)

Step-2 Downloading Hadoop 2.7.3

Commands for downloading the hadoop are:

wget <http://apache.claz.org/hadoop/common/hadoop-2.7.3/hadoop-2.7.3.tar.gz>

tar xzf hadoop-2.7.3.tar.gz

mv hadoop-2.7.3 hadoop

Step-3 Configure Hadoop

We need to set environmental variables uses by hadoop. The following files need to be updated:

Core-site.xml

Hdfs-site.xml Mapred-site.xml Yarn-site.xml

Additionally, we also require to update .bashrc file for setting environmental variables for java path and Hadoop path.

Step-4

We will format the namenode by using following command:

Hdfsnamenode -format

6.3 System Uses Instruction

We need to start the Hadoop cluster. It can be done by using following command:

./start-dfs.sh

./start-yarn.sh

22

Give password whenever it is asked.

We can access Apache Hadoop Services in browser by typing localhost:50070 in the browser.

For stopping Hadoop daemons, The following commands can be used:

./stop-dfs.sh

./stop-yarn.sh

Alternatively, we can also use the command: ./stop-all.sh

7. CONCLUSION

Opinion mining on unstructured big data using game theory has been a research interest for recent years. Although several notable works have come in this field, a fully automated and highly efficient system has not been introduced till now. This is because of the unstructured nature of natural language, Big Data. The vocabulary of natural language is very large that things become even hard. Various methods and hybrid approaches discussed above can be used for fully automated and efficient sentiment analysis on big data. By performing an extensive research in the related area, we identified many research challenges in sentiment analysis that are yet to be addressed. Several challenges still exist in the field of machine learning and some of them are co-reference Resolution, domain dependency etc. These problems have to be tackled separately and those solutions can be used to improve the methods to do effective sentiment analysis and opinion extraction from big data.

7.1Project Benefits

Our project will make decision making easier and efficient. We will extract some meaningful information from unstructured and raw data and present it in the form of statistical charts so that one can easily understand whatever the meaning we want to convey. We can easily visualize the data and will be able to take appropriate decisions so that business can be made more profitable.

The most interesting benefit of our project is to create new products and services for customers. Online companies have done this for a decade or so, but now predominantly offline firms are doing it too. Through our analysis of data, companies may know the type of product whose demand is high in a particular region and also try to come up with products with improved functionality. Companies may add value to its project through our analysis. It will also help to reduce the production cost of products.

23

7.2Future Scope

Our project can be extended to perform opinion mining using machine-learning techniques. We can perform opinion mining using supervised learning and unsupervised learning. Then, we can use game theory approach to compare both the techniques.

