A PROJECT REPORT

ON

"SENTIMENT ANALYSIS OF TWITTER FOR MOVIE RECOMMENDATION"

FOR

"GEEKS OF PUNE"

SUBMITTED BY

Abhijit Bhosale	**B80318513**
Shruti Gadkari	**B80318526**
Soumya Krothapalli	**B80318551**
Abhishek Kulkarni	**B803185552**

B.E. (INFORMATION TECHNOLOGY)

UNDER THE GUIDANCE OF

NIMESH DESAI

DEPARTMENT OF INFORMATION TECHNOLOGY

PESs
MODERN COLLEGE OF ENGINEERING,
SHIVAJINAGAR, PUNE-5

UNIVERSITY OF PUNE

*** 2014-2015 ***

A PROJECT REPORT

ON

"SENTIMENT ANALYSIS OF TWITTER FOR MOVIE RECOMMENDATION"

SUBMITTED BY

Abhijit Bhosale	**B80318513**
Shruti Gadkari	**B80318526**
Soumya Krothapalli	**B80318551**
Abhishek Kulkarni	**B803185552**

B.E. (INFORMATION TECHNOLOGY)

UNDER THE GUIDANCE OF

Mrs. Sampda Kulkarni

DEPARTMENT OF INFORMATION TECHNOLOGY

PESs
MODERN COLLEGE OF ENGINEERING,
SHIVAJINAGAR, PUNE-5

UNIVERSITY OF PUNE

*** 2014-2015 ***

Progressive Education Societys
Modern College of Engineering, Shivajinagar,
Pune 411005.

C E R T I F I C A T E

This is to certify that the following students of Final Year Information Technology have successfully completed the project entitled **"SENTIMENT ANALYSIS OF TWITTER FOR MOVIE RECOMMENDATION"** in the academic year 2014-2015.

The Group Members names are:

B80318513:Abhijit Bhosale

B80318526:Shruti Gadkari

B80318552:Soumya Krothapalli

B80318551:Abhishek Kulkarni

This is in partial fulfillment of Bachelor of Information Technology under University of Pune,

Place: Pune

Date:

Mrs.Samapda Kulkarni		Mrs. S. D. Deshpande
(Guide)	External Examiner	(Head of Department)

ACKNOWLEDGEMENT

We take this opportunity to thank our project guide **Mrs. Sampada Kulkarni** and Head of the Department **Mrs. S. D. Deshpande** for their valuable guidance and for providing all the necessary facilities, which were indispensable in the completion of this project report. We are also thankful to all the staff members of the Department of Information Technology of **Progressive Education Societys Modern College of Engineering** for their valuable time, support, comments, suggestions and persuasion. We would also like to thank the institute for providing the required facilities, Internet access and important books.

Name of Students
Abhijit Bhosale
Shruti Gadkari
Soumya Krothapalli
Abhishek Kulkarni

ABSTRACT

Twitter currently receives about 500 million tweets per day, through which people all over the world share their comments regarding a wide range of topics. A large number of tweets include opinions about movies. With each day, twitter is increasingly becoming a more familiar platform for sharing opinions and one of the largest social media sites. Hence the need to analyze the sentiments expressed through tweets arises.

This project provides a way of analyzing the sentiments expressed on twitter for providing review for a movie. Sentiment analysis basically aims at determining the attitude of the speaker or the writer with respect to the topic or overall feeling in a document. This project mainly concentrates on backend processing of Tweet data collected from Twitter API. Tweets are collected using Hashtags and keywords related to the topic. This is done using a tool called Flume which is a part of Hadoop Ecosystem. Collected Tweets are stored in Hadoop Distributed File System(HDFS). Textual analysis is done to extract the sentiments from the Tweets. This is purely a Backend implementation. The results of Sentiment Analysis are stored again back to HDFS. This result is then fetched from HDFS and visualized using Excel.

Contents

1 INTRODUCTION

1.1 Problem Statement

1.To analyze the sentiments of people about a particular movie expressedthrough Twitter
2.To predict whether the people find the movie worthwatching.

1.2 Project Scope

1.To determine the polarity of sentiments expressed through tweets, to predict whether the people find the movie worth watching.
2.To deal with tweets that express neutral sentiments.
3.Handling tweets that contain the word but where the phrase before but is positive and the phrase after but is negative or vice versa.
4.To deal with tweets containing negation (dont, not, cant etc).

1.3 Project Objectives

Today the review systems available are through print and electronic media which are based on opinion of set of expertswho do not represent the opinion of the masses. The number of opinions shared on twitter is growing abundantly. Analysing the tweets with respect to specific movie will give a more clearer, better and realistic review about the movie.
Thus, our project will help to get enhanced insights about mentioned drawbacks.

1.4 Assumptions

1.4.1 Authentication of tweets:

For processing, tweets need to be collected from Twitter API. We assume that each tweet collected, is authentic.

1.4.2 Non-Biased nature of tweets:

We assume that each Twitter account is genuine and not fake. Also, the tweets expressed by users are not tweeted for the purpose of influencing or biasing the overall opinion in general.

1.4.3 Genuineness of tweets

It is assumed that each tweet expressed by the user is genuine and reflects their personal opinion.

1.5 Dependencies

1.5.1 Training Data Set:

The functioning of algorithm is highly dependent on the training dataset, which is used to determine the polarity of the tweets. More accurate datasets will produce more accurate output (polarity). Also, as the size of training dataset increases the accuracy of result will be affected.

1.5.2 Spell Check of tweets:

Before the tweets are processed, the spellings in tweets are checked. This is done because tweets are expressed in free language where people write words in short forms; and our algorithm does not correct the spellings. Our algorithm is based on Nave Bayes unigram model which deals with single words. Hence, it is a necessity that the words need to be meaningful in order to obtain accurate results.

1.5.3 Phrase Expansion:

Words with joint negations (like cant for cannot, doesnt for does not, didnt for did not) need to be expanded, as the training data set might not contain these words. This will help to increase the accuracy in handling negative cases.

