"Visualizing Sentiments of Reviews"

A Project report Submitted in partial fulfillment of the requirements
For the Degree of

Bachelor of Technology
in
Computer Science & Engineering

Submitted by:

Name	Roll No.
Jay Shankar Sah	**R100211067**
Juhi Bisht	**R100211025**
Surabhi Chaudhary	**R100211056**
Tejas Pandey	**R100211057**

Under the Esteemed Guidance of
Mr. Hitesh Kumar Sharma
Assistant Professor
Center for Information Technology

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES
College Of Engineering Studies
Center of Information Technology
Dehradun,

UNIVERSITY OF PETROLEUM AND ENERGY STUDIES

College Of Engineering Studies

Center of Information Technology,

Dehradun, 248007

CERTIFICATE

This is to certify that the project report titled

"Sentiment analysis of movie tweets"

has been submitted

In partial fulfillment for the award of the Degree

of

Bachelor of Technology

in Computer Science & Engineering

By

Jay Shankar Sah (R100211067)

Juhi Bisht (R100211025)

Surabhi Chaudhary (R100211056)

Tejas Pandey (R100211057)

of

University of Petroleum and Energy Studies

For the academic session 2011-2015

Internal Examiner External Examiner

CANDIDATE'S DECLARATION

I/We hereby certify that the project work entitled **"Sentiment Analysis on Movie Tweets"** in partial fulfillment of the requirements for the award of the Degree of BACHELOR OF TECHNOLOGY in COMPUTER SCIENCE & ENGINEERING with specialization in Open Source Software & Open Standards and submitted to the Department of Computer Science & Engineering at Center for Information Technology, University of Petroleum & Energy Studies, Dehradun, is an authentic record of my/ our work carried out during a period from **August**, 2014 to **December, 2014** under the supervision of **Mr. Hitesh Kumar Sharma, Assistant professor ,CIT UPES**.

The matter presented in this project has not been submitted by me/ us for the award of any other degree of this or any other University.

Jay Shanakar Sah

Juhi Bisht

Surabhi Chaudhary

Tejas Pandey

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

Date: _15th December,2014 **Mr. Hitesh Kumar Sharma**

Project Guide

Dr. Vinay Avasthi

Program Head – B. Tech CS-OSS & MFT

Center for Information Technology

University of Petroleum & Energy Studies

Dehradun – 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide **Mr. Hitesh Kumar Sharma, Mr. Pratyush Kumar Deka** for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our respected Program Head of the Department, **Dr. Vinay Avasthi**, for his great support in doing our project in **Sentiment Analysis** at **CIT**.

We are also grateful to **Dr. Manish Prateek, Associate Dean** and **Dr. Kamal Bansal,** Dean CoES, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our **friends** for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our **parents** who have shown us this world and for every support they have given us.

Name	**Jay Shankar Sah**	**Juhi Bisht**	**Surabhi Chaudhary**	**Tejas Pandey**
Roll No.	**R100211067**	**R100211025**	**R100211056**	**R100211057**

ABSTRACT

The process of sentiment analysis involves text analytics, linguistics and accepted language processing to determine and dig subjective information from source materials. It is commonly known for the term “opinion mining.” This process aims to determine how a certain person or group reacts to a topic they are being referred to. They react because they are either interested or involved. And, these reactions go to none other than their social media accounts which makes social media as one of the leading platforms in the internet where anyone can basically do sentiment analysis. Twitter and Facebook are two of the places where one can find a lot of sentiments and they are the best considerations whenever opinion mining is done.

Companies mostly benefit from sentiment analysis today. Some refer to it as social media analysis as well, since it also typically analyzes the ongoing activities on major social networking sites. Companies see sentiment analysis as a major aid in measuring sales and improving their marketing strategies as well. To accomplish this, some companies develop their own tools and others rely on outsourcing companies that specialize in sentiment analysis.

The company involved in sentiment analysis are muWebFluenz ,semantria ,rumbleLabs. It is one of the most growing area in computer science involving lot of research and economic demand by the industry. As Knowledge Learning from the data has become most critical part of all the company irrespective of the industrial vertical or sector.

In this project technology used are:-

1) Python tweepy.
2) R Programming
3) R Studio

The motive of the project is to collect data from twitter, clean the tweets, process the tweets and apply classification algorithms like rpart and Random Forest Tree to the data set and check the sentiment of the reviewers about the movie as positive or negative.

TABLE OF CONTENTS

LIST OF FIGURES

S.No. Figure Page No

1. INTRODUCTION

1.1. History

The web is huge repository of data. In recent years, the exponential increase in the Internet usage and exchange of public opinion is the driving force behind Sentiment Analysis today. The explosion of Web 2.0 and has led to increased activity in micro blogging, tagging, social bookmarking and Contributing to RSS etc. Many sites on the Web allow users to write their experiences or opinion about a product or service in form of a review. The Web is now full of user reviews for different items ranging from mobile phones, holiday trips, and hotel services to movie reviews etc. It is interesting to observe that these reviews not only express opinions of a group of users but is also a valuable source for harnessing collective intelligence. As a result there has been an eruption of interest in people to mine these vast resources of data for opinions. Opinion mining is the computational treatment of opinions and sentiments. The analysis of this data to extract latent public opinion and sentiment is a challenging task.

1.2. Requirement Analysis

Word of mouth (WOM) is the process of conveying information from person to person and plays a major role in customer buying decisions. In commercial situations, WOM involves consumers sharing attitudes, opinions, or reactions about businesses, products, or services with other people. WOM communication functions based on social networking and trust. People rely on families, friends, and others in their social network. Research also indicates that people appear to trust seemingly disinterested opinions from people outside their immediate social network, such as online reviews. This is where Sentiment Analysis comes into play. Growing availability of opinion rich resources like online review sites, blogs, social networking sites have made this “decision-making process” easier for us.

With explosion of Web 2.0 platforms consumers have a soapbox of unprecedented reach and power by which they can share opinions. Major companies have realized these consumer voices affect shaping voices of other consumers. Sentiment Analysis thus finds its use in Consumer Market for Product reviews, marketing for knowing consumer attitudes and trends, Social Media for finding general opinion about recent

hot topics in town, Movie to find whether a recently released movie is a hit.

Twitter is a social networking and micro blogging service that lets its users post real time messages, called tweets. Tweets have many unique characteristics, which implicates new challenges and shape up the means of carrying sentiment analysis on it as compared to other domains. Tweets are short messages, restricted to 140 characters in length. Due to the nature of this micro blogging service (quick and short messages), people use acronyms, make spelling mistakes, use emoticons and other characters that express special meanings.

1.3. Objective

Twitter is an effective tool for company to get people excited about its product. User can provide word-of-mouth marketing for companies by discussing their products. So, the sentiment analysis of tweets or most trendy news on twitter can be very productive for industry and the individuals. In this project, the main focus is on opinion expressions that convey people's positive or negative sentiments by analyzing the data sets collected from the Web such as twitter, movie review sites (IMDb), etc.

