MAJOR PROJECT REPORT

ON

TWITTER SENTIMENT ANALYSIS

submitted by

Ameen Sidhique M Y (12170010)
Aushin Raj (12170022)
Midhun K V (12170042)
Muhammed Fabinsha (12170043)

In partial fulfilment of the requirements for the award of degree of Bachelor of Technology in Computer Science and Engineering.

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

APRIL 2020

DIVISION OF COMPUTER SCIENCE AND ENGINEERING
SCHOOL OF ENGINEERING
COCHIN UNIVERSITY OF SCIENCE AND TECHNOLOGY

CERTIFICATE

Certified that this is a bonafide record of the Major Project titled

TWITTER SENTIMENT ANALYSIS

done by
Ameen Sidhique M Y (12170010)
Aushin Raj (12170022)
Midhun K V (12170042)
Muhammed Fabinsha (12170043)

of VIII Semester, Computer Science and Engineering in the year 2019-2020 in partial fulfillment requirements for the award of Degree of Bachelor of Technology in Computer Science and Engineering of Cochin University of Science and Technology.

Dr. Latha R Nair	Dr.Sheena Mathew	Ms. Sheena S
Head of Division	Project Coordinator	Project Guide

Acknowledgement

We take this opportunity to thank the supreme being, the source of all knowledge whose blessings are our guiding light in any venture we take up. We're in short of words to express our gratitude to Mrs Sheena S, our project guide who guided us and helped us constantly with her inputs and suggestion without which we couldn't have implemented this project. We are highly indebted to her, for her constant supervision and support in completing this project. We'd like to thank Prof. Sheena Mathew for her valuable inputs, and we also express our heartfelt thanks to Ms Minu Poulose who was in charge of the project lab during the semester , for helping us by providing all the necessary amenities for completing the project. A bouquet of gratitude to Dr. Latha R Nair, Head of Division of Computer Science and Engineering for all kinds of encouragement extended to us.

Ameen Sidhique M Y (12170010)
Aushin Raj (12170022)
Midhun K V (12170042)
Muhammed Fabinsha (12170043)

Declaration

We, Muhammed Fabinsha, Aushin Raj, Midhun K V and Ameen Sidhique hereby declare that this project is the record of authentic work carried out by us during the academic year 2019 - 2020 and has not been submitted to any other University or Institute towards the award of any degree.

Abstract

Twitter is a popular social networking website where members create and interact with messages known as "tweets". This serves as a mean for individuals to express their thoughts or feelings about dierent subjects. Various different parties such as consumers and marketers have done sentiment analysis on such tweets to gather insights into products or to conduct market analysis. Furthermore, with the recent advancements in machine learning algorithms, we are able improve the accuracy of our sentiment analysis predictions. In this report, we will attempt to conduct sentiment analysis on "tweets" using various different machine learning algorithms. We attempt to classify the polarity of the tweet where it is either positive or negative. If the tweet has both positive and negative elements, the more dominant sentiment should be picked as the final label. We intend to use the dataset from Kaggle which was crawled and labeled positive/negative. The data provided comes with emoticons, usernames and hashtags which are required to be processed and converted into a standard form. We also need to extract useful features from the text such unigrams and bigrams which is a form of representation of the "tweet". We plan to use various machine learning algorithms to conduct sentiment analysis using the extracted features.

Contents

List of Figures

Chapter 1

Introduction

Sentiment analysis is the task of finding the opinions and affinity of people towards specific topics of interest. Be it a product or a movie, opinions of people matter, and it affects the decision-making process of people. The first thing a person does when he or she wants to buy a product online, is to see the kind of reviews and opinions that people have written. Social media such as Facebook, blogs, twitter have become a place where people post their opinions on certain topics. The sentiment of the tweets of a particular subject has multiple usage, including stock market analysis of a company, movie reviews, in psychology to analyze the mood of people that has a variety of applications, and so on.

This project of analyzing sentiments of tweets comes under the domain of "Pattern Classification" and "Data Mining". Both of these terms are very closely related and intertwined, and they can be formally defined as the process of discovering "useful" patterns in large set of data, either automatically (unsupervised) or semi-automatically (supervised). The project would heavily rely on techniques of "Natural Language Processing" in extracting significant patterns and features from the large data set of tweets and on "Machine Learning" techniques for accurately classifying individual unlabelled data samples (tweets) according to whichever pattern model best describes them

The features that can be used for modeling patterns and classification can be divided into two main groups: formal language based and informal blogging based. Language based features are those that deal with formal linguistics and include prior sentiment polarity of individidual words and phrases, and parts of speech tagging of the sentence. Prior sentiment polarity means that some words and phrases have a natural innate tendency for expressing particular and specific sentiments in general. For example the word "excellent" has a strong positive connotation while the word "evil" possesses a strong negative connotation. So whenever a word with positive connotation is used in a sentence, chances are that the entire sentence would be expressing a positive sentiment. Parts of Speech tagging, on the other hand, is a syntactical approach to the problem. It means to automatically identify which part of speech each individual word of a sentence belongs to: noun, pronoun, adverb, adjective, verb, interjection, etc. Patterns can be extracted from analyzing the frequency distribution of these parts of speech (ether individually or collectively with some other part of speech) in a particular class of labeled tweets. Twitter based features are more informal and relate with how people express themselves on online social platforms and compress their sentiments in the limited space of 140 characters offered by twitter. They include twitter hashtags, retweets, word capitalization, word lengthening , question marks, presence of url in tweets, exclamation marks, internet emoticons and internet shorthand/slangs. The data, being labeled by humans,has a lot of noise, and its hard to achieve good accuracy.