1.6 General Contraints

1.6.1 English Language:

In our project we are considering training dataset which contains list of words in English language. Hence we consider only those tweets writtenin English language. If this constraint is satisfied then only we can implement next step of pre-processing of tweets which consists of spell checking and phrase expansion.

1.6.2 Live Data Nodes:

The processing is done using Map Reduce framework and it is necessary that all the data nodes in the cluster must be live i.e. they must be functional and in running state.

1.6.3 Hadoop Services:

All Hadoop Services must be in running state in order to start the processing of tweets.

1.6.4 Ignoring Emoticons:

The algorithm that we are using does not consider the emoticons to determine the polarity of tweets, therefore, they are ignored.

1.7 Literature Survey

Twitter is a micro-blogging service that has fast become commonplace in our daily lives. Twitter allows users and organizations to publish messages, communicate with other users in real time, and even use scripted programs or bots to perform tasks which utilize delivery of short messages. The explosion of twitter usage these days has made it a platform where people express themselves comfortably. The volume of opinions thus generated are of great importance to the marketer as customer emotions are indirect motivators of purchase behavior[1]. Customer feedback cards, surveys, interviews were a few among the traditional methods adopted to understand what exactly the customer felt. But these manual analysis tasks were time consuming and not efficient enough to handle large data sets. Sentiment analysis addresses these problems by systematically collecting and analyzing online sentiments from a very large sample of customers in real time[1].

Movies provide entertainment to almost everyone and analyzing opinions about a movie will be beneficial to both the marketing and producing team as well as the viewers in general. This analysis would provide a basis for the success of the movie. The tweets are obtained from the twitter website using the twitter API which would provide us with a large source of information for conducting the sentiment analysis. The tweets are first checked for relevance to the movie being analyzed by using a list of keywords. The system is trained using the training dataset which makes it capable to analyze the input tweets. The input tweets are then checked word by word and noise is filtered and the words expressing opinion are taken into account. The sentiment analysis of the tweets is performed by the system after which the tweets are classified into positive, negative and neutral categories[1].

SENTIMENT ANALYSIS OR OPINION MINING

Emotions are feelings generated from both conscious and unconscious processing. The emotional assessment of a situation is a general evaluation of a situation that affects our opinions and decisions that we make[1] . Sentiment analysis or Opinion Mining is a computational study of peoples opinions, attitudes and emotions toward

an entity. It refers to a broad area of natural language processing, computational linguistics and text mining to identify and extract subjective information from the available data. The main aim is to determine the attitude of the speaker; which may be their judgment or evaluation, their affective state or the intended emotional communication. The two expressions SA or OM are interchangeable and express a mutual meaning. However, some researchers stated that OM and SA have slightly different notions. Opinion Mining extracts and analyzes peoples opinion about an entity while Sentiment Analysis identifies the sentiment expressed in a text then analyzes it[2]. Sentiment Analysis is a Natural Language Processing and Information Extraction task that aims to obtain writers feelings expressed in positive or negative comments, questions and requests, by analyzing a large numbers of documents.
Liu et al. (2009) defines a sentiment or opinion as a quintuple-
¡oj, fjk, soijkl, hi, tl ¿
where
oj is a target object,
fjk is a feature of the object oj,
hi is an opinion holder,
tl is the time when the opinion is expressed,
soijkl is the sentiment value of the opinion of the opinion holder hi on feature fjk of object oj at time tl; soijkl is +ve,-ve, or neutral, or a more granular rating.[3]

2 REQUIREMENT ANALYSIS

2.1 Introduction

2.1.1 Purpose:

This SRS is intended to provide a detailed description of how a Sentiment Analysis Application for movie Recommendation can be build using commodity Hardware and open source Software such as Hadoop.

2.1.2 Document conventions:

Main Section Titles
1.Font: Times New Roman
2.Face: Bold
3.Size: 16
Sub Section Titles
1.Font: Times New Roman
2.Face: Bold
3.Size: 14
Other Text Explanations
1.Font: Times New Roman
2.Face: Normal
3.Size: 12

2.1.3 Intended Audience:

This SRS is intended for reading to developers ,users, and testers.

2.1.4 Scope of Project

1. To determine the polarity of sentiments expressed through tweets, to predict whether the people find the movie worth watching.
2. To classify the polarities into three classes as Positive, Negative and Neutral

2.1.5 Definitions, Acronyms, and Abbreviations:

1.Apache Hadoop: Apache Hadoop is an open-source software framework for storage and large-scale processing of data-sets on clusters of commodity hardware.
2. Apache Flume: Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data.

3. HDFS : Hadoop Distributed File System.
4. Apache Hive:Apache Hive is an open-source data warehouse system for querying and analyzing large datasets stored in Hadoop files.
5. Map Reduce:Map Reduce is a batch-based, distributed computing framework. It is a programming model for large scale data processing.

2.1.6 References:

1. Anonymous. High Availability for Hadoop Distributed System (HDFS). http://www.cloudera.com
2. Anonymous. Apache Hadoop. http://hadoop.apache.org
3. Dan Jurafsky. Natural Language Processing[Video]. Retrieved from http://www.Coursera.com

2.2 Overall Description

2.2.1 Product Perspective:

This Application is meant to provide analysis of the overall mood of the Twitter users about a particular movie. It can be embedded with existing systems such as IMDB and Rotten Tomatoes to increase the overall accuracy of prediction of movie success especially to the movie production team

2.2.2 Product Functions:

To analyze the sentiments or opinions of Twitter users about a particular movie and classify them into 3 classes as
1.Positive
2.Negative
3.Neutral

2.2.3 User Classes and Characteristics:

1. Movie Production Team:
The Movie Production team is a group of people associated with the production department of the movie. This team is concerned and is responsible for making the movie successful.
2. Movie Audience/viewers:
The Movie Audience are the one who view the movie and express their opinions about it.

2.2.4 Operating Environment:

1.Hardware: Commodity Hardware like Computer with Intel i3 processer, 4 GB RAM.