2. System Analysis

2.1. Existing system

Sentimental analysis is a grooming field and there are a lot of approaches that are followed in the field of sentimental analysis. Some of the early and recent results on sentiment analysis of Twitter data are by Go et al. (2009), (Bermingham and Smeaton, 2010) and Pak and Paroubek (2010). Go et al. (2009) use distant learning to acquire sentiment Data. They use tweets ending in positive emoticons like ":)" ":-)" as positive and negative emoticons like ":(" ":-(" as negative. Moreover, the data they use for training and testing is collected by search queries and is therefore biased.

In contrast, Our data is a random sample of streaming tweets unlike data collected by using specific queries. The size of our hand-labeled data allows us to perform crossvalidation experiments and check for the variance in performance of the classifier across folds.

Another significant effort for sentiment classification on Twitter data is by Barbosa and Feng (2010). They use polarity predictions from three websites as noisy labels to train a model and use 1000 manually labeled tweets for tuning and another 1000 manually labeled tweets for testing. They however do not mention how they collect their test data. They propose the use of syntax features of tweets like retweet, hashtags, link, punctuation and exclamation marks in conjunction with features like prior polarity of words and POS of words. We extend their approach by using real valued prior polarity, and by combining prior polarity with POS.

2.2. Overview

This project focuses on analyzing the users sentiments based on the tweets. The datasets has been collected from the Twitter. Various sentiment classification techniques has been used in the system which comprises of machine learning technique in which computer algorithms are trained to act accordingly based on the various datasets that have collected in earlier phases.
Supervised learning is used in this context. In this major concern random forest algorithm is used which uses Bayes rule as major equation and naïve based classifier built model by fitting a distribution of number of occurrence of each feature in the document. A confusion matrix is typically used as a visualization tool to present the results attained by a learner. The columns of the matrix represent instances in a predicted class, and the rows represent instances in an actual class.

The R language has been used for various purposes like cleaning of data that is removing of various unknown tags from the datasets for the optimized output which is sole concern of the project.

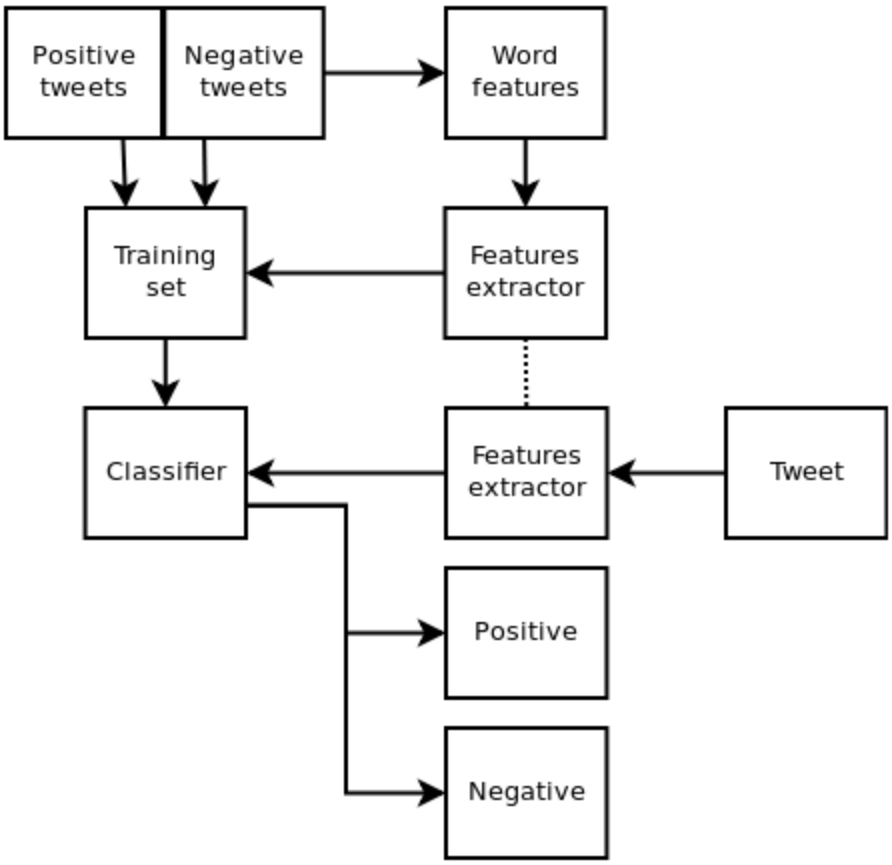

Fig: 2. 1: Overview of tweets classification and extraction

2.3. Motivation

Working with these informal text genres presents challenges for natural language processing beyond those typically encountered when working with more traditional text genres, such as newswire data. Tweets and texts are short: a sentence or a headline rather than a document. The language used is very informal, with creative spelling and punctuation, misspellings, slang, new words, URLs, and genre-specific terminology and abbreviations, such as, RT for "re-tweet" and # hashtags, which are a type of tagging for Twitter messages. How to handle such challenges so as to automatically mine and understand the opinions and sentiments that people are communicating has only very recently been the subject of research.

Another aspect of social media data such as Twitter messages is that it includes rich structured information about the individuals involved in the communication. For

example, Twitter maintains information of who follows whom and re-tweets and tags inside of tweets provide discourse information. Modeling such structured information is important because:

(i) it can lead to more accurate tools for extracting semantic information

(ii) because it provides means for empirically studying properties of social interactions

2.4. Proposed system

The system has designed to give the maximum efficiency on analyzing the tweets. In order to implement the system , a lot of sentiment analysis techniques investigated some of the that were successfully used in the past, for longer texts, and performed adjustments for making them suitable for micro blogging text.

Two classifiers that are used for this kind of system : a positive/negative classifier that deals with dual-sided sentiment of the tweet. For training these classifiers, we developed two corpora based on the ones used in other studies combined with data extracted from twitter.The two classifiers were the outcome of a series of experiments that includes exploring the performances of different supervised learning algorithms such as Naïve Bayes, Decision Tree and Random Forest Algorithm etc. when triggered with different parameters on various corpora, as well as dealing with different feature sets. The proposed techniques and methodologies are written in Python and R.

Additionally, since people are sometimes too busy and in need of a quick and easy way to understand the overall sentiment of the tweets, without having to look through the entire history of arguments and comments, three visualizations have been implemented. The visualizations make use of Chernoff Faces, stream graphs and line charts. Finally, the classifiers are evaluated and visualizations using tweets for another set of keywords.

2.5. Hardware and software requirement

2.5.1. Hardware requirement

- Operating system –Unix, Linux and Windows.
- RAM -2 GB
- Hard Disk -1 GB(Free Space)
- Processor -64bit (Preferred)

2.5.2. Software requirement

- TweepyAPI (python)
- R Studio
- R Version 3

3. Design

3.1. Model Architecture

The module that has presented is completely built on Machine Learning Architecture. This approach needs a data set, a classifier to train. Basic idea behind this approach is that first we collect the dataset which is movie review tweets from the Twitter. This data set is freely available on the internet. Then we pre-process the data set and prepare a training set for our classifier, after this training we provide data set to the classifier and a proper comparison would made in order to derive the expected result.