The four main important topics include data pre-processing, the machine learning algorithms used, the tools required to execute the project as well as the feature extraction techniques along with features to be used.

Chapter 2

Literature Review

With the popularity of blogs and social networks, opinion mining and sentiment analysis became a field of interest in the scientific community. A very broad overview of the existing work was presented in (Pang and Lee, 2008)[1]. In their survey, the authors describe existing techniques and approaches for an opinion-oriented information retrieval. In Yang et al. (2007)[2], the authors use web-blogs to construct a corpora for sentiment analysis and use emoticons assigned to blog posts as indicators of users' mood. The authors applied SVM and CRF learners to classify sentiments at the sentence level and then investigated several strategies to determine the overall sentiment of the document.

Some of the early and recent results on sentiment analysis of Twitter data are by Go et al. (2009)[3]. Go et al. use distant learning to acquire sentiment data. They use tweets ending in positive emoticons like ":)" ":-)" as positive and negative emoticons like " :(" ":-(" as negative. They build models using Naive Bayes, MaxEnt and Support Vector Machines (SVM), and they report that SVM outperforms other classifiers. (Agarwal et al., 2011) performed three class (positive, negative and neutral) classification of tweets [4] On the features, they have used Unigram,Bigram, along with Part-of-speech (POS) tagging. They note that unigram feature outperforms all other models and also mention that bigrams and POS tagging does not help.They also perform some pre-processing of the data that was used in modeling the pre-processing techniques that we intend to use in this project. The text processing they perform includes removal of URLs, username references and repeated characters in words.They collected their dataset using Twitter stream API and asked human judges to annotate the data into three classes. They had 1709 tweets of each class making a total of 5127 in all. In their research, they introduced POS-specific prior polarity features along with twitter specific features.

Researchers have also begun to investigate various ways of automatically collecting training data. Several researchers rely on emoticons for defining their training data (Pak and Paroubek 2010) [5]. (Barbosa and Feng 2010) exploit existing Twitter sentiment sites for collecting training data[6]. (Davidov, Tsur, and Rappoport 2010) also use hashtags for creating training data, but they limit their experiments to sentiment/non-sentiment classification, rather than 3-way polarity classification.[7]

As the twitter data is noisy with lot of slang and short words some of the pre-processing techniques using the slang dictionary, along with it removal of the stop words. They also use the emoticon dictionary which we plan to implement in this project to be used in numeric features. They also implement a prior polarity scoring which scores many English words between 1(Negative) to 3(Positive). From the algorithm point of view they provide a tree kernel and feature based models. Un-igram baseline model is combined with other features and modeled in this paper. Different combination of the features are selected. Part-of-speech tagging and emoticons list from wikipedia are used for the features.

Chapter 3

System Study

In this section, we are going to present the SRS, system objectives and hardware and software tool requirements.

3.1 Software Requirements Specification

A software requirements specification (SRS) is a document that captures complete description about how the system is expected to perform.

Purpose

This Software Requirements Specification (SRS) documents key specifications, describes a prototype in terms of functional and non-functional requirements for Twitter Sentiment Analysis.

In this report, we will attempt to conduct sentiment analysis on "tweets using various different machine learning algorithms. We attempt to classify the polarity of the tweet where it is either positive or negative. If the tweet has both positive and negative elements, the more dominant sentiment should be picked as the final label.

We intend to use the dataset from Kaggle which was crawled as labeled positive and negative. The data provided comes with emoticons, usernames and hashtags which are required to be processed and converted into a standard form. We also need to extract useful features from the text such unigrams and bigrams which is a form of representation of the “tweet”.

We intend to use various machine learning algorithms to conduct sentiment analysis using the extracted features.

The purpose of this document is to lay out the functional and non-functional requirements for the application. It also provides a detailed overview of our product, its parameters and goals. This document describes the project’s target audience, its user interface, its software and hardware requirements.

Project Overview

The project has following functionalities:

Functional Requirements

- Pre-processing :

  Raw tweets scraped from twitter generally result in a noisy dataset. This is due to the casual nature of people’s usage of social media. Tweets have certain special characteristics such as re- tweets, emoticons, user mentions, etc. which have to be suitably extracted. Therefore, raw twitter data has to be normalized to create a dataset which can be easily learned by various classifiers. We have applied an extensive number of pre-processing steps to standardize the dataset and reduce its size.