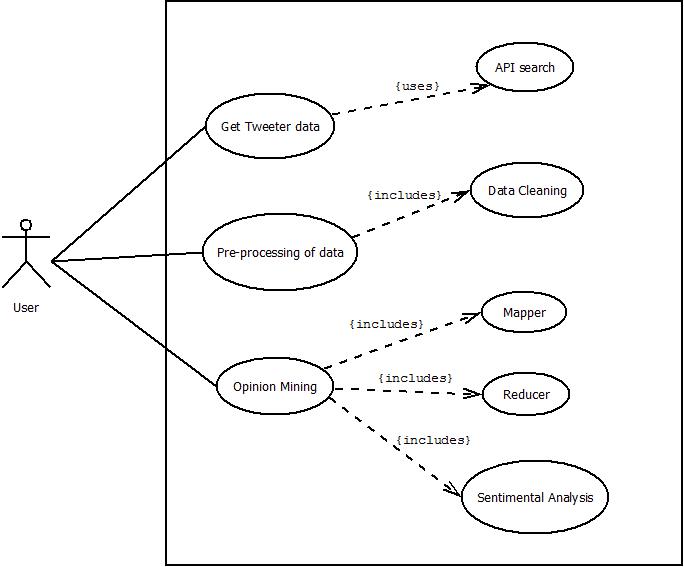
8. REFERENCES

1. AvitaKatal, Mohammad Wazid and R H Goudar. “Big Data: Issues, Challenges, Tools and Good Practices” Graphic Era University, Dehradun, India 2013.
2. Lorie Mann Bruce. “Game Theory Applied to Big Data Analytics in Geo Sciences and Remote Sensing” Mississippi State University, USA 2013.
3. Zhan Bu, Huijia Li, Jie Cao, Zhiang Wu, Lu Zhang Elseveir. “Game Theory Based Emotional Evolution Analysis for Chinese Online Reviews”. Jiangsu Provincial Key Laboratory of E-Business, Nanjing University of Finance and Economics, Nanjing 210003, Chinaand school of management science and engineering, Central University of finance and economics, Bejing 10080, China 2016
4. K. Dave , S. Lawrence , D.M. Pennock , Mining the peanut gallery: Opinion ex-traction and semantic classification of product reviews, in: Proceedings of the 12th International Conference on World Wide Web, ACM, 2003, pp. 519–528 .
   1. S. Morinaga, K. Yamanishi , K. Tateishi , T. Fukushima , Mining product rep-utations on the web, in: Proceedings of the Eighth ACM SIGKDD Interna- tional Conference on Knowledge Discovery and Data Mining, ACM, 2002, pp. 341–349 .

[6] X. Fu, K. Yang , J.Z. Huang , L. Cui , Dynamic non-parametric joint sentiment topic mixture model, Knowl. Based Syst. 82 (2015) 102–114 . [21] E. Airoldi , X. Bai , R. Padman , Markov blankets and meta-heuristics search: Sen- timent extraction from unstructured texts, in: Advances in Web Mining and Web Usage Analysis, Springer, 2006, pp. 167–187 .

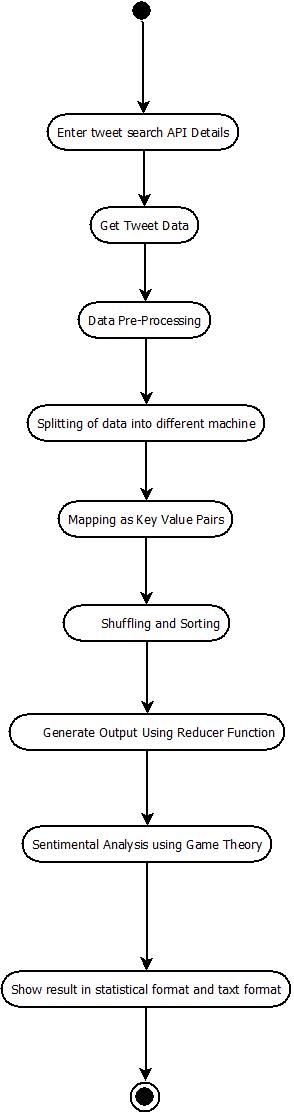
24

APPENDIX A



*Figure 13:Use Case Diagram*

25



*Figure 14:Activity Diagram*

26

APPENDIX B

Flume Configuration code

MyTwitAgent.sources = Twitter MyTwitAgent.channels = MemChannel MyTwitAgent.sinks = HDFS

MyTwitAgent.sources.Twitter.type = flume.mytwittersource.MyTwitterSourceForFlume MyTwitAgent.sources.Twitter.channels = MemChannel MyTwitAgent.sources.Twitter.consumerKey = XXXXXXXXXXXXXXXXXXXXXX MyTwitAgent.sources.Twitter.consumerSecret =

XXXXXXXXXXXXXXXXXXXXXXX

MyTwitAgent.sources.Twitter.accessToken =

XXXXXXXXXXXXXXXXXXXXXXXXXXX MyTwitAgent.sources.Twitter.accessTokenSecret =

XXXXXXXXXXXXXXXXXXXXXXXXXXXXXX MyTwitAgent.sources.Twitter.keywords = nokia MyTwitAgent.sinks.HDFS.channel = MemChannel MyTwitAgent.sinks.HDFS.type = hdfs MyTwitAgent.sinks.HDFS.hdfs.path = hdfs://localhost:9000/tweets MyTwitAgent.sinks.HDFS.hdfs.fileType = DataStream MyTwitAgent.sinks.HDFS.hdfs.writeFormat = Text MyTwitAgent.sinks.HDFS.hdfs.batchSize = 1000 MyTwitAgent.sinks.HDFS.hdfs.rollSize = 0 MyTwitAgent.sinks.HDFS.hdfs.rollCount = 10000 MyTwitAgent.channels.MemChannel.type = memory MyTwitAgent.channels.MemChannel.capacity = 10000 MyTwitAgent.channels.MemChannel.transactionCapacity = 1000

