Mining Indian Tweets to Understand Food Price Rise Crisis

Project report submitted in partial fulfillment of the requirement

for the degree of

Master of Technology
In
Computer Science and Engineering

By

Sheenu (132219)

Under the supervision of

Dr. Pardeep Kumar

May – 2015

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING

JAYPEE UNIVERSITY OF INFORMATION TECHNOLOGY,

WAKNAGHAT, Solan-173234

Himachal Pradesh,India

Certificate

This is to certify that the work titled "**Mining Indian Tweets to Understand Food Price Rise Crisis**" submitted by "**Sheenu**" in partial fulfillment for the award of degree of Master of Technology in Computer Science and Engineering to Jaypee University of Information Technology, Waknaghat has been carried out under my supervision.

This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor ……………………..

Name of Supervisor Dr. Pardeep Kumar

Designation Assistant Professor (Senior Grade)

Date ……………………..

Acknowledgements

I **Sheenu** would like to thank **Prof. Dr. RMK Sinha (Dean CSE and IT)** and **Prof. Dr. S.P Ghrera (Head, Dept. of CSE)** for all the support and guidance they have provided to me during my M.Tech programme. I am thankful for his continuous motivation and encouragement provided to me at every point in time.

I would like to thank **Dr. Pardeep Kumar (Associate M.Tech Research Coordinator)** for offering me the opportunity to do my post-graduation project under his supervision at Jaypee University of Information Technology. In the meetings and discussions, he always gave me right advice and he directed my research in the right direction.

I am thankful to my seniors and friends for their support and help provided during the completion of this thesis. I would sincerely like to thank the entire librarian for their support and help by providing me the study material.

Finally I would like to express my profound thanks to my parents for teaching me how to soar. I would not have made it this so far without their unbounded love, guidance, support and most importantly their prayers.

Signature of the student ……………………..

Name of Student ……………………..

Date ……………………..

Declaration

I hereby declare that the thesis entitled **” Mining Indian Tweets to Understand Food Price Rise Crisis ”** submitted by me for the award of degree of Master of Technology in Computer Science and Engineering, Jaypee University of Information Technology, Waknaghat is original and it has not submitted previously to this or any other university for any degree or diploma.

Signature of Student:

Name of Student:

Date:

Abstract

The work presented in this thesis takes place in the field of text mining and aims more particularly at finding the sentiments of a text. Sentiment analysis is the ongoing field of research in text mining field. The most important achievement of the web, which enables a large number of web users to discuss different issues, is providing immense help to a number of organizations. The help is provided in the form of product reviews, movie reviews and stock market predictions etc. In this paper, the possible role of sentiment analysis in different domains has been projected. The objective of this work is to comparatively analyze the number of techniques used in different domains and various challenges embedded in sentiment analysis. In sentiment analysis, we have to extract the sentiments associated with particular text and to calculate the polarity based on the context of data. Finally, this research provides a hybrid approach for mining Indian tweets for understanding food price crisis. This approach deals with extracting features for creating lexicon and calculating polarity. An enhanced scheme for sentiment analysis of social networking sites can help to understand the food price rise crisis as food price has direct impact on the purchasing power of a large part of Indian population.

Contents

List of Figures

List of Tables

Introduction

Human life is filled with emotions and opinions. We cannot imagine the world without them. Emotions and opinions play a vital role in nearly all human actions. They lead the human life by influencing the way we think, what we do and how we act. Having an access to large quantities of data through internet and its transformation into a social web is no longer an issue, as there are terabytes of new information produced on the web every day that are available to any individual [4]. Even more importantly, it has changed the way we share information. The receivers of the information do not only consume the available content on web, but in turn, actively annotate this content and generate new pieces of information. Today people not only comment on the existing information, bookmark pages and provide ratings but they also share their ideas, news and knowledge with the community at large. In this way, the entire community becomes a writer, in addition to being a reader [7]. The existing mediums like Blogs, Wikis, Forums and Social Networks where users can post information, give opinions and get feedback from other users on different topics, ranging from politics and health to product reviews and travelling. The increasing popularity of personal publishing services of different kinds suggests that opinionated information will become an important aspect of the textual data on the web. Recently, many researchers have focused on this area [1]. They are trying to fetch opinion information to analyze and summarize the opinions expressed automatically with computers. This new research domain is usually called Opinion Mining and Sentiment Analysis [6]. Until now, researchers have evolved several techniques to the solution of the problem. Current-day Opinion Mining and Sentiment Analysis is a field of study at the crossroad of Information Retrieval (IR) and Natural Language Processing (NLP) and share some characteristics with other disciplines such as text mining and Information Extraction [15].