  Input: Raw tweet
  Output : Pre-processed and normalized data

- Feature Extraction :

  We extract two types of features from our dataset, namely unigrams and bigrams. We create a frequency distribution of the unigrams and bigrams present in the dataset and choose top N unigrams and bigrams for our analysis.

Input : Pre-processed and normalized data
Output: Unigrams and Bigrams

- Feature Representation

  After extracting the unigrams and bigrams, we represent each tweet as a feature vector in either sparse vector representation or dense vector representation depending on the classification method.

  Input: Unigrams and Bigrams
  Output: Feature vector

- Classifiers :

  The feature vector is fed into different machine learning algorithms to calculate the sentiment of the tweet

  Input: Feature vector
  Output: Sentiment Score

Non Functional Requirements

1. Performance Requirements

   The optimal algorithm should have an accuracy of more than 80%.

   As for this project we will keep on detecting if the system crashed, hanged or an operating system error occurred. Also detecting the performance of the system in terms of the efficiency of integration of the different components

2. Safety Requirements

The system must not be damaged or manipulated by unauthorized access to the database.

3.2 Hardware and Software Requirements

Hardware Interface

- Core i5 or higher
- RAM: 16 GB

Software Requirements

- Python
- NumPy
- scikit-learn
- scipy
- nltk
- IPython
- Pandas

Chapter 4

System Design

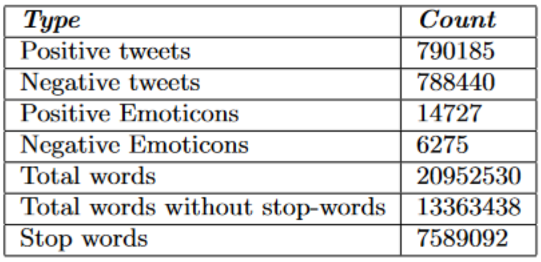
4.1 Introduction

The main approach involved in this project are the various data pre-processing steps, the machine learning classifiers and feature extraction. The main machine learning algorithms we intend to use are Naive Bayes, Maximum Entropy(MaxEnt) and XGBoost. The main data pre-processing steps include URL and username filtering, twitter slang removal, stop words removal and stemming. Feature extraction includes POS tagging, unigram, bigram (all the above in various combinations) and numeric features all of which are described below.

4.2 Modular Design

Data

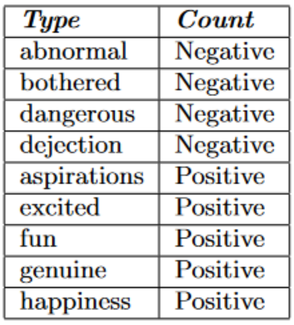
Tweets are short length messages and have a maximum length of 140 characters. This limits the amount of information that the user can share with every message. Due to this reason, users use a lot of acronyms, hashtags, emoticons, slang and special characters. Acronyms and slang such as 2moro for tomorrow and so on are used to keep sentences within the word limit. People also refer to other users using the @ operator. Users also post URLs of webpages to share information. Emoticons are a great way to express emotions without having to say much. More details on these are explained in the next section. The data to be used for this project is based out of Sentiment140 and contains about 1.5 million classified tweets, each row is marked as 1 for positive sentiment and 0 for negative sentiment. More details about the data are as shown in Figure 4.1.



Type	*Count*
Positive tweets	790185
Negative tweets	788440
Positive Emoticons	14727
Negative Emoticons	6275
Total words	20952530
Total words without stop-words	13363438
Stop words	7589092

Figure 4.1: Data Statistics

Along with the twitter data, the project also required other datasets like stopwords , a dictionary of negative and positive words , an emoticon dictionary and an acronym dictionary for twitter slang words . The use of these are described in the next section. Dictionary of negative and positive words. The dictionary of negative and positive words is a dataset containing around 6800 negative and positive words. This dataset is used to determine the numeric features of number of negative and positive words in the tweets, based on which sentiment classification is done. The process of stemming, as explained below is also performed on this dataset, so that it maps to the training and test dataset. Some negative and positive words from the dataset are shown in Figure 4.2 :

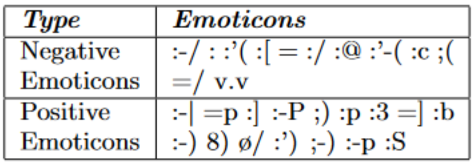


Type	*Count*
abnormal	Negative
bothered	Negative
dangerous	Negative
dejection	Negative
aspirations	Positive
excited	Positive
fun	Positive
genuine	Positive
happiness	Positive

Figure 4.2: Negative and positive words dataset

Emoticons

Emoticons are a great way to express emotions, especially given the restriction on the length of tweets. Emoticons also form an effective way in determining the sentiment of the tweet. In this project, the emoticons are used as numeric features - positive and negative emoticons. Some of the positive and negative emoticons are shown in Figure 4.3.