2.Operating System: CentOs -6.0
3.Softwares: Hadoop Ecosystem tools.

2.3 External Interface Requirements

2.3.1 User Interfaces:

UI-1 : The user can view the overall Sentiment polarity through an Android App

2.3.2 Hardware Interfaces:

HI-1 : It consists of Master consisting of Name Node of Hadoop Distributed Files System.
HI-2 : It consists of Slave consisting of Data Node of of Hadoop Distributed Files System.

2.3.3 Software Interfaces:

SI -1: It consists of Flume which fetches the Tweets from Twitter API to HDFS.
SI -2: It consists of Hive which pre-processes the Tweets.
SI -3: It consists of Map-reduce Framework for analyzing the raw Tweet polarity.

2.3.4 Communications Interfaces:

CI- 1 : It consists of Flume connecting to the Twitter API for collection of Tweets.

2.4 Functional Requirements

1. Connect to Twitter and fetch tweets from Twitter API using flume and store in HDFS.
a. Retrieve the metadata of each tweet along with text content.
1. Tweet Id.
2. Sender Id.
3. Receiver Id.
4. Sender Name.
5. Receiver Name.
6. Date and time of tweet creation.
7. Geo Coordinates (latitude and longitude).
8. Actual text of the tweet.
2. Extract significant or key phrases from each tweet.
3. Perform sentiment analysis on each tweet text and find out the sentiment of each tweet, and calculate the score may it is

a. Positive.
b. Negative.
c. Neutral (zero).
4. Store the results in HDFS and visualize them.

2.5 Non-Functional Requirements

1. Register the application with twitter and get the access keys.
2. The cluster needs high performance machines, with minimum 2.4 GHz processing speed with a physical memory of min. 4 GB.
3. Java class path set to external library jars.
4. Study the specifications and configuration setting of external libraries and APIs, while integrating with user application.

2.6 Other Non-functional Requirements

2.6.1 Performance Requirements:

1 For Collection of Tweets: High Speed Internet Connection [Broadband or 3G] is required for collecting Tweets at high speed.
2 For Sentiment Analysis: The Processing Speed is dependent on the number of nodes in the Cluster.

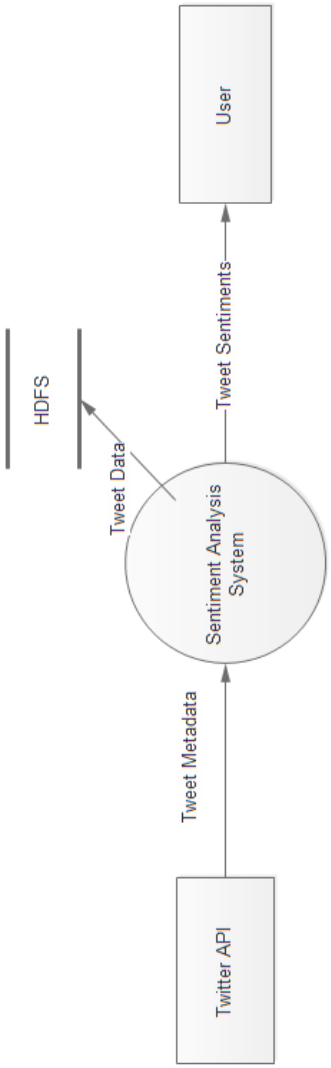
2.7 Quality Attributes

Availability:
This Application can work consistently even if some failures occur because of the availability offered by Hadoop itself.
Scalability:
This application can be easily scaled up and down as needed by adding new nodes to the Cluster.
Efficiency:
This Application gives very efficient results as needed as the quality Scalability is offered easily by it.