Tweets Table creation

CREATE EXTERNAL TABLE TWEETS

(

idBigint,

created\_at string, source string, favoritedboolean, retweet\_countint, retweeted\_statusstruct<text:string,

user1:struct<screen\_name:string, name:string>

>, entitiesstruct<urls:array<struct<expanded\_url:string>>,

user\_mentions:array<struct<screen\_name:string, name:string>>, hashtags:array<struct<text:string>>

>,

text string,

27

user1 struct<screen\_name:string,name:string,friends\_count:int,followers\_count:int,statuses\_co unt:int,verified:boolean,utc\_offset:string,time\_zone:string>, in\_reply\_to\_screen\_namestring,yearint,monthint,dayint,hourint)

Row format serde 'com.cloudera.hive.serde.JSONSerDe' location '/home/ashish/Desktop/data/tweets';

Dictionary Table Creation

CREATE EXTERNAL TABLE dictionary( type string,

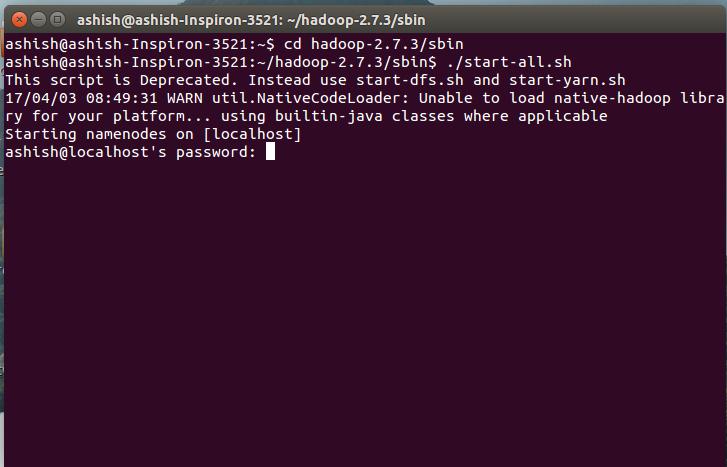
lengthint, word string, pos string, stemmed string, polarity string

)

ROW FORMAT DELIMITED FIELDS TERMINATED BY '\t'

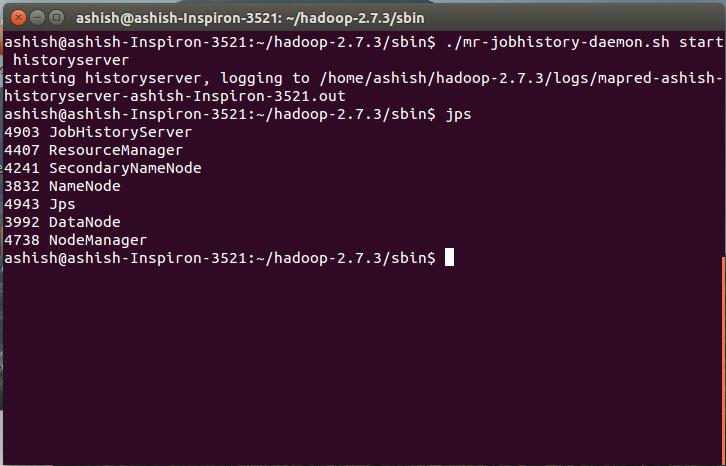
STORED AS TEXTFILE

LOCATION '/home/ashish/Desktop/dictionary';