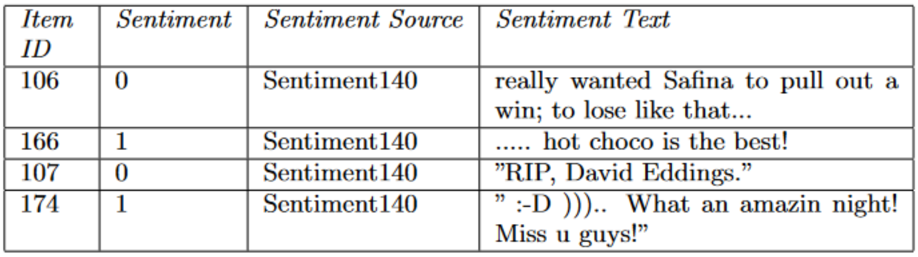


Type	***Emoticons***
Negative Emoticons	:-/ : :'(:[= :/ :@ :'-(:c ;(=/ v.v
Positive Emoticons	:-| =p :] :-P ;) :p :3 =] :b :-) 8) ø/ :') ;-) :-p :S

Figure 4.3: Negative and positive emoticons

Data Pre-processing

- Filtering



Item ID	*Sentiment*	*Sentiment Source*	*Sentiment Text*
106	0	Sentiment140	really wanted Safina to pull out a win; to lose like that...
166	1	Sentiment140	..... hot choco is the best!
107	0	Sentiment140	"RIP, David Eddings."
174	1	Sentiment140	" :-D))).. What an amazin night! Miss u guys!"

Figure 4.4: Unfiltered Data Examples

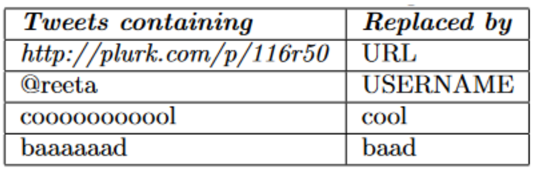
  – URLs

    People use twitter not only for expressing their opinions but also for sharing information with others. Given the short maximum length of tweets, one way of sharing is using links.Tweets include various links or URLs and these do not con-tribute to the sentiment of the tweet. The URLs in the data used in this project are of the form http://plurk.com/p/116r50.These do not contribute to the sentiment of the tweet. Hence these are to be parsed and replaced by a common word, URL.

- Usernames

  Tweets often refer to other users and such references begin with the @ symbol. These again do not contribute to the sentiment and hence are to be replaced by the generic word USERNAME
- Duplicates or repeated characters People use a lot of casual language on twitter. For example, 'happy' is used in the form of 'haaaaaaappy'. Though this implies the same word 'happy', the classifiers consider these as two different words.

  To improve this and make words more similar to generic words, such sets of repeated letters are to be replaced by two occurrences. Thus haaaaappy would be replaced by haappy.Some examples of the same are shown in Figure 4.5

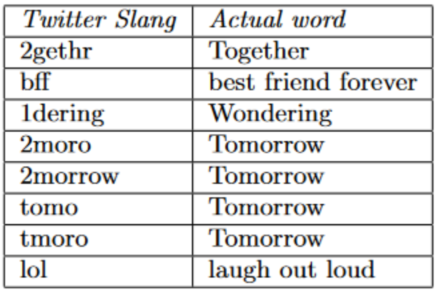


Tweets containing	*Replaced by*
http://plurk.com/p/116r50	URL
@reeta	USERNAME
cooooooooool	cool
baaaaaad	baad

Figure 4.5: Data Filtering

- Twitter slang removal

  As mentioned in the previous statement, tweets contain a lot of casual language. Also, given that the maximum length of a tweet is 140 characters, people tend to use abbreviations or some short forms for words. These short words are to be replaced by the actual words that they represent to improve performance of the learning algorithms. The advantage of doing this is evident from the table. The word Tomorrow is used by people using many short forms like 2moro, 2morrow, tomo, tmoro and so on. If these are not mapped to the common original word, then training on them would not produce good accuracy and may also cause overfitting, as these might not be found in the test Data. Some examples are shown in Figure 4.6.



Twitter Slang	*Actual word*
2gethr	Together
bff	best friend forever
1dering	Wondering
2moro	Tomorrow
2morrow	Tomorrow
tomo	Tomorrow
tmoro	Tomorrow
lol	laugh out loud

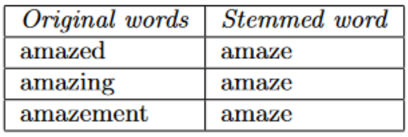
Figure 4.6: Twitter Slang

- Stop-words removal

  In information retrieval, there exists many words that are added as conjunctions in sentences. For example, words like the, and, before, while, and so on do not contribute to the sentiment of the tweet. Also these words do not help in classifying the tweets as they appear in all classes of tweets. These words are to be removed from the data so as to avoid using them as features. The stop words corpus is to be obtained from NLTK. Some modifications are required to this as the corpus will also have some negative words such as nor, not, neither which are important in identifying negative sentiments and should not be removed.