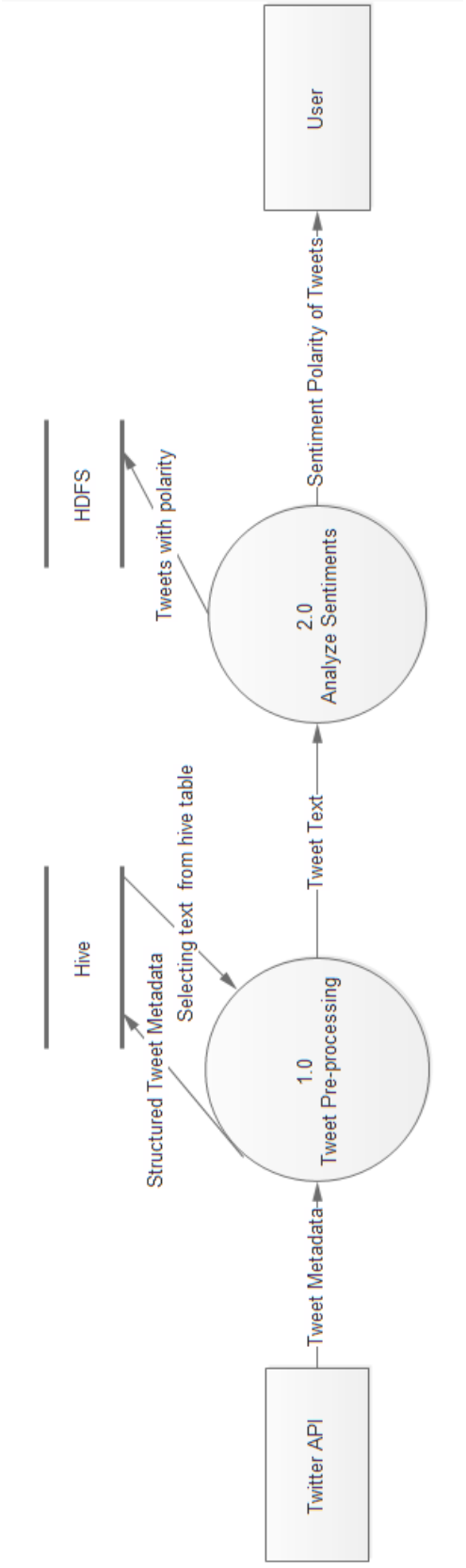
3 PROJECT DESIGN

3.1 DFD

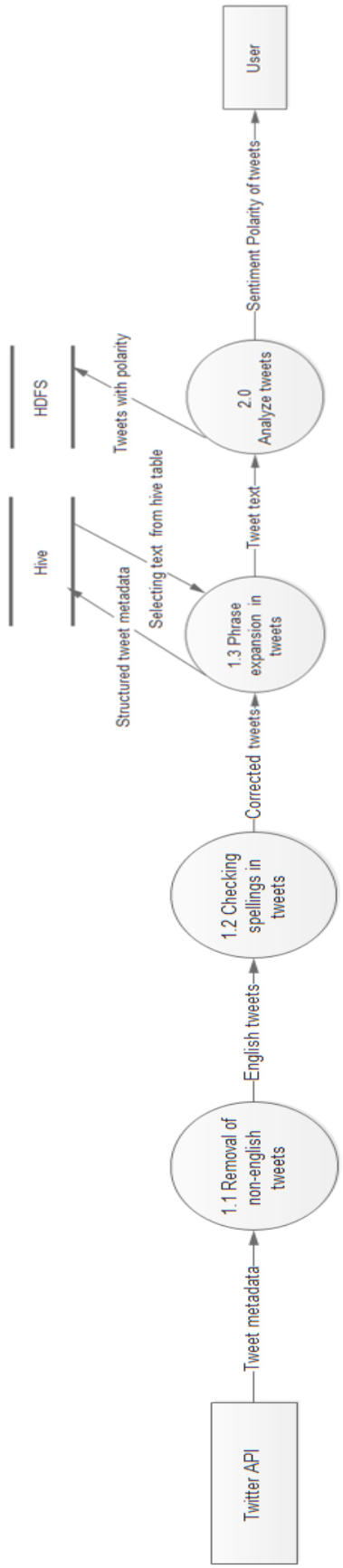
3.1.1 DFD Level 0:



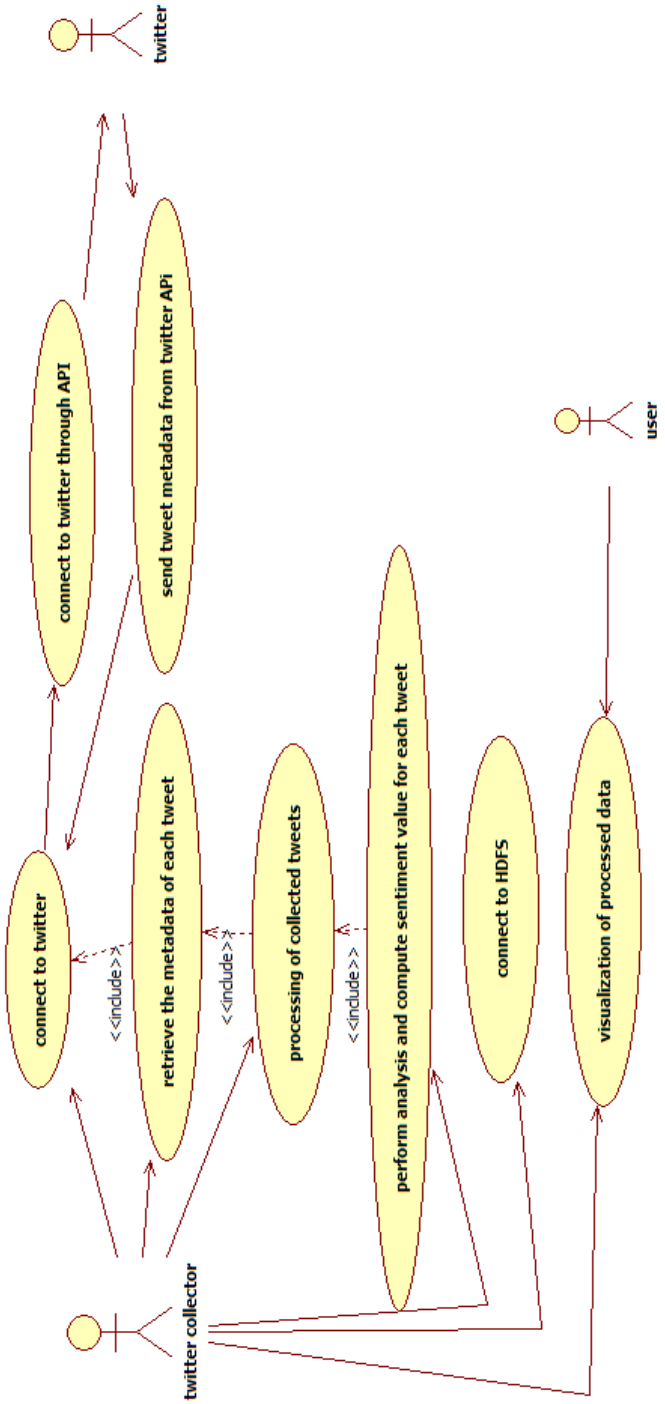
3.1.2 DFD Level 1:



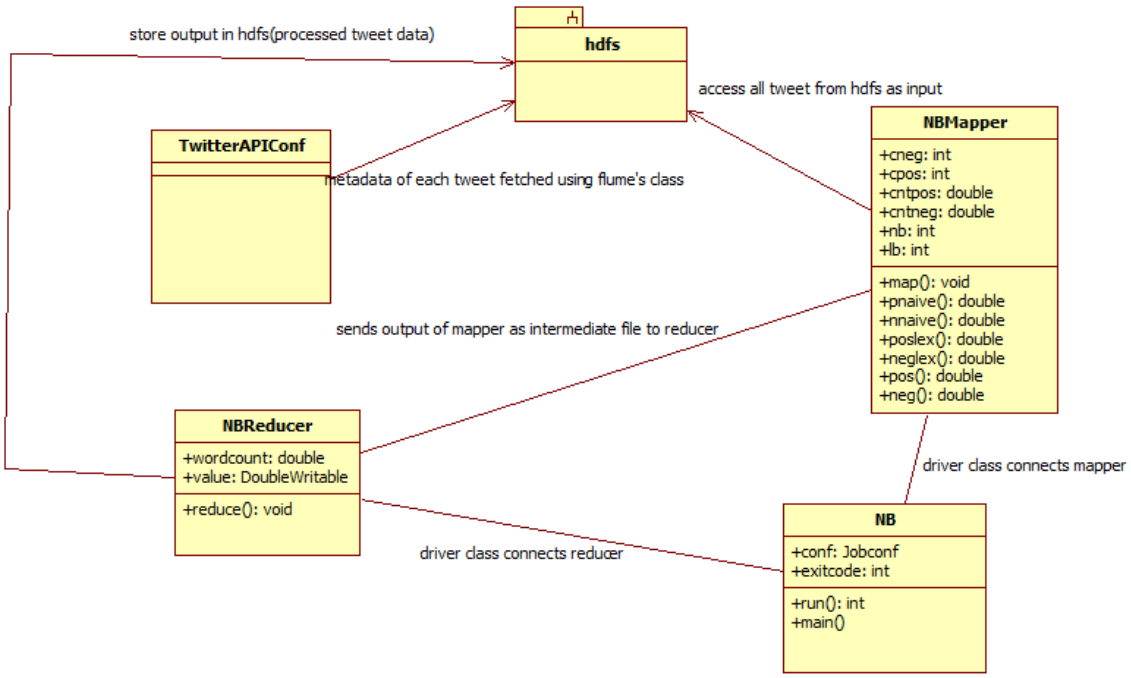
3.1.3 DFD Level 2:



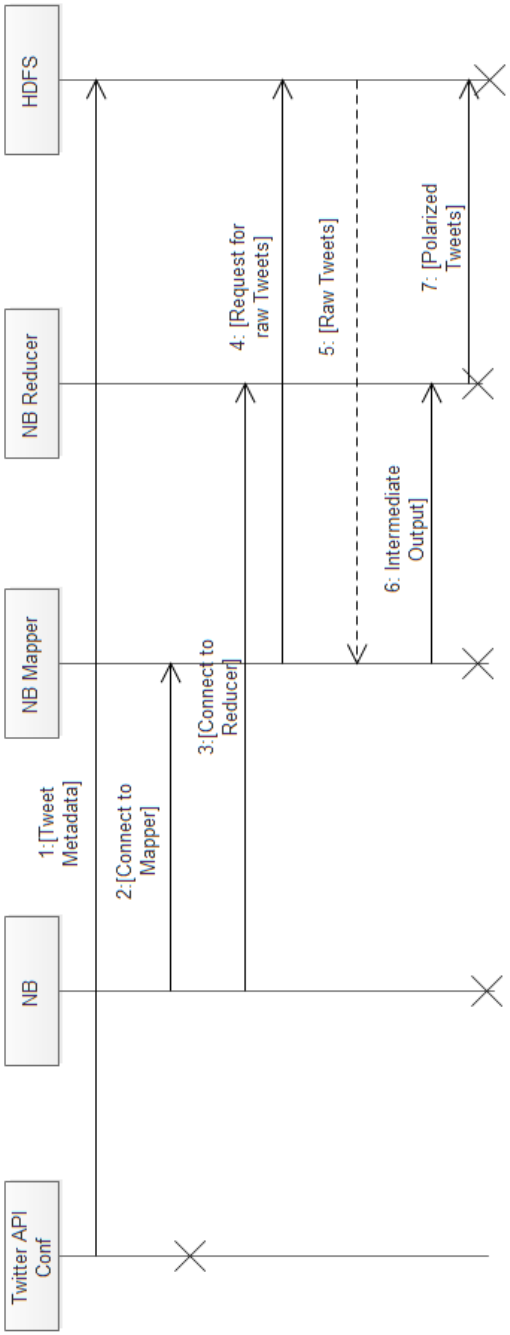
3.2 Use Case Diagram



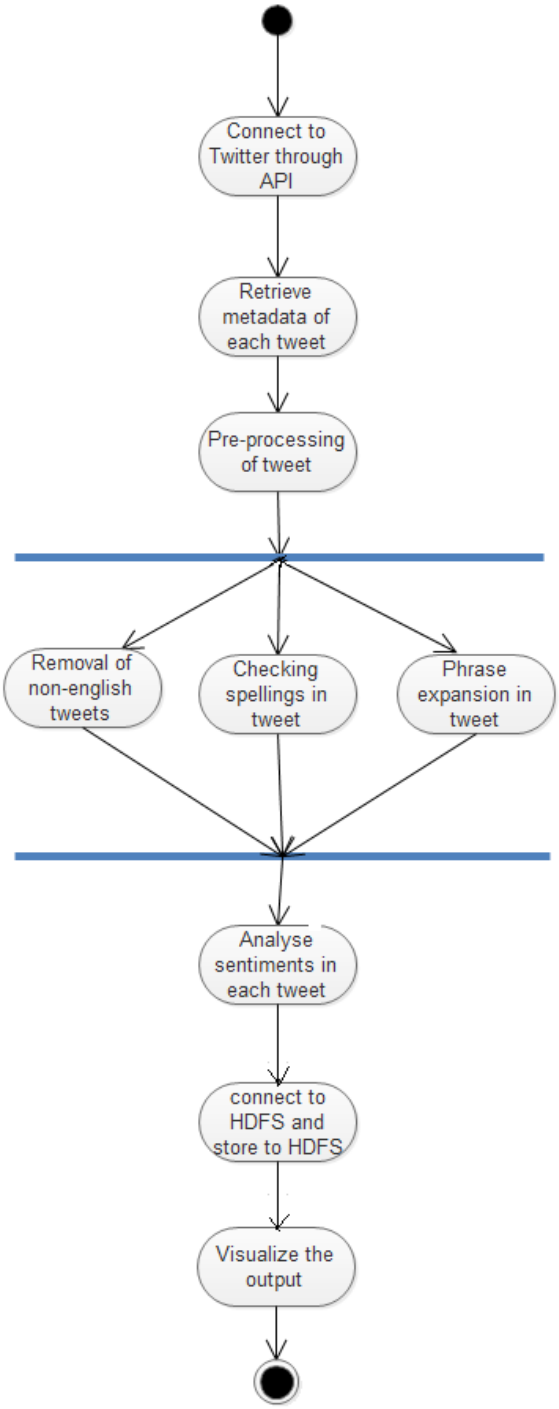
3.3 Class Diagram



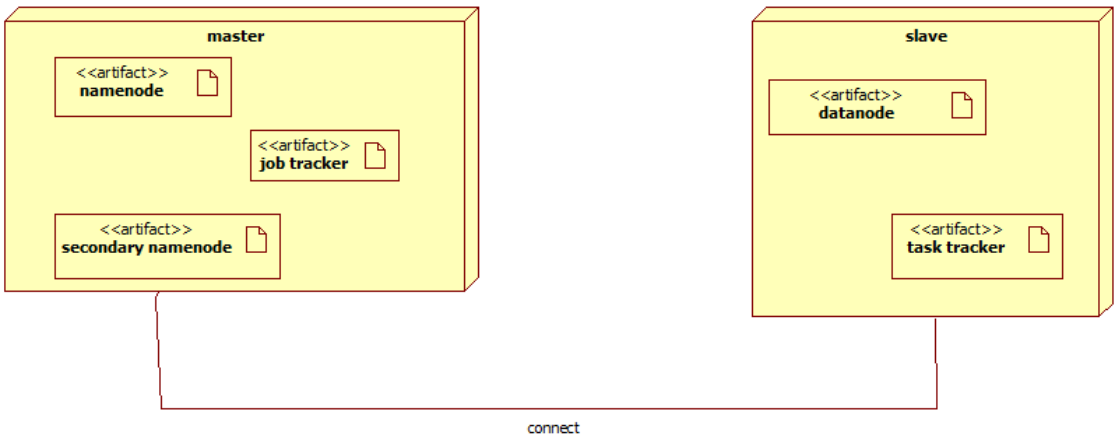
3.4 Sequence Diagram



3.5 Activity Diagram



3.6 Deployment Diagram



4 IMPLEMENTATION DETAILS

4.1 Project Architecture

4.1.1 Pipes and Filters Architectural Style:

Applications consist of a series of components in which component i produces output which is read and processed by component i+1, in the same order in which it is written by component i.
Meta Data is taken from Twitter server through Flume using API.
The collected meta data from Twitter is stored in HDFS through memory channel.
This data is processed by the Mapper and the result of it is sent to the Reducer. Reducer generates output.

4.2 Algorithm Used

4.2.1 Bayes Classifier:

We have used Naive Bayes classifier for determining the polarity of the tweet i.e. positive, negative or neutral tweet. Naive Bayes classifier model computes the posterior probability of a class, based on the distribution of the words in the document. The model works with the Bag of Words feature extraction. It uses Bayes Theorem to predict the probability that a given feature set belongs to a particular label. Naive Bayes is a conditional probability model: given a problem instance to be classified, represented by a vector $\mathbf{x} = (x_1, \ldots, x_n)$ representing some n features, it assigns to this instance probabilities $p(C_k|x_1, \ldots, x_n)$ for each of k possible outcomes or classes. Using Bayes' theorem, the conditional probability can be decomposed as $p(C_k|\mathbf{x}) = \frac{p(C_k)\ p(\mathbf{x}|C_k)}{p(\mathbf{x})}$.

In plain English, using Bayesian probability terminology, the above equation can be written as

$$\text{posterior} = \frac{\text{prior} \times \text{likelihood}}{\text{evidence}}.$$

In practice, there is interest only in the numerator of that fraction, because the denominator does not depend on and the values of the features are given, so that the denominator is effectively constant. The numerator is equivalent to the joint probability model $p(C_k, x_1, \ldots, x_n)$ which can be rewritten as follows, using the chain rule for repeated applications of the definition of conditional probability:

$$
\begin{aligned}
p(C_k, x_1, \ldots, x_n) &= p(C_k)\ p(x_1, \ldots, x_n | C_k) \\
&= p(C_k)\ p(x_1|C_k)\ p(x_2, \ldots, x_n | C_k, x_1) \\
&= p(C_k)\ p(x_1|C_k)\ p(x_2|C_k, x_1)\ p(x_3, \ldots, x_n | C_k, x_1, x_2) \\
&= p(C_k)\ p(x_1|C_k)\ p(x_2|C_k, x_1)\ \ \ldots p(x_n|C_k, x_1, x_2, x_3, \ldots, x_{n-1})
\end{aligned}
$$

4.3 Technology and Libraries used

We are using Apache Hadoop for data storage and processing, Apache Flume is used to collect the tweets from the twitter server and Apace Hive is used for converting the unstructured data in to structured form.

4.3.1 Apache Hadoop:

Apache Hadoop is on open source software framework for storage and large scale processing of data-sets on clusters of commodity hardware. It is a Java-based programming framework that supports the processing of large data sets in a distributed computing environment. It is fault tolerant, flexible, scalable and cost effective. It includes:
1. Hadoop Common: contains libraries and utilities needed by other Hadoop modules.
2. Hadoop Distributed File System (HDFS): a distributed file-system that stores data on commodity machines, providing very high aggregate bandwidth across the cluster. It includes Name Node, Secondary Name Node and Data Node.
3. Hadoop Map Reduce: a programming model for large scale data processing. Map Reduce is a batch-based, distributed computing framework. It includes Job Tracker, Task Tracker. MapReduce is a programming model for processing and generating large data sets with a parallel, distributed algorithm on a cluster. A MapReduce program is composed of a Map procedure that performs filtering and sorting and a Reduce procedure that performs a summary operation.