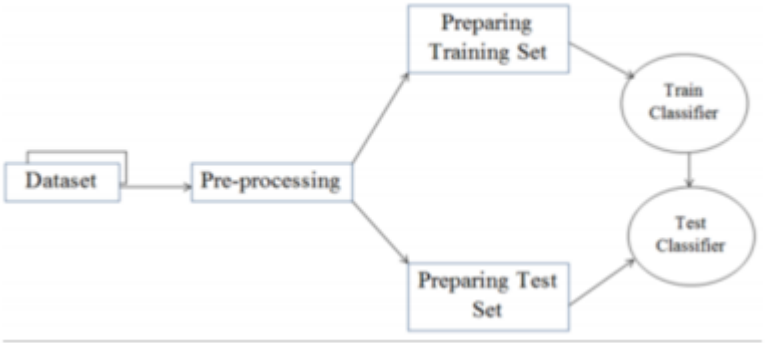

Fig: 3. 2: Implementation architecture using Machine Learning

3.2. Method to evaluate Machine a Machine Classifier

An important aspect of constructing a classifier using different techniques is finding ways to evaluate each one of them. We usually evaluate the performance of a classifier relative to the classifications that would have been made by a human expert. The automatic classification is considered to be correct if it matches the stored value of the gold standard classifications. It is worth noting that the gold standard classifications were made by humans so it is possible that errors were made.

Nevertheless, the gold standard corpus is by definition "correct". A classifier is trained with training examples and tested on a new set of data, test examples.The reason for testing on new data set is that the model can memorise its inputs and would incorrectly give high classification rate. Training examples are usually further divided into training data and validation data. The validation data set is used for optimizing the parameters of classifiers by performing error analysis which will indicate how to adjust the feature set.

3.3. Data Matrix

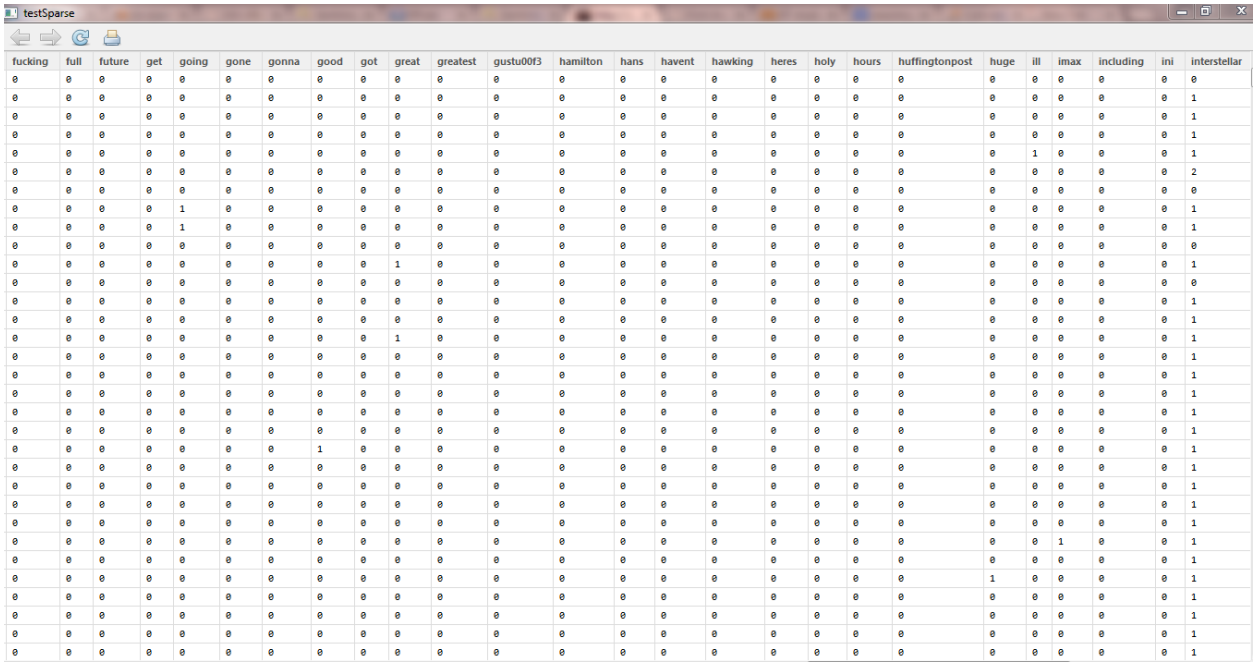

Fig: 3. 2: Data Matrix

3.4. Confusion Matrix

A confusion matrix is typically used as a visualization tool to present the results attained by a learner. The columns of the matrix represent instances in a predicted class, and the rows represent instances in an actual class.

	predictRF	
	FALSE	TRUE
FALSE	272	5
TRUE	30	8

Fig: 3.3: Confusion Matrix

3.5. Natural Language Processing Approach

Natural Language processing approach uses Corpus lexicon. This consists of positive, negative score for each of the term occurring in Corpus. Implementation is done by extracting the adjectives out of the sentence and then searching it in the corpus to find out its positive, negative score. In this way the total net score of the sentence is calculated and whichever is greater (either positive or negative) becomes the review for the sentence.

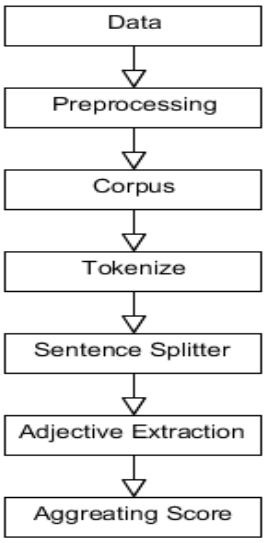

Fig: 3. 4: Implementation Architecture of NLP

3.6. The design model

To uncover the opinion direction, we will first extract the opinion words in the tweets and then find out their orientation, i.e., to decide whether each opinion word reflects a positive sentiment, negative sentiment or a neutral sentiment. In our work, we are considering the opinion words as the combination of the adjectives along with the verbs and adverbs. The corpus-based method is then used to find the semantic orientation of adjectives and the dictionary-based method is employed to find the semantic orientation of verbs and adverbs. The overall tweet sentiment is then calculated using a linear equation which incorporates emotion intensifiers too.