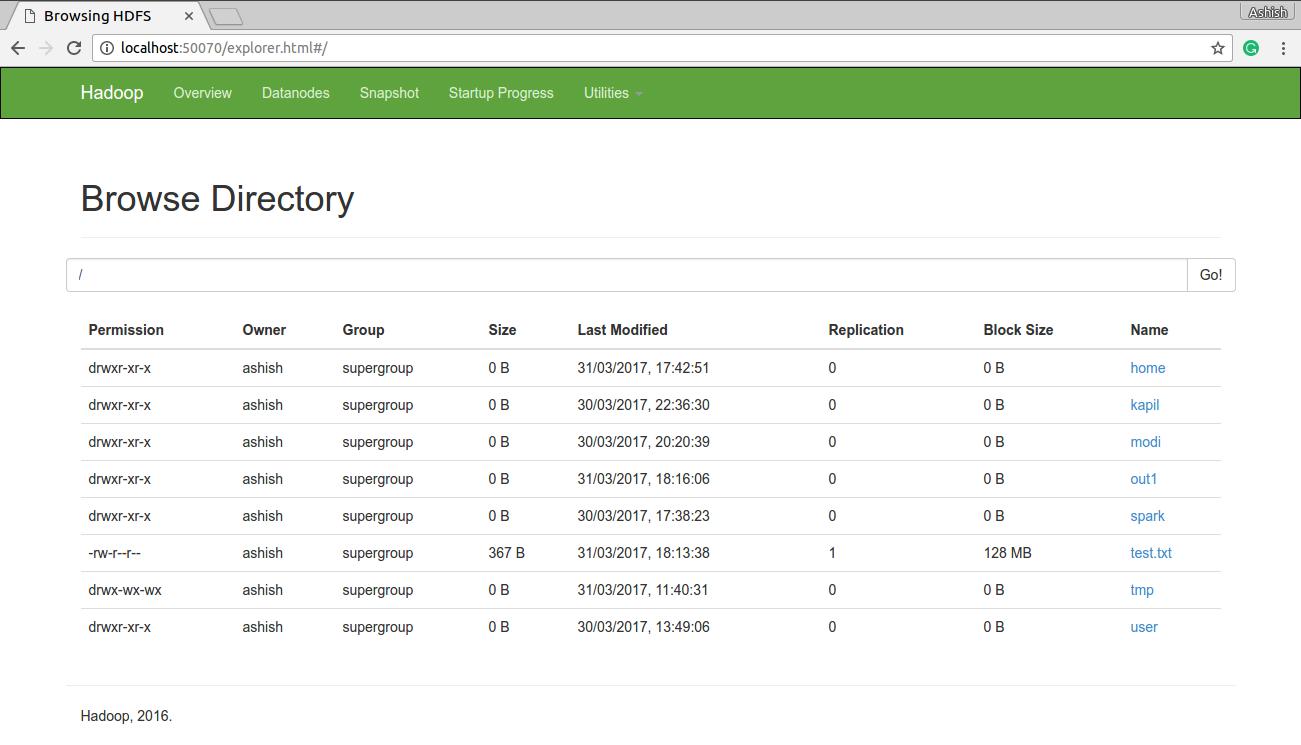


*Figure 15: Starting Daemons*

28

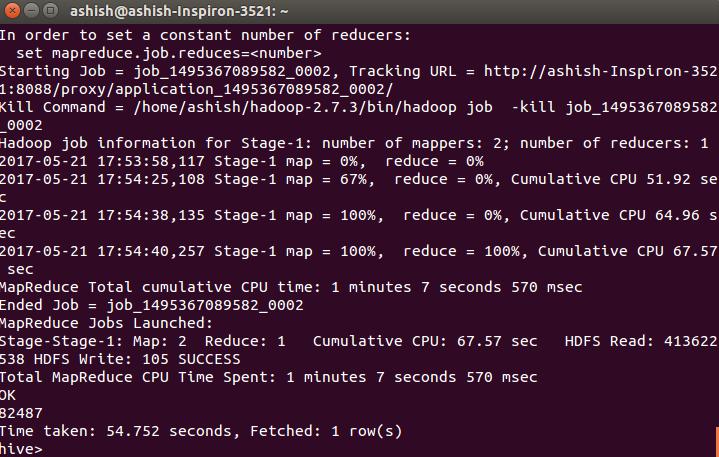


*Figure 16:Running Daemons*

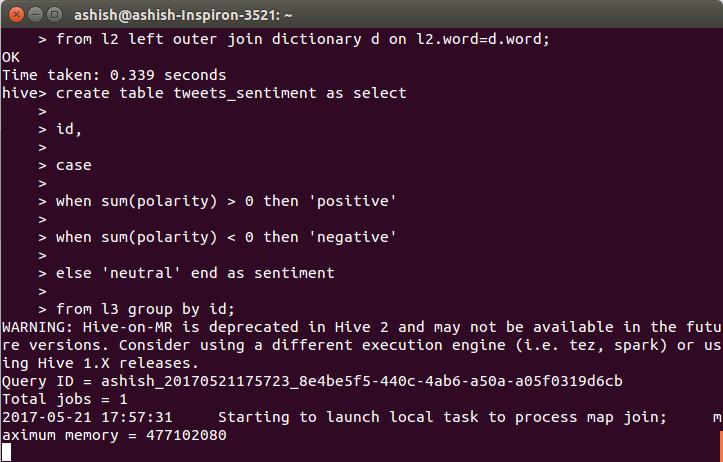


*Figure 17:HDFS*

29

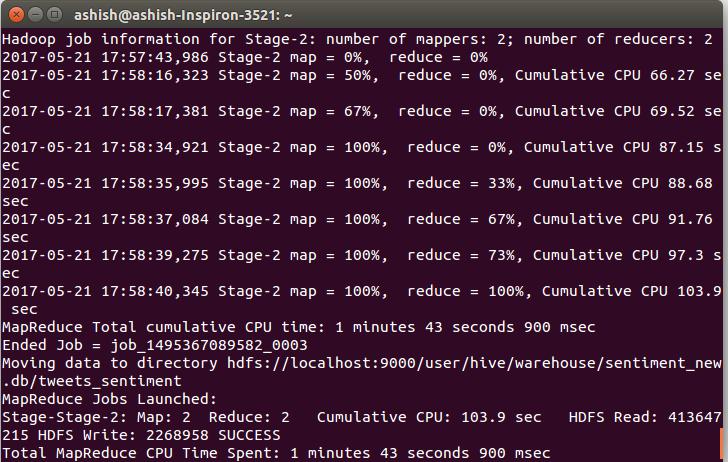


*Figure 18:Map-reduce*

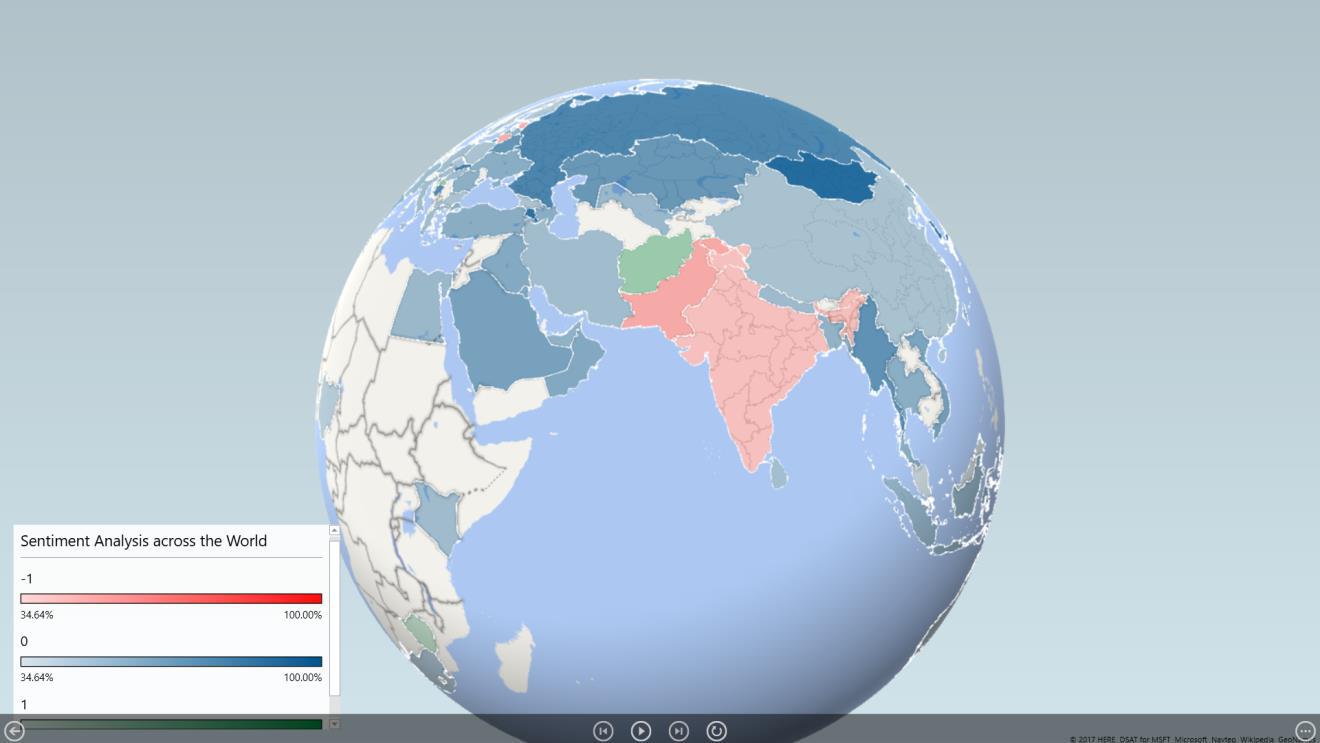


*Figure 19:Opinion-Mining-1*

30



*Figure 20:Opinion-Mining-2*



*Figure 21:Visualization*

31

SAMPLE EXAMPLE:

Tweet: The hotel rooms were very beautiful and comfortable. The Water and electricity facility was good. Only food quality was poor.

Step-1: Breaking into sentences

Sentence 1: The hotel rooms were very beautiful and comfortable.

Sentence 2: The Water and electricity facility was good.

Sentence 3: Only food quality was poor.

Step-2: Breaking into words

Sentence 1: {“The”, “hotel”, “rooms”, ”were”, ”very”, ”beautiful”, ”and”, ”comfortable”}

