|  |
| --- |
|  |
| Twitter Sentiment Analysis |
| Implementation and Workflow Document |
|  |
| **Abdul Raseeth** |
| **3/22/2013** |

|  |
| --- |
| [Type the abstract of the document here. The abstract is typically a short summary of the contents of the document. Type the abstract of the document here. The abstract is typically a short summary of the contents of the document.] |

Contents

[Introduction 2](#_Toc351742496)

[Components 3](#_Toc351742497)

[Scraper 3](#_Toc351742498)

[Classifier 4](#_Toc351742499)

[APIs exposed 4](#_Toc351742500)

[Classify 4](#_Toc351742501)

[Message 4](#_Toc351742502)

[Positive 4](#_Toc351742503)

[Negative 5](#_Toc351742504)

[To start apache solr 5](#_Toc351742505)

[To upload dictionary data to solr (First time only) 5](#_Toc351742506)

[Starting Analyzer to listen in specific port 5](#_Toc351742507)

[Sample 5](#_Toc351742508)

[RESTful API Services 6](#_Toc351742509)

[Trends 6](#_Toc351742510)

[Trendsentiment 7](#_Toc351742511)

[User 7](#_Toc351742512)

[Usersentiment 7](#_Toc351742513)

[Geo 7](#_Toc351742514)

[Geosentiment 7](#_Toc351742515)

[Java MR 7](#_Toc351742516)

[Prerequisite 7](#_Toc351742517)

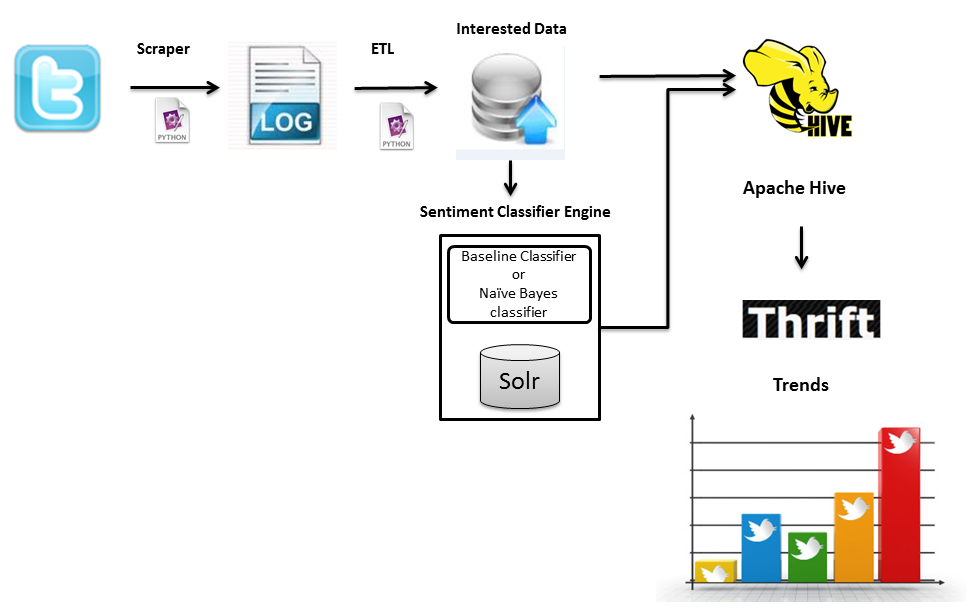
[To build and run 8](#_Toc351742518)

# Introduction

The objective is to expose APIs for Twitter trends based on demographics and also provides users sentiment about that product.

Python scripts are used to fetch Twitter data based on top trends in each demographics. Pig scripts are used to load the Twitter response into HDFS. Hive tables are loaded with the HDFS data path. Django MVC framework is used as UI which interacts with the hive using Thrift API. Piston framework is used to expose RESTful APIs.

Natural language training kit is used for sentiment analyzing in which the training sets can be feed based on positive and negative input tweets. Baseline classifier is also used to classify the text which is based on positive and negative dictionary words which are stored in Solr.



# Components

The system contains of five major modules

1. Scraper
2. PIG ETL Script
3. Java MRs
4. Restful APIs
5. Classifier Engine

# Scraper

* Python based script which authenticates with access token and retrieves Top trends messages based on Woeid.

twitter.Api(consumer\_key=consumerKey, consumer\_secret=consumerSecret, access\_token\_key=accessTokenKey, access\_token\_secret=accessTokenSecret)

* The access token is used to avoid the hour limit of the twitter restricted to 100hits per hour
* The script runs for the specified time interval specified in the start function

def start(interval):

* Stores the output results based on the requested time as timestamp with milliseconds

def getCurrentTime():

return int(round(time.time() \* 1000))

* The script avoids the duplicate tweets by storing the last returned object id (max\_id\_str) from the twitter response and post it back in the next request as since\_id which avoids the duplicates in the request level

Previous request’s object id is persisted

maxId\_trend[term] = result['max\_id\_str']

New request param updation

if maxId\_trend.has\_key(term):

params['since\_id'] = maxId\_trend[term]

* To run the scraper in background use the following command
* ***python Main.py [woeid] &***

# Classifier

The classifier is a sentiment analysis that identifies sentiment of the text message, the system in implemented with Baseline classifier which breaks the text message into words and compares it with dictionary of positive and negative words. By this classification the weightage of the positive and negative content is identifies and that classifies the sentiment of the message.