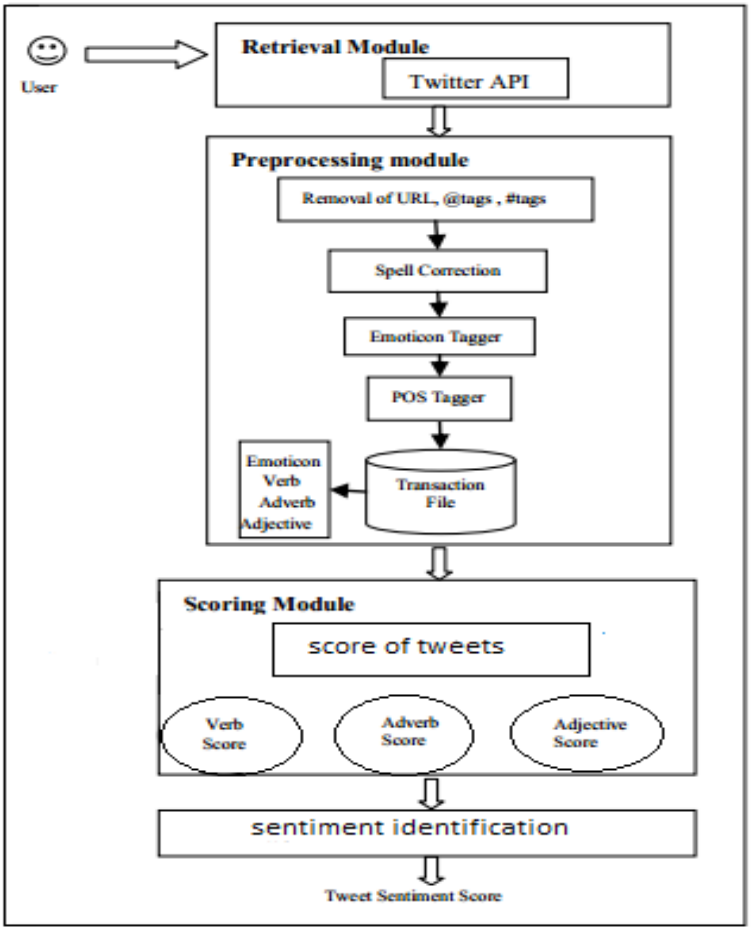

Fig: 3. 5: Detailed Architecture

3.7. Use cases for requirement analysis

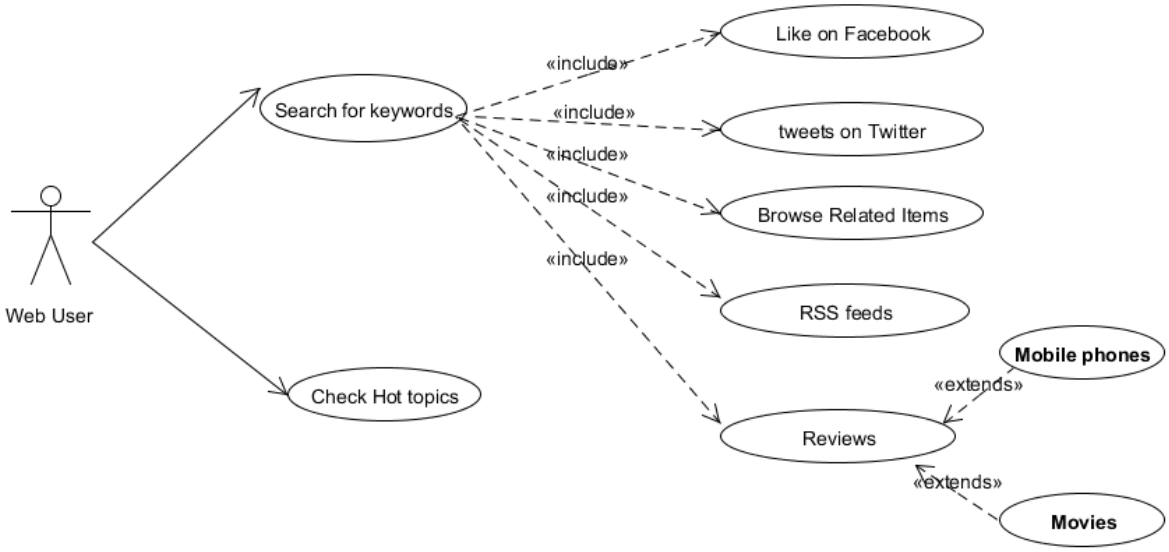

Fig: 3. 6: Use case for movie sentiment analysis

4. Implementation

4.1. Data description

Following are some key characteristics of tweets:

- Message Length: The maximum length of a Twitter message is 140 characters. This is different from previous sentiment classification research that focused on classifying longer texts, such as product and movie reviews.
- Availability: The amount of data available is immense. More people tweet in the public domain as compared to Facebook (as Facebook has many privacy settings) thus making data more readily available. The Twitter API facilitates collection of tweets for training.

- Topics: Twitter users post messages about a range of topics unlike other sites which are designed for a specific topic. This differs from a large fraction of past research, which focused on specific domains such as movie reviews.
- Real time: Blogs are updated at longer intervals of time as blogs characteristically are longer in nature and writing them takes time. Tweets on the other hand being limited to 140 letters and are updated very often. This gives a more real time feel and represents the first reactions to events.

1366 observations of 4 variables

	dates	tweets
1	1416371433	I was totally inter the stellar INTERSTELLAR despite the ending that was part CONTACT part ELYSI
2	1416371435	INTERSTELLAR WAS THE BEST MOVIE HOLY SHIT
3	1416371436	RT @EPcine: #AbsoluteZero El c\u00f3mic de Christopher Nolan que explica (algo) de #Interstellar h
4	1416371437	Just saw Interstellar and I can say that I did not sign a permission slip for that feel trip. Wel
5	1416371439	Ya me dieron ganas de ver Interstellar.
6	1416371446	Mind blown. #Interstellar
7	1416371447	I would pay to see interstellar again ugh that is such a good movie
8	1416371447	RT @Ohdamn_val: Interstellar is such a trip wtf
9	1416371448	Watching interstellar makes me rethink my life and all of my decisions. That movie hit me like a f
10	1416371449	RT @boxofficemojo: Monday November 17: 'Interstellar' - $2.15 million 'Dumber To' - $2.07 millio
11	1416371451	@KyzeaDC_ pinagiisipan ko pa pero baka gawin ko nalang interstellar or castle monarch king quen th
12	1416371451	I'd watch that Interstellar again. Anyone want to go see it let me know. I'll go with ya \ud83d\u
13	1416371452	Finally going to watch Interstellar. And I heard that they also filmed some scenes in Okotoks O.O
14	1416371453	Just saw Interstellar the story kept up pretty stronf great work indeed! \u2014 watching Interst
15	1416371456	interstellar is the most brilliant movie and if you haven't seen it yet just drop what your doing
16	1416371457	INTERSTELLAR\n13:15 16:30 19:45\nKOTA TUA JAKARTA\n12:30 16:50 21:10\nMANTAN TERINDAH\n14:40 19:00
17	1416371458	Interstellar got me like \ud83d\ude2f\ud83d\ude2f\ud83d\ude2f
18	1416371460	I conveniently bought the #Interstellar ost the day it got released without knowing. The organs in
19	1416371463	RT @DearCinema: How Christopher Nolan\u2019s Interstellar affirms Hollywood\u2019s faith in God ht
20	1416371464	Cuman kita doang?serasa milik sendiri ini bioskop bhakkk \u2605 Interstellar \u2014 https:\/\/t.co
21	1416371464	RT @cubosensei: Ayer que me v\u00ed otra vez Interstellar record\u00e9 a @MiNombreEsLucho y @aRedD
22	1416371466	RT @Interstellar: The @Interstellar soundtrack is here! Get it now on @iTunesMusic: http:\/\/t.co\
23	1416371470	Interstellar just blew my fucking mind.
24	1416371471	Interstellar was...phew...a lot to take in
25	1416371471	!! RT @doseofmySoul: interstellar was fucking spectacular

Displayed 1000 rows of 1366 (366 omitted)