- Stemming

  In information retrieval, stemming is the process of reducing a word to its root form. For example, walking, walker, walked all these words are derived from the root word walk. Hence, the stemmed form of all the above words is walk. NLTK provides various packages for stemming such as the PorterStemmer, LancasterStemmer and so on. The PorterStemmer is to be used in this project which uses various rules for sux stripping. In addition to stemming the train and test data, the positive and negative word corpus should be also stemmed. Stemming reduces the feature space as many derived words are reduced to the same root form. Multiple features now point to the same word and hence it increases the probability of the word. Some examples of stemming is shown in Figure 4.7.



Original words	*Stemmed word*
amazed	amaze
amazing	amaze
amazement	amaze

Figure 4.7: Stemming

Feature Extraction

- Unigram

  Unigrams are the simplest features that can be used for learning tweets. The bag-of-words model is a powerful technique in sentiment analysis. This technique involves collecting all words in the document and using them as features.The features can either be the frequency of words, or simply 0s and 1s to indicate if the word is present in the documentor not. In this project, 0s and 1s are to be used to indicate the absence or presence of a word in the tweet.

- Bigram

  Bigrams are features consisting of sets of two adjacent words in a sentence. Unigram sometimes cannot capture phrases and multi-word expressions, effectively disregarding any word order dependence. For example, words like 'not happy', 'not good' clearly say that the sentiment is negative,but a unigram might fail to identify this. In such cases, bi-grams help in recognizing the correct sentiment of the tweet

- POS Tagging

  Part-of-speech tagging in linguistics and information retrieval is the process of tagging each word in a sentence to a particular part of speech. There are many parts of speech such as noun, adjective, pronoun, preposition, adverb, and so on. A word can take different meanings in different sentences, i.e a word can act as a noun in one sentence, and as an adjective in another.For this project, the tagger modelmaxenttreebankpostagger provided by NLTK is intended to be used. Figure 4.8 shows a POS-tagging.

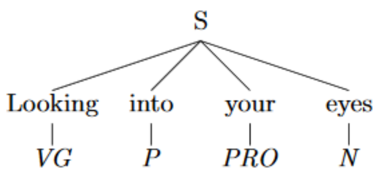


Figure 4.8: NLTK POS tagging

Machine Learning Algorithms To Be Implemented

- Baseline

  This experiment uses Naive Bayes with Unigrams as a baseline.

- Naive Bayes

  The Naive Bayes classifier is one of the basic text classification algorithms. It is a simple classifier based on Bayes theorem and makes naive independence assumptions of the feature variables. Despite this very naive assumption, it is seen to perform very well in many real-world problems. Mathematical representation: Consider attributes X_1 , X_2 ....X $_n$ to be conditionally independent of each other given a class Y. This assumption gives us,

$$P(X_1....X_n|Y) = \prod_{i=1}^{n} P(X_i|Y)$$

By Bayes theorem, we have,

$$P(Y|X_i) = \frac{P(X_i|Y)P(Y)}{P(X_i)}$$

Using Bayes theorem in the previous equation, we can find the probability of predicting the class Y given the features X_i . The class that gives the maximum probability that the given features predict it, is the class that the tweet will belong to. In this experiment, the NaiveBayesClassifier from NLTK is intended to be used to train and test the data.

- MaxEnt

  The Max Entropy classifier is a discriminative classifier commonly used in Natural Language Processing, Speech and Information Retrieval problems. The max entropy classifier uses a model very similar to the Naive bayes model but it does not make any independence assumption, unlike Naive Bayes. The max Ent classifier is based on the principle of maximum entropy and from all the models, chooses the once which has the maximum entropy. The goal is to classify the text(tweet, document, reviews) to a particular class, given unigrams, bigrams or others as features. If w_1, w_2,...,w_m are the words that can appear in a document, according to bag- of-words model, each document can be represented by 1s and 0s indicating if the word w_i is present in the document or not. The parametric form of the MaxEnt model can be represented as below:

$$P(c|d, \lambda) = \frac{exp[\sum_i \lambda_i f_i(c,d)]}{\sum_c [\sum_i \lambda_i f_i(c,d)]}$$

  Here, c is the class to be predicted, d is the tweet, and is the weight vector. The weight vector defines the importance of a feature. Higher weight means that the feature is a strong indicator for the class c. The parameters are chosen by iterative optimization, and for the same reason, this classifier takes a long time to learn when training size, features are large.

- XGBoost

  Xgboost is a form of gradient boosting algorithm which produces a prediction model that is an ensemble of weak prediction decision trees. We use the ensemble of K models by adding their outputs in the following manner (where F is the space of trees, Xi is the input and ˆYi is the final output.)