Map step: Master node takes large problem input and slices it into smaller sub problems distributes these to Data nodes. Data nodes processes smaller problem and hands back to master. Each data node applies the map function to the local data, and writes the output to a temporary storage. A master node orchestrates that for redundant copies of input data, only one is processed.

Shuffle step: Data nodes redistribute data based on the output keys produced by the map function, such that all data belonging to one key is located on the same Data node.

Reduce step: Master node takes the answers to the sub problems and combines them in a predefined way to get the output/answer to original problem. Data nodes now process each group of output data, per key, in parallel.

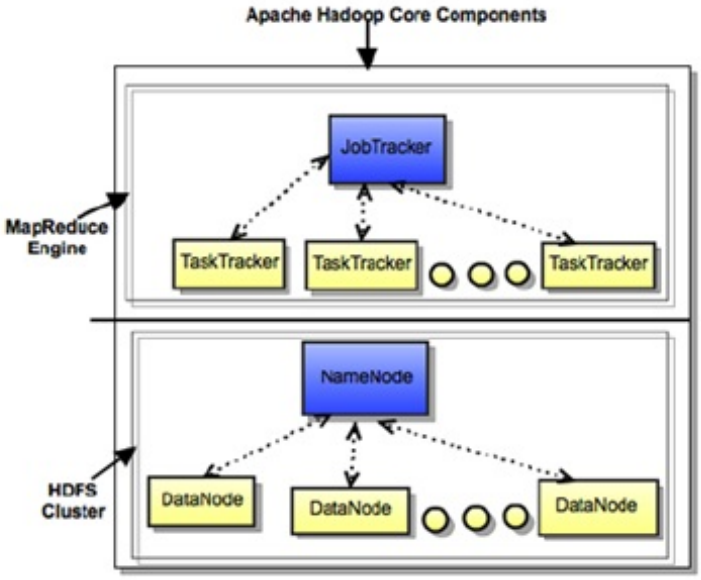


Fig: Core components for the Hadoop stack.

4.3.2 Apache Flume:

Flume is a distributed, reliable, and available service for efficiently collecting, aggregating, and moving large amounts of log data. It has a simple and flexible architecture based on streaming data flows. It is robust and fault tolerant with tunable reliability mechanisms and many failover and recovery mechanisms. Flume has following components:

1. Event: a singular unit of data that is transported by Flume.
2. Source: the entity through which data enters into Flume. Sources either actively poll for data or passively wait for data to be delivered to them.
3. Sink: the entity that delivers the data to the destination. A variety of sinks allow data to be streamed to a range of destinations. One example is the HDFS sink that writes events to HDFS.
4. Channel: the conduit between the Source and the Sink. Sources ingest events into the channel and the sinks drain the channel.
5. Agent: any physical Java virtual machine running Flume. It is a collection of sources, sinks and channels.
6. Client: produces and transmits the Events to the Source operating within the Agent.

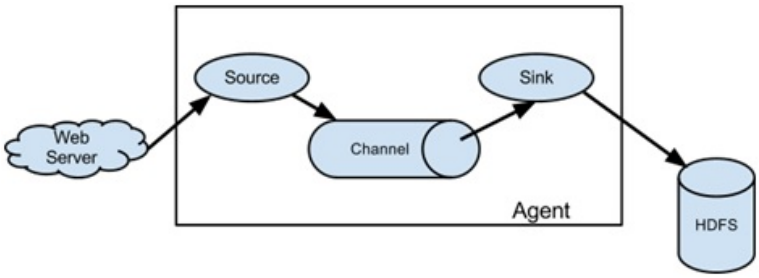


Fig: Data flow model for Flume.

4.3.3 Apache Hive:

Apache Hive is an open-source data warehouse system for querying and analyzing large datasets stored in Hadoop files. Hadoop is a framework for handling large datasets in a distributed computing environment. Hive has three main functions: data summarization, query and analysis. Hive also enables data serialization/deserialization and increases flexibility in schema design by including a system catalog called Hive-Metastore. Hive defines a simple SQL-like query language, called QL that enables users familiar with SQL to query the data.
It provides:
Tools to enable easy data extract/transform/load (ETL).
A mechanism to impose structure on a variety of data formats.
Access to files stored either directly in Apache HDFS or in other data storage systems such as Apache HBase.
Query execution via Map Reduce.

4.4 Database details

We are using Hadoop Distribute File System (HDFS) for storing the data, which is the storage component of Hadoop. The Hadoop distributed file system (HDFS) is a distributed, scalable, and portable file-system written in Java for the Hadoop framework. HDFS stores large files across multiple machines. It achieves reliability by replicating the data across multiple hosts. HDFS provides a homogeneous block distribution across the cluster. Hadoop enables businesses to easily access new data sources and tap into different types of data (both structured and unstructured) to generate value from that data.

4.5 Interface details

4.5.1 Twitter API:

We used Flume to collect data from the Twitter Streaming API, and forward it to HDFS. The Streaming APIs give developers low latency access to twitters global

stream of Tweet data. It helps streams of the public data flowing through Twitter, suitable for the following specific users or topics, and data mining. The individual messages streamed by this API are JSON encoded. The API provides access to Twitter data, both public and protected, and gives the meta data of the tweet.