Fig: 4.1: Data frame view of data sets

4.2. Resources and pre-processing of data

4.2.1. Pre-processing of Tweets

We prepare the transaction file that contains opinion indicators, namely the adjective, adverb and verb along with emoticons (we have taken a sample set of emoticons and manually assigned opinion strength to them). Also we identify some emotion intensifiers, namely, the percentage of the tweet in Caps, the length of repeated sequences & the number of exclamation marks, amongst others. Thus, we pre-process all the tweets as follows:

a) Remove all URLs (e.g. www.example.com), hash tags (e.g. #topic), targets (@username), and special Twitter words ("e.g. RT").

b) Calculate the percentage of the tweet in Caps.

c) Correct spellings; A sequence of repeated characters is tagged by a weight. We do this to differentiate between the regular usage and emphasized usage of a word.

d) Replace all the emoticons with their sentiment polarity.

e) Remove all punctuations after counting the number of exclamation marks.

```
> findFreqTerms(freq, lowfreq = 20)
 [1] "amaz"          "best"          "bikin"         "book"
 [5] "can"           "cerita"        "chapter"       "christoph"
 [9] "confus"        "degrass"       "end"           "entendist"
[13] "ever"          "film"          "free"          "fuck"
[17] "get"           "good"          "got"           "gustu00f3"
[21] "here"          "huffingtonpost" "imax"         "ini"
[25] "interstellar"  "ive"           "just"          "kip"
[29] "know"          "komik"         "like"          "lost"
[33] "love"          "mind"          "movi"          "need"
[37] "neil"          "neiltyson"     "new"           "nolan"
[41] "now"           "one"           "onlin"         "planet"
[45] "porqu"         "prekuel"       "que"           "realli"
[49] "rescu"         "rttowin"       "sallesino"     "saw"
[53] "scienc"        "see"           "seen"          "soundtrack"
[57] "still"         "think"         "thorn"         "time"
[61] "tyson"         "via"           "want"          "watch"
```

Fig: 4. 2: Frequent occurring words >20

4.3. Algorithm

4.3.1. Decision tree algorithm

Decision tree learning uses a decision tree as a predictive model which maps observations about an item to conclusions about the item's target value. It is one of the predictive modeling approaches used in statistics, data mining and machine learning. Tree models where the target variable can take a finite set of values are called **classification trees**. In these tree structures, leaves represent class labels and branches represent conjunctions of features that lead to those class labels. Decision trees where the target variable can take continuous values (typically real numbers) are called **regression trees**. In decision analysis, a decision tree can be used to visually and explicitly represent decisions and decision making. In data mining, a decision tree describes data but not decisions; rather the resulting classification tree can be an input for decision making. This page deals with decision trees in data mining.

In data mining, decision trees can be described also as the combination of mathematical and computational techniques to aid the description, categorization and generalization of a given set of data.

Data comes in records of the form:

$$(\mathbf{x}, Y) = (x_1, x_2, x_3, ..., x_k, Y)$$

The dependent variable, Y, is the target variable that we are trying to understand, classify or generalize. The vector **x** is composed of the input variables, x_1, x_2, x_3 etc., that are used for that task.

Decision trees used in data mining are of two main types:

- **Classification tree** analysis is when the predicted outcome is the class to which the data belongs.
- **Regression tree** analysis is when the predicted outcome can be considered a real number (e.g. the price of a house, or a patient's length of stay in a hospital).

Some techniques, often called *ensemble* methods, construct more than one decision tree:

- **Bagging** decision trees, an early ensemble method, builds multiple decision trees by repeatedly resembling training data with replacement, and voting the trees for a consensus prediction.

- **Random Forest** classifier uses a number of decision trees, in order to improve the classification rate.

- Boosted **Trees** can be used for regression-type and classification-type problems

Information gain is based on the concept of entropy from information theory.

$$I_E(f) = -\sum_{i=1}^{m} f_i \log_2 f_i$$

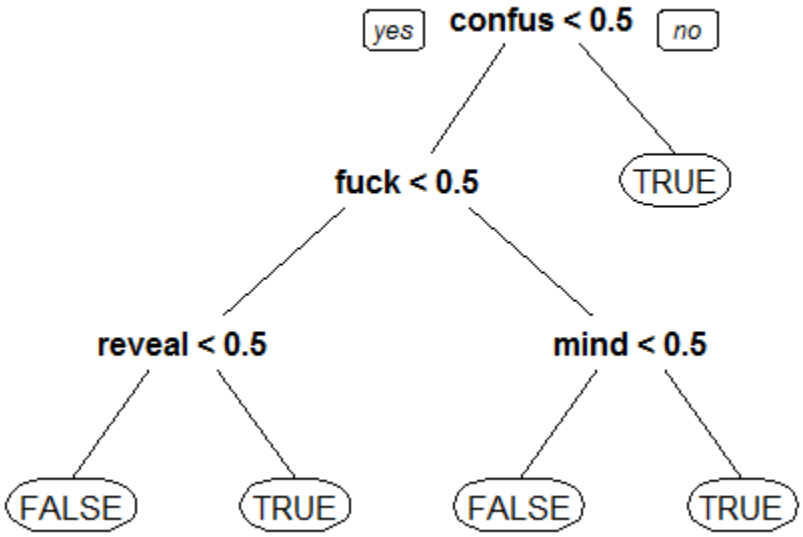

Fig: 4. 3: Decision Tree

4.3.2. Random forest algorithm

Random forests are an ensemble learning method for classification (and regression) that operate by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes output by individual trees. The algorithm for inducing a random forest was developed by Leo Breiman and Adele Cutler, and "Random Forests" is their trademark. The term came from **random decision forests** that was first proposed by Tin Kam Ho of Bell Labs in 1995. The

method combines Breiman's "bagging" idea and the random selection of features, introduced independently by Ho and Amit and Geman in order to construct a collection of decision trees with controlled variance.

We use 65% of our data set for training purpose which leaves approximately 890 reviews. To construct a decision tree each review must perform the following steps:

- Bootstrap sampling with replacement. In other words, sample from the set of 890 reviews,890 times. Duplicate are allowed because of replacement.
- Randomly select K of 14 features without replacement, and K different thresholds between the min and max values to split the samples.
- Select the split that has the lowest index. In other words, select the split that best separates the samples into the different classes. A review's class is its star rating (1, 2, 3, 4 or 5).
- Create a node in the decision tree.
- Repeat this process for each branch until the leaves at each node are of an acceptable purity. If the purity required is extremely high then you risk overfitting to the training data.