$$\hat{y}_i = \sum_{k=1}^{K} f_k(x_i), f_k \in F$$

4.3 Data Flow Diagrams

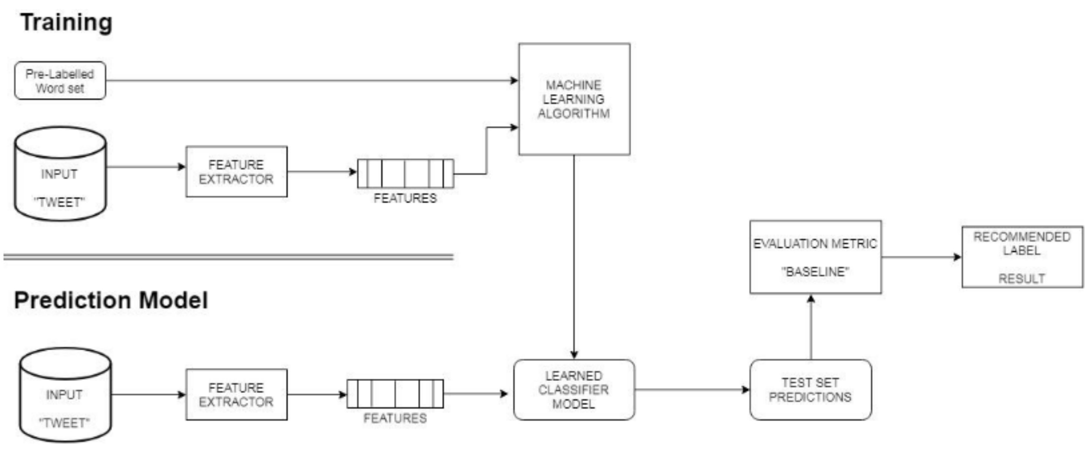


Figure 4.9: Machine Learning Model for Twitter System Analysis System

Chapter 5

System Implementation

5.1 Platform and Tools

IPython

Ipython is a command shell for interactive computing mainly for Python. Few of its main features are its input history across sessions, tab completion, support for visualization and use of GUI tool kits. The IPython also oers a rich text web interface called the IPython notebook. This project used IPython and IPython notebook extensively for data processing, learning, analysis and visualization, the results of which are discussed in the next section.

Natural Language Toolkit

The NLTK is platform for building python programs to work with text data. It provides a variety of corpora and re- sources 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 Porter-Stemmer of NLTK), and classification.The NLTK classifiers used were NaiveBayesClassifier andthe MaxentClassifier

Pandas

Pandas is a software library written for Python and is used for data analysis and manipulation. Pandas is an open source library and it also interoperates with the IPython and other Python libraries.

Sci-kit Learn

Scikit-Learn is an open source machine learning library for the Python programming language. It features various classification, regression and clustering algorithms including support vector machines, logistic regression, naive Bayes, random forests, gradient boosting and k-means, and is de- signed to interoperate with the Python numerical and sci- entific libraries NumPy and SciPy. In this project, Sci-kit Learn was mainly used for the SVM classifier, in particular, the Linear SVC.

Chapter 6

Experiments

6.1 Baseline

This experiment uses Naive Bayes with Unigrams as a baseline. Using this baseline model, we achieve a classification accuracy of 77.64% on Kaggle public leaderboard.

6.2 Naive Bayes

We used MultinomialNB from sklearn.naive_bayes package of scikit-learn for Naive Bayes clas sification. We used Laplace smoothed version of Naive Bayes with the smoothing parameter set to its default value of 1. We used sparse vector representation for classification and ran experiments using both presence and frequency feature types. We found that presence features outperform fre-quency features because Naive Bayes is essentially built to work better on integer features ratherthan oats. We also observed that addition of bigram features improves the accuracy. We obtain a best validation accuracy of 81.794% using Naive Bayes with presence of unigrams and bigrams.

6.3 Maximum Entropy

The nltk library provides several text analysis tools. We use the MaxentClassifier to perform sentiment analysis on the given tweets. Unigrams, bigrams and a combination of both were given as input features to the classifier. The Improved Iterative Scaling algorithm for training provided better results than Generalised Iterative Scaling.

6.4 XGBoost

We also attempted tackling the problem with XGboost classifier. We set max tree depth to 25 where it refers to the maximum depth of a tree and is used to control over-fitting as a high value might result in the model learning relations that are tied to the training data. Since XGboost is an algorithm that utilises an ensemble of weaker trees, it is important to tune the number of estimators that is used.

Chapter 7

Results

7.1 Features

We have used

- Unigrams
- Bigrams
- Unigrams + Bigrams

as features. We have used three model with above mentioned features. Note that all the results shown here are of test results which is obtained by submitting the output on the test file to kaggle.

For all of the classifiers shown above we can see that only using unigrams gives the least accuracy where as maximum accuracy is achieved by using Maximum entropy classifier using uni+bigrams as features.