4.6 Modules code

4.6.1 Android App code:

Main Page.java: package com.demo.new1;

```
import android.app.Activity;
import android.content.Intent;
import android.os.Bundle;
import android.view.Menu;
import android.view.View;
public class MainActivity extends Activity
{
@Override
protected void onCreate(Bundle savedInstanceState){
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_main);
}
public void gotointerstellar(View v )
{ Intent intent = new Intent("com.demo.new1.Interstellar");
startActivity(intent);
}
public void gotofastfurious7(View v )
{ Intent intent = new Intent("com.demo.new1.FastandFurious7");
startActivity(intent);
}
public void gotochappie(View v )
{ Intent intent = new Intent("com.demo.new1.Chappie");
startActivity(intent);
}
public void gotoub(View v )
{ Intent intent = new Intent("com.demo.new1.Unfinishedbusiness");
startActivity(intent);
}
public void gotojurassic(View v )
```

```
{ Intent intent = new Intent("com.demo.new1.JurassicWorld");
startActivity(intent);
}
public void gotoavng(View v )
{ Intent intent = new Intent("com.demo.new1.AvengersAgeofUltron");
startActivity(intent);
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.main, menu);
return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
// Handle action bar item clicks here. The action bar will
// automatically handle clicks on the Home/Up button, so long
// as you specify a parent activity in AndroidManifest.xml.
id = item.getItemId();
if (id == R.id.action_settings) {
return true;
}
return super.onOptionsItemSelected(item);
}
}
```

Intersellar.java:

```
package com.demo.new1;
import android.app.Activity;
import android.net.Uri;
import android.os.Bundle;
import android.view.Menu;
import android.view.MenuItem;
import android.view.View;
import android.widget.ImageView;
import android.widget.MediaController;
import android.widget.VideoView;
public class Interstellar extends Activity {
ImageView ing;
```

```
@Override
protected void onCreate(Bundle savedInstanceState) {
super.onCreate(savedInstanceState);
setContentView(R.layout.activity_interstellar);
String fileName="android.resource://" + getPackageName() + "/" + R.raw.intersellar;
VideoView vv= (VideoView)this.findViewById(R.id.videoView1);
vv.setVideoURI(Uri.parse(fileName));
MediaController mediaController = new MediaController(this);
mediaController.setAnchorView(vv);
vv.setMediaController(mediaController);
vv.start();
}
public void analysis(View v )
{
ImageView ing= (ImageView)this.findViewById(R.id.imageView1);
ing.setVisibility(View.VISIBLE);
}
@Override
public boolean onCreateOptionsMenu(Menu menu) {
// Inflate the menu; this adds items to the action bar if it is present.
getMenuInflater().inflate(R.menu.interstellar, menu);
return true;
}
@Override
public boolean onOptionsItemSelected(MenuItem item) {
// Handle action bar item clicks here. The action bar will
// automatically handle clicks on the Home/Up button, so long
// as you specify a parent activity in AndroidManifest.xml.
int id = item.getItemId();
if (id == R.id.action_settings) {
return true;
}
return super.onOptionsItemSelected(item);
}
}
```

4.6.2 Word Count Code in Map Reduce:

Driver Class: WordCount.java

```
import org.apache.hadoop.fs.Path;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.FileInputFormat;
import org.apache.hadoop.mapred.FileOutputFormat;
import org.apache.hadoop.mapred.JobClient;
import org.apache.hadoop.mapred.JobConf;
import org.apache.hadoop.conf.Configured;
import org.apache.hadoop.util.Tool;
import org.apache.hadoop.util.ToolRunner;
public class WordCount extends Configured implements Tool {
@Override
public int run(String[] args) throws Exception {
if (args.length != 2) { System.out.printf("Usage: %s [generic options] ¡input dir¿ ¡out-
put dir¿ n ",
getClass().getSimpleName());
ToolRunner.printGenericCommandUsage(System.out);
return -1;
}
JobConf conf = new JobConf(getConf(), WordCount.class);
conf.setJobName(this.getClass().getName());
FileInputFormat.setInputPaths(conf, new Path(args[0]));
FileOutputFormat.setOutputPath(conf, new Path(args[1]));
conf.setMapperClass(WordMapper.class);
conf.setReducerClass(SumReducer.class);
conf.setMapOutputKeyClass(Text.class);
conf.setMapOutputValueClass(IntWritable.class);
conf.setOutputKeyClass(Text.class);
conf.setOutputValueClass(IntWritable.class);
JobClient.runJob(conf);
return 0;
}
public static void main(String[] args) throws Exception {
int exitCode = ToolRunner.run(new WordCount(), args);
System.exit(exitCode);
}
```

}

Mapper Class: WordMapper.java

```
import java.io.IOException;
import java.util.*;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.LongWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Mapper;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.Reporter;
public class WordMapper extends MapReduceBase implements
Mapper¡LongWritable, Text, Text, IntWritable¿ {
@Override
public void map(LongWritable key, Text value,
OutputCollector¡Text, IntWritable¿ output, Reporter reporter)
throws IOException {
String s = value.toString().toLowerCase();
for (String word : s.split("
W+")) {
if (word.length() ¿ 0) {
output.collect(new Text(word), new IntWritable(1));
}
}
}
}
```

Reducer class: SumReducer.java

```
import java.io.IOException;
import java.util.Iterator;
import org.apache.hadoop.io.IntWritable;
import org.apache.hadoop.io.Text;
import org.apache.hadoop.mapred.OutputCollector;
import org.apache.hadoop.mapred.MapReduceBase;
import org.apache.hadoop.mapred.Reducer;
import org.apache.hadoop.mapred.Reporter;
public class SumReducer extends MapReduceBase implements
```

Reducer¡Text, IntWritable, Text, IntWritable¿ {
@Override
public void reduce(Text key, Iterator¡IntWritable¿ values,
OutputCollector¡Text, IntWritable¿ output, Reporter reporter)
throws IOException {
int wordCount = 0;
while (values.hasNext()) {
IntWritable value = values.next();
wordCount += value.get();
} output.collect(key, new IntWritable(wordCount));
}
}

References

[1] MARKETING RESEARCH: THE ROLE OF SENTIMENT ANALYSIS Meena Rambocas Department of Management Studies, The University of the West Indies (St. Augustine Campus), Trinidad and Tobago (mrambocas@yahoo.com) Joo Gama Laboratory of Artificial Intelligence and Decision Support and Faculty of Economics, University of Porto, Portugal (jgama@fep.up.pt)

[2] *Sentiment Analysis*(n.d) [Online] Available: http://en.wikipedia.org/wiki/Sentiment_analysis .

[3] Sentiment Analysis A Literature Survey by Subhabrata Mukherjee Roll No: 10305061 Supervisor:Dr. Pushpak Bhattacharyya June 29, 2012 Indian Institute of Technology, Bombay Department of Computer Science and Engineering.