When classifying a piece of text, the random forest passes the item to each decision tree and the output is the majority vote. This may not be the optimal setup. For example, if we have 24 trees 10 of which are predicting 1 star, 7 are predicting 4 stars and 7 are predicting 5 stars. In this scenario, it may be better to select either 4 or 5 stars. It would be worthwhile to explore different setups.

```
> getTree(tweetRF, 1, labelVar=TRUE)
    left daughter right daughter           split var split point status prediction
1                2              3              ending         0.5      1       <NA>
2                4              5                mind         0.5      1       <NA>
3                0              0                <NA>         0.0     -1       TRUE
4                6              7               still         0.5      1       <NA>
5                8              9              fucked         0.5      1       <NA>
6               10             11                 que         0.5      1       <NA>
7               12             13               wanna         0.5      1       <NA>
8               14             15               still         0.5      1       <NA>
9                0              0                <NA>         0.0     -1       TRUE
10              16             17               brain         0.5      1       <NA>
11               0              0                <NA>         0.0     -1      FALSE
12              18             19                cant         0.5      1       <NA>
13               0              0                <NA>         0.0     -1      FALSE
14               0              0                <NA>         0.0     -1      FALSE
15               0              0                <NA>         0.0     -1       TRUE
16              20             21           mysteries         0.5      1       <NA>
17              22             23                like         0.5      1       <NA>
18               0              0                <NA>         0.0     -1      FALSE
19               0              0                <NA>         0.0     -1      FALSE
20              24             25             prekuel         0.5      1       <NA>
21              26             27                 can         0.5      1       <NA>
22               0              0                <NA>         0.0     -1       TRUE
23               0              0                <NA>         0.0     -1      FALSE
24              28             29                just         0.5      1       <NA>
25               0              0                <NA>         0.0     -1      FALSE
26              30             31             science         0.5      1       <NA>
27               0              0                <NA>         0.0     -1      FALSE
28              32             33                 see         0.5      1       <NA>
29               0              0                <NA>         0.0     -1      FALSE
30               0              0                <NA>         0.0     -1      FALSE
31               0              0                <NA>         0.0     -1       TRUE
32              34             35               think         0.5      1       <NA>
33              36             37                ever         0.5      1       <NA>
34              38             39             chapter         0.5      1       <NA>
35              40             41                 far         0.5      1       <NA>
36              42             43        interstellar         0.5      1       <NA>
```

Fig: 4. 3: Random Forest

4.4. Tools

Programming Language and Toolkits Used

The NLTK is platform for building python programs to work with text data. It provides a variety of corpora and resources and various libraries for text classification, tagging, stemming, tokenization and parsing.

In this project, NLTK was used extensively for tokenizing (tokenizing the tweets), POS tagging, the tagger model being maxent treebank pos tagger, stemming (as described above, it used the PorterStemmer of NLTK), and classification.The NLTK classifiers used were NaiveBayesClassifier and the MaxentClassifier.

4.5. Scoring of module

The next step is to find the semantic score of the opinion carriers i.e. the adjectives, verbs and adverbs. As mentioned previously, in our approach we use corpus based method to find the semantic orientation of adjectives and the dictionary-based method to find the semantic orientation of verbs and adverbs.

4.5.1. Score of adjectives

An adjective are a describing word and is used to qualify an object. The semantic orientation of adjectives tend to be domain specific, therefore we use a corpus based approach to quantify the semantic orientation of adjectives in the Twitter domain. Motivated by Hatzivassiloglou and McKeown , we ascribe same semantic orientation to conjoined adjectives in most cases and in special cases when the connective is “but”, the situation is reversed.

The seed lists of positive and negative adverbs and verbs whose orientation we know is created and then grown by searching in corpus. Based on intuition, we assign the strengths of a few frequently used adverbs and verbs with values ranging from -2 to +2. We consider some of the most frequently used adverbs and verbs along with their strength as given below.

Table 2: Verb and Adverb Strengths

Verb	*Strength*	*Adverb*	*Strength*
Love	1	complete	+1
adore	0.9	most	0.9
like	0.8	totally	0.8
enjoy	0.7	extremely	0.7
smile	0.6	too	0.6
impress	0.5	very	0.4
attract	0.4	pretty	0.3
excite	0.3	more	0.2
relax	0.2	much	0.1
reject	-0.2	any	-0.2
disgust	-0.3	quite	-0.3
suffer	-0.4	little	-0.4
dislike	-0.7	less	-0.6
detest	-0.8	not	-0.8
suck	-0.9	never	-0.9
hate	-1	hardly	-1

4.6. Result

4.6.1. Results with smaller dataset

The following are the results obtained by data processing, analysis and visualization on a smaller portion of the data, i.e. using 65% tweets for training and 35% tweets for testing.

4.6.1.1. Result Analysis using Naive Bayes:

The Baseline, i.e. Naive Bayes with Unigram gives more details on the most informative features after the classifier was run. The below table shows these details

```
> table(testSparse$Neg, predictCART)
       predictCART
        FALSE TRUE
  FALSE   275    2
  TRUE     30    8
>
> # Compute accuracy
> (272+10)/(272+5+28+10)
[1] 0.8952381
> # Baseline accuracy
> table(testSparse$Neg)

FALSE  TRUE
  277    38
```

Fig: 4. 4: training set accuracy

4.6.1.2. Effect of Stop words:

The algorithms were first run on the dataset without any data preprocessing. When Naive Bayes was run, it gave an accuracy of 89.25 percent, which is considered as the baseline result. The next thing used was stop word removal. When stop words were removed and random Forest was run, it gave an accuracy of 89.67 percent.

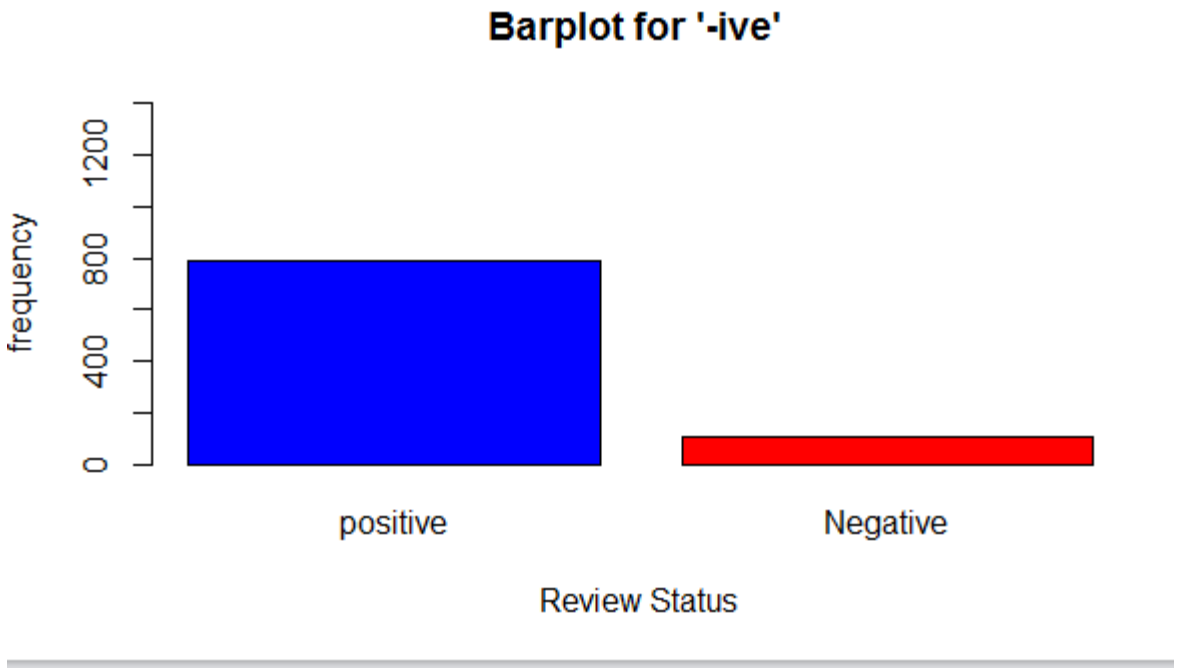

Fig: 4. 5: Bar Plot of Training data sentiment

4.6.2. Results with entire dataset

As the table shows, when the processing, analysis was done on the bigger dataset, the accuracy scaled up to a great extent. rpart baseline scaled up to 88.67 and Random forest up to 89.92 percent. The best result tested thus far, was obtained when random Forest was used on a feature set of a combination of Unigram, Bigram with stemming, giving an accuracy of 89.92.