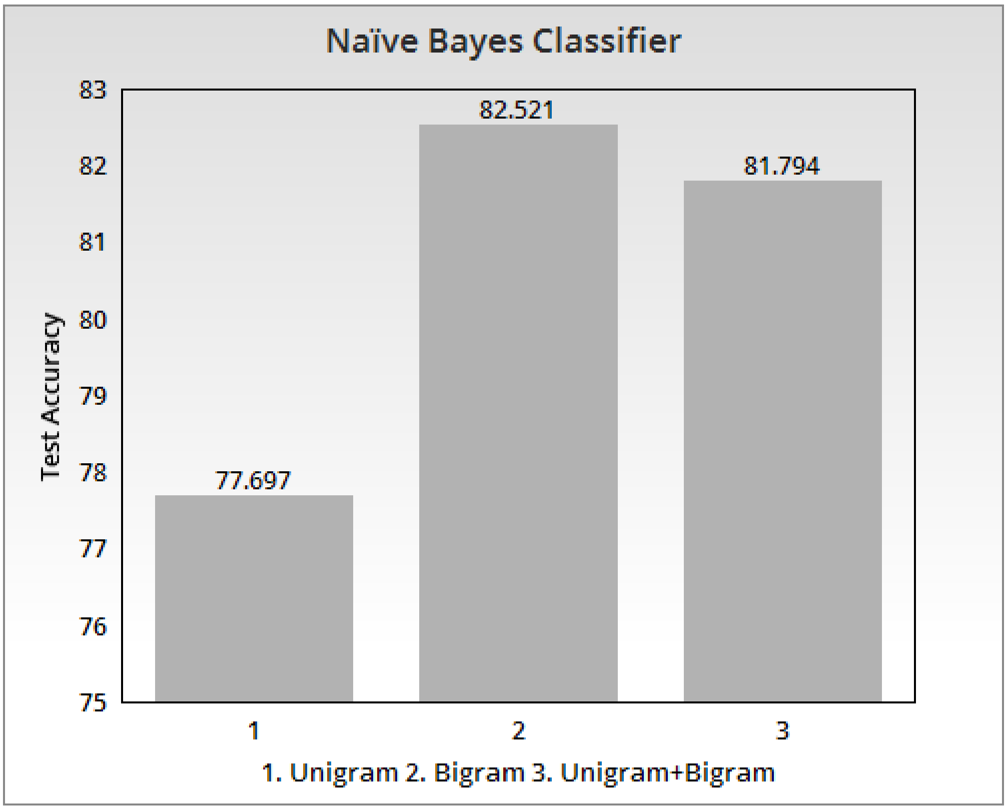


Figure 7.1: Naïve Bayes Results

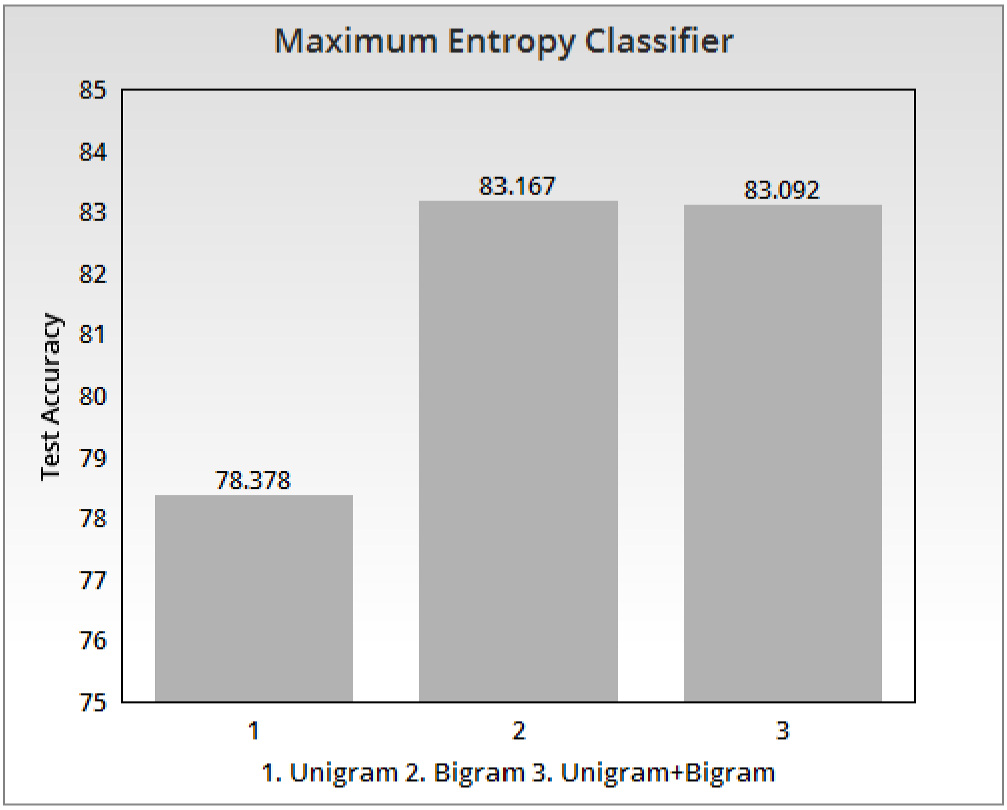


Figure 7.2: Maximum Entropy Results

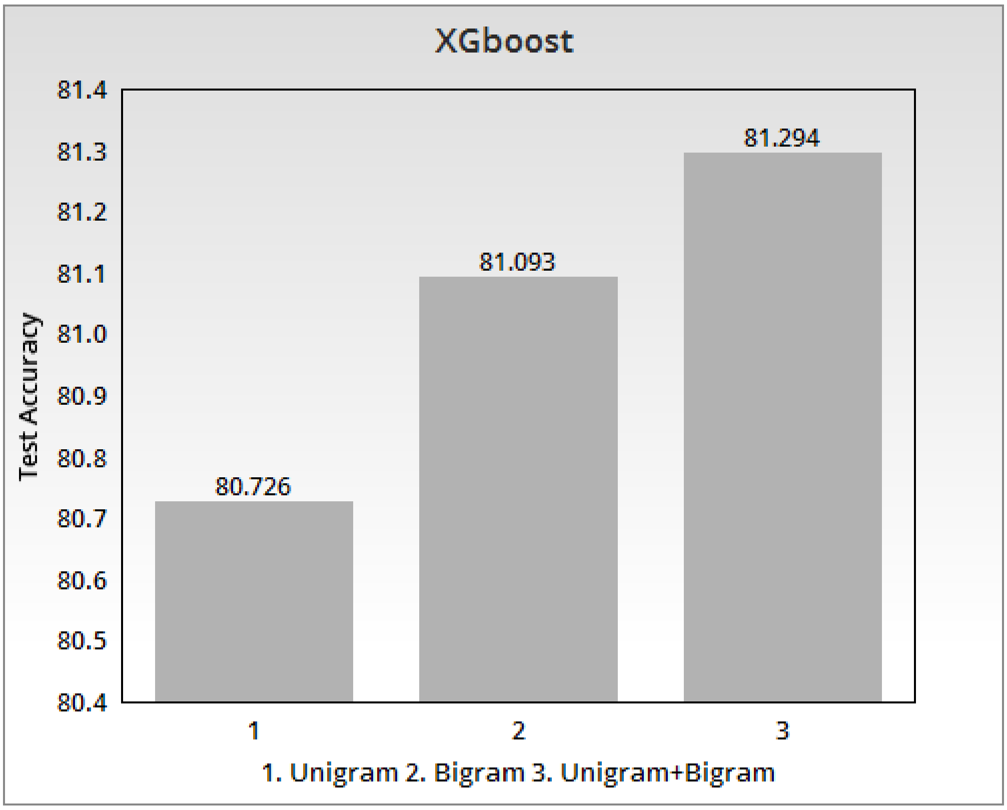


Figure 7.3: XGBoost Results

Chapter 8

Future Work

- Neutral Tweets need to be classified in order to increase the utility of a classifier. Many tweets lack any particular sentiment and are more focused on direct and unbiased statements about facts or events.
- Parts Of Speech tag can be utilized to interpret emotions in a better way. E.g. 'over' as a verb conveys a negative emotion, but as a noun it is neutral (over in cricket).
- Using symbols: During our pre-processing, we discard most of the symbols like commas, full-stops, and exclamation mark. These symbols may be helpful in assigning sentiment to a sentence

Chapter 9

Conclusion

Twitter sentiment analysis comes under the category of text and opinion mining. It focuses on analyzing the sentiments of the tweets and feeding the data to a machine learning model to train it and then check its accuracy, so that we can use this model for future use according to the results. It comprises of steps like data collection, text pre-processing, sentiment detection, sentiment classification, training and testing the model. This research topic has evolved during the last decade with models reaching the efficiency of almost 85%-90%. But it still lacks the dimension of diversity in the data. Along with this it has a lot of application