# APIs exposed

## Classify

This API accepts an array of text messages and classifies them.

URI - /classify/

Method – POST

Body – [“Text sentence1”,”Text sentence2”]

Hades URI – <http://192.168.10.231:8080/classify/>

## Message

This API accepts single message and classify them, mainly used by the Java MR for classification

URI -/message/

Method –POST

Body – Text sentence1

Hades URI – <http://192.168.10.231:8080/message/>

## Positive

This API is used to update the positive word dictionary in the solr

URI - /positive/

Method – POST

Body – [“Postiveword1”,”Positiveword2”]

Hades URI – <http://192.168.10.231:8080/positive/>

## Negative

This API is used to update the negative word dictionary in the solr

URI - /negative/

Method – POST

Body – [“Negativeword1”,” Negativeword2”]

Hades URI – <http://192.168.10.231:8080/negative/>

## To start apache solr

1)      Download apache solr above 4.1.

2)      Start solr with

cd solr- /example/

java -jar start.jar

## To upload dictionary data to solr (First time only)

1)      Download Twitter sentiment analysis source from [https://github.com/smartek/TwitterSentimentAnalysis/tree/master/trunk/src/Analysis\_Engine](https://by2prd0710.outlook.com/owa/redir.aspx?C=OdG-pPAB_kWTGzYYZWrEkWDP26xe-s8Icmd1B6FGVk3T4Q4dNGOq5K72WEGLbSIu_C2zgvClitc.&URL=https%3a%2f%2fgithub.com%2fsmartek%2fTwitterSentimentAnalysis%2ftree%2fmaster%2ftrunk%2fsrc%2fAnalysis_Engine)

2)      Navigate to Analysis\_Engine/Data folder

3)      Run python  solr\_upload.py (Dictionary values will be uploaded to the solr)

## Starting Analyzer to listen in specific port

1)      Navigate to Analysis\_Engine/ in downloaded source

2)      Run python classifier\_API.py 192.168.10.103:8080

Note – please specify the server ip rather than localhost:8080 if you are running in virtual machine

## Sample

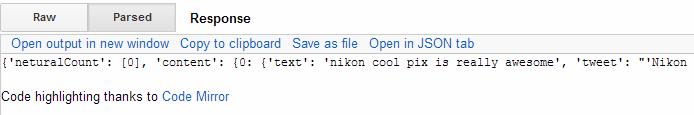
URL : [http://192.168.10.103:8080/classify/](https://by2prd0710.outlook.com/owa/redir.aspx?C=OdG-pPAB_kWTGzYYZWrEkWDP26xe-s8Icmd1B6FGVk3T4Q4dNGOq5K72WEGLbSIu_C2zgvClitc.&URL=http%3a%2f%2f192.168.10.103%3a8080%2fclassify%2f)

Method : POST

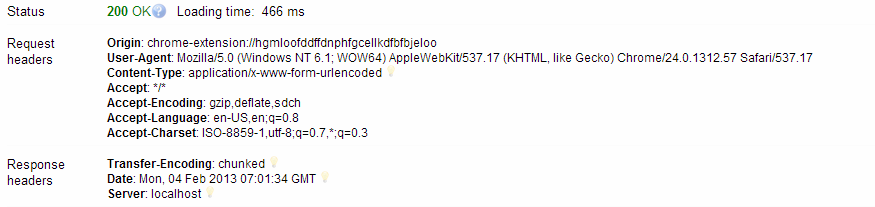
Body : ['Nikon Cool pix is really awesome','IPhone 5 is the worst iphone model ever!!!!!']



Response



{'neturalCount': [0], 'content': {0: {'text': 'nikon cool pix is really awesome', 'tweet': "'Nikon Cool pix is really awesome'", 'label': 'positive'}, 1: {'text': 'iphone 5 is the worst iphone model ever!!!!!', 'tweet': "'IPhone 5 is the worst iphone model ever!!!!!'", 'label': 'negative'}}, 'positiveCount': [1], 'negativeCount': [1]}



# RESTful API Services

These API services are to expose HDFS data as reports from the mysql server.

## Trends

Lists all the trends count and can be filtered using date

URI - /trend?date=YYYYMMDD

Method – GET

## Trendsentiment

List all the trends with positive and negative count that can be filtered with data

URI - /trendsentiment?date=YYYYMMDD

Method – GET

## User

List all the users count that can be filtered with data and userid

URI - /users?date=YYYYMMDD

Method – GET

## Usersentiment

List all the users with positive and negative count that can be filtered with data

URI - /usersentiment?date=YYYYMMDD

Method – GET

## Geo

List all the Geo count that can be filtered with data

URI - /geo?date=YYYYMMDD

Method – GET

## Geosentiment

List all the Geo with positive and negative count that can be filtered with data

URI - /geosentiment?date=YYYYMMDD

Method – GET

# Java MR

Mapreduce to process the HDFS data to filter interested data

## Prerequisite

Mapreduce depends on Classifier engine that should be running on a server

Classifier /message/ API URI should be configured in conf/mapreduce.properties file

Classifier depends on the solr with should be up and running

## To build and run

Java MR can be compiled with mvn install command

MR in hadoop can be executed with the following command

* bin/hadoop jar Jarfilename-version HDFS\_source\_path HDFS\_destination\_path