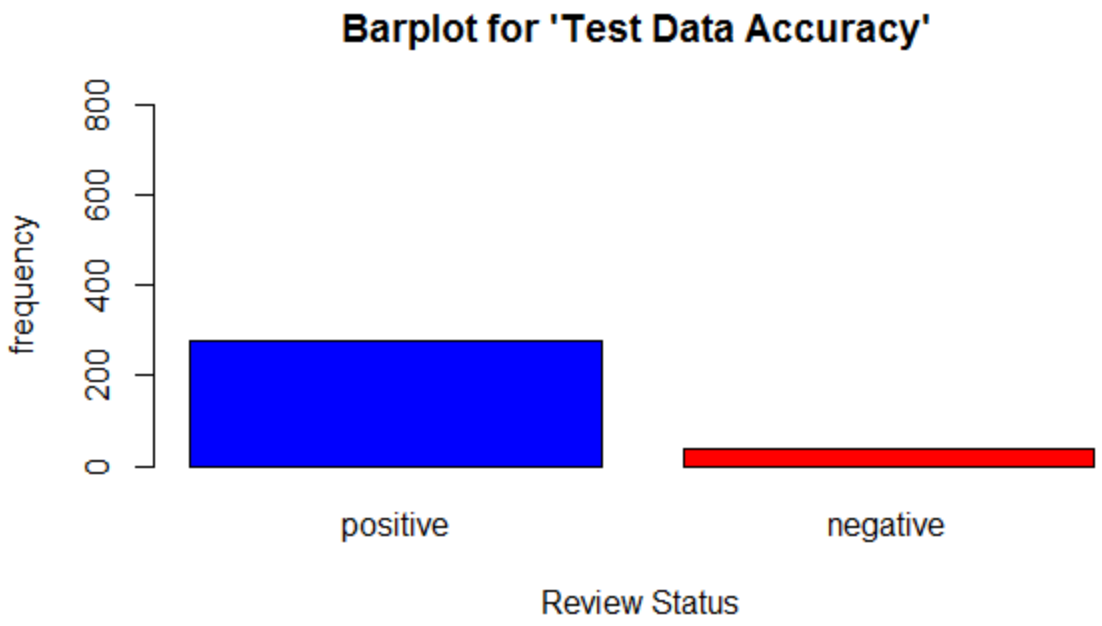

Fig: 4. 6: Bar plot of test data sentiment

4.5. Cross validation

It is technique of estimating the performance of a predictive model. It is some time called **rotation estimation.** It is a model validation technique for accessing how the result of a statistical analysis will generalize to an independent data set. It is mainly used in setting where the goal is prediction and one want to estimate how accurately the predictive model will perform in practice. In a predication problem, the model is usually given a dataset of known data on which the training is done and dataset of unknown data against which the model is tested. Cross validation is important in guarding against testing hypotheses suggested by the data, especially where the future sample are hazardous, costly and impossible to collect.

4.5.1 Rpart

```
> ctrl <- trainControl(method = "cv",repeats = 10)
> rpart.grid <- expand.grid(.cp=0.2)
> #Tuning parameter 'cp' was held constant at a value of 0.2
> (train.rpart <- train(Neg ~ ., data=trainSparse, method="rpart",trControl=ctrl,tuneGr
id = rpart.grid))
CART

892 samples
126 predictors
  2 classes: 'FALSE', 'TRUE'

No pre-processing
Resampling: Cross-Validated (10 fold)

Summary of sample sizes: 803, 802, 803, 803, 803, 803, ...

Resampling results

  Accuracy   Kappa      Accuracy SD  Kappa SD
  0.9193009  0.3827015  0.01260576   0.1221297

Tuning parameter 'cp' was held constant at a value of 0.2
```

Fig: 4. 7: 10 fold cross validation (rpart)

4.5.2 Random Forest

```
> (train.rpart <- train(Neg ~ ., data=trainSparse,trControl=ctrl))
Random Forest

892 samples
126 predictors
  2 classes: 'FALSE', 'TRUE'

No pre-processing
Resampling results across tuning parameters:

  mtry  Accuracy   Kappa
    2   0.9204036  0.3941377
   64   0.9035874  0.4302098
  126   0.8789238  0.3810013

Accuracy was used to select the optimal model using  the largest value.
The final value used for the model was mtry = 2.
```

Fig: 4. 8: 10 fold cross validation (Random Forest)

5. Limitations and Future Enhancements

5.1 Multi-class classification

Till now, I have only dealt with binary classification of tweets, either as positive or negative sentiment. There are many tweets, for instance, those with URL are which do not have any sentiment, or, are neutral. These tweets are mainly for sharing some useful information with people, and not necessarily for raising an opinion. As a part of my future work, I would like to explore multi-class classification into various levels of sentiment such as extremely positive, positive, neutral, negative and extremely negative.

5.2 More numeric feature

The numeric features that were used in this experiment include number of negative and positive words, emoticons, length of tweets and number of special characters such as exclamations, hash tags and so on. The numeric features did not yield good accuracy and gave around 83 percent accuracy. Hence, as a part of my future work on this, I would like to generate more as well as smarter numeric features.

5.3 Use more classifiers

In this project, rpart, Random Forest were used extensively. I would also like to explore other machine learning algorithms like Artificial Neural networks. Also generation of more numeric features will allow me to use more binary classifiers such as logistic regression and so on.