issues with the slang used and the short forms of words. Many analyzers don't perform well when the number of classes are increased. Also, it's still not tested that how accurate the model will be for topics other than the one in consideration. Hence sentiment analysis has a very bright scope of development in future.

Chapter 10

References

[1] Bo Pang and Lillian Lee: *Opinion mining andsentiment analysis, 2008*
[2] Yang C, Lin KH-Y, Chen H-H (2007): *Building emotion lexicon from weblog corpora, 2007*
[3] Alec Go, Richa Bhayani and Lei Huang: *TwitterSentiment Classification using Distant Supervision, 2009*
[4] Apoorv Agarwal, Boyi Xie, Ilia Vovsha, Owen Rambow, and Rebecca Passonneau: *Sentiment analysis of twitter data. In Proceedings of the Workshop on Languages in Social Media, 2011*
[5] Alexander Pak and Patrick Paroubek: *Twitter as a corpus for sentiment analysis and opinion mining, 2010*
[6] Luciano Barbosa and Junlan Feng: *Robust sentiment detection on twitter from biased and noisy data, 2010*
[7] Tsur O, Davidov D, Rappoport A: *A great catchy name: semi-supervised recognition of sarcastic sentences in online product reviews, 2010*

Chapter 11

Bibliography

[1] http://www.noslang.com/dictionary/
[2] http://en.wikipedia.org/wiki/List of emoticons
[3] http://help.sentiment140.com/for-students
[4] http://www.cs.uic.edu/ liub/FBS/sentiment-analysis.html
[5] ”Opinion mining and sentiment analysis(2008)” Bo Pang and Lillian Lee, 1st ed. ISBN-13: 978-1601981509