Sentence 2: {“The”, “Water”, “and”, “electricity”, “facility”, “was”, “good”} Sentence 3: {“Only”, “food”, “quality”, “was”, “poor”}

Step-3: Assigning sentiment value from dictionary

The 0

Hotel 0

Rooms 0

Were 0

Very 0

Beautiful 1

And 0

Comfortable 1

The 0

Water 0

And 0

Electricity 0

Facility 0

Was 0

Good 1

Only 0

Food 0

Quality 0

Was 0

Poor -1

Step-4: Calculating sentiment of whole tweet

Sum of sentiments = Beautiful(1)+Comfortable(1)+Good(1)+Poor(-1)=2

32

Since sum of sentiments is greater than 0. It is a positive tweet.

GAME THEORY IMPELMENTATION:

We calculated the percentage of positive, negative and neutral sentiments for both the approaches and tried to compare them.

Let us suppose, A+, A- and AN denotes the positive, negative and neutral sentiments for approach A respectively. Similarly, B+, B- and BN denotes for approach B.

The Game can be designed in following manner:

Approach B

Approach A

|  |  |  |
| --- | --- | --- |
|  | Data-set-1 | Data-set-2 |
| Data-set-1 | D1\_1+,D1\_2+ | U1,U2\* |
| Data-set-2 | U2,U1\* | D2\_1+,D2\_2+ |

Let us suppose the probability of positive sentiments of Approach A and data-set-1 is ‘p’ and for negative, it is (1-p). Similarly, for approach B, ‘q’ and (1-q).

For nash equilibrium, mixed strategy should be calculated. The process is:

|  |  |  |
| --- | --- | --- |
| For approach A data-set-1, (D1\_1+).p +U1.(1-p) | ………..(1) | |
| For approach A data-set-2, U2.p +(D2\_1+).(1-p) | ………..(2) | |
| For approach B positive sentiments, (D1\_2+).q +U1\*.(1-q) | | ………..(3) |
| For approach B positive sentiments, U2\*.q +(D2\_2+).(1-q) | | ………..(4) |

Value of p can be calculated by equating equation (1) and (2) and value of q can be calculated using equation (3) and (4). We will prefer the Approach having higher probability.

33