6. Output User Interface:

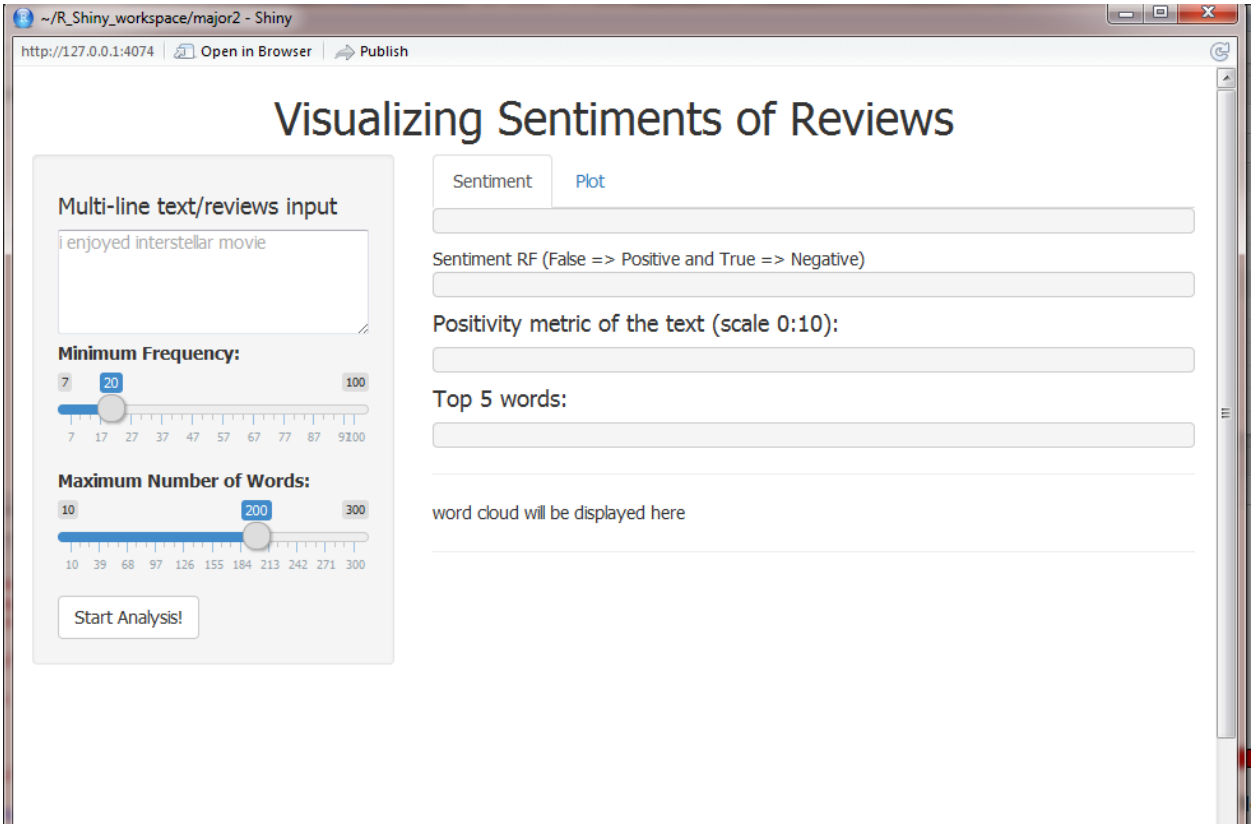

Fig: 5. 1: User Interface

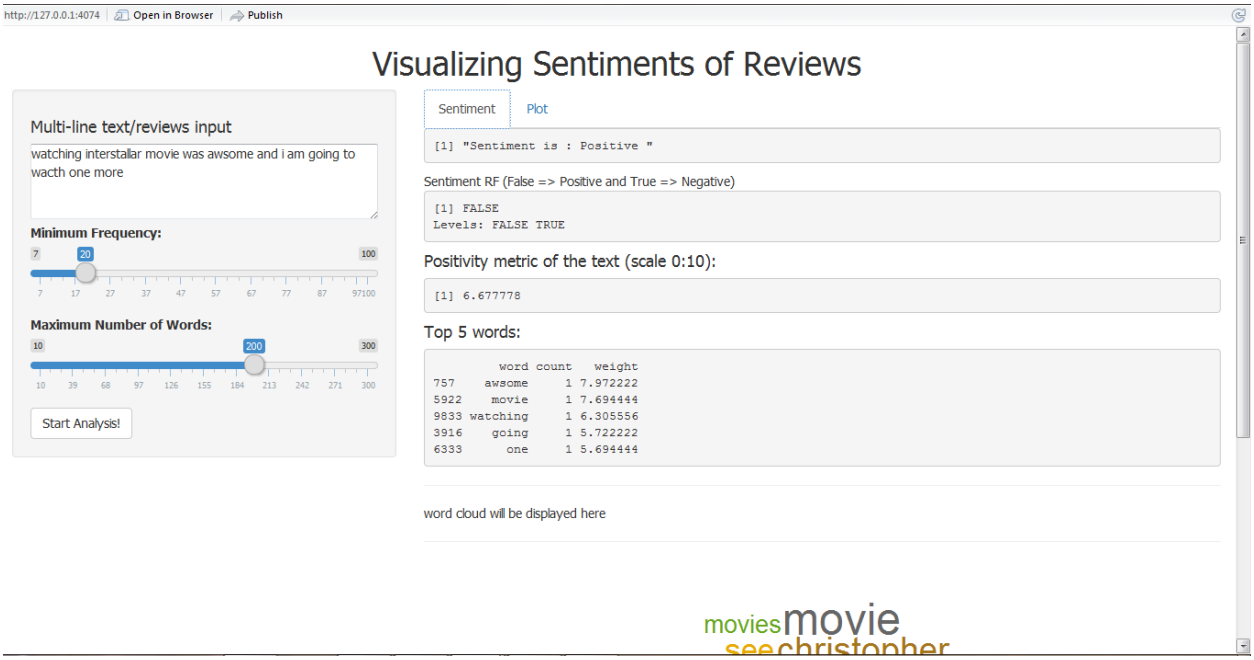

Fig: 5. 2: output screen

word cloud

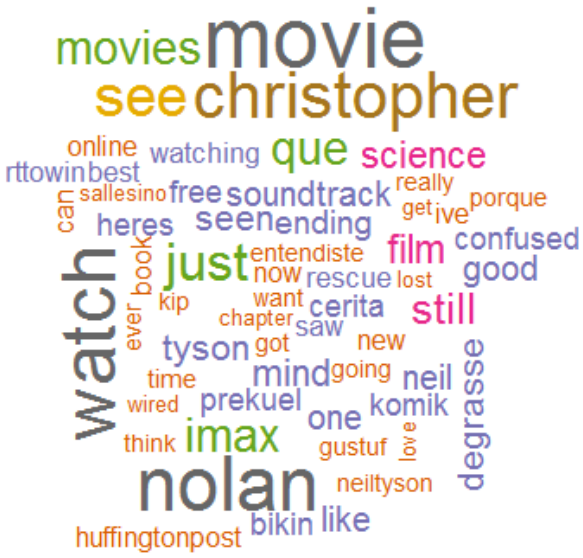

Fig: 5. 3: word cloud

7. Conclusion

The proliferation of micro blogging sites like Twitter offers an unprecedented opportunity to create and employ theories & technologies that search and mine for sentiments. We proposed the use of semantic features in Twitter sentiment classification and explored three different approaches for incorporating them into the analysis; with replacement, augmentation, and interpolation. We found that best results are achieved when interpolating the generative model of words given semantic concepts into the unigram language model of the NB classifier. We conducted extensive experiments on three Twitter datasets and compared the semantic features with the Unigrams. The overall tweet sentiment was then calculated using a linear equation which incorporated emotion intensifiers too. This work is exploratory in nature and the prototype evaluated is a preliminary prototype.

References

[1] Bo Pang and Lillian Lee**,** Opinion mining and sentiment analysis, Proceedings of the ACL, 2004

[2] Alec Go, Richa Bhayani and Lei Haung, Tweets classification using distant supervision, 2009

[3] G. Mishne. Experiments with mood classification in blog posts. In *1st Workshop on Stylistic Analysis Of Text For Information Access*, 2005.

[4] K. Nigam, J. La®erty, and A. Mccallum. Using maximum entropy for text classi¯cation. In *IJCAI-99 Workshop on Machine Learning for Information Filtering*, pages 61{67, 1999.

[5] B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classi¯cation using machine learning techniques. In *Proceedings of the Conference on Empirical Methods in Natural Language Processing* (*EMNLP)*, pages 79-86, 2002.