In today’s era, almost four out of five users of internet use social media for some or other context. Some of these include friendship networks, blogging and micro-blogging sites, content and video sharing sites, e-commerce sites etc. The involvement and contribution of the users on the web is increasing day by day. One such contribution is reviews of users in social networking sites. The current trend of giving online reviews enables users to take better decisions who want to use a particular service or purchase a particular product. It helps them to check the popularity of the product. It also enables them to extract the positive or negative features of the products by reading reviews. But manual analysis of such a huge amount of reviews can lead to biased decision. So to provide automation, we are studying sentiment analysis. Sentiment analysis is the modern methodology which analyze huge amount of data to extract sentiments associated with the data. The growth of internet has a special significance in online service. Today, a large amount of population uses social media to give their reviews. The social media Universe is expanding.

Table 1. Social Media users Worldwide

Platform	Monthly active Users
Facebook	1.28 Bilion
Twitter	255 million
Linkedin	1.84 million
Youtube	1 Billion+
Google Plus	540 million

The use of social media is increasing day by day and this is represented by the no of monthly users in the **Table 1.** Social Media users Worldwide. Increasing growth of social media users over internet has also increased their participation in various discussions and activities simultaneously. In case of a product, reviews of users will help to take many important decisions about the services of the product. But manually reading such a bulk amount reviews is a very difficult task. So there is a need of an automatic system which will lead to automatically extract the positive and negative features of the product and make the decision making process easier. There are many sites which do this.

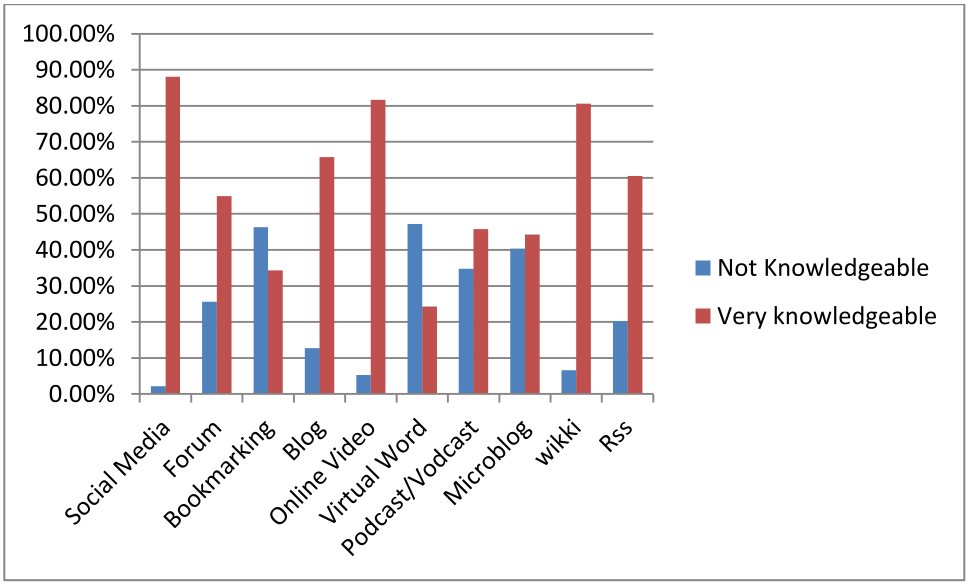


Figure 1 : No. Of Users Knowledgeable about the Social Media tools

1.1.Types of Social Media Applications

There are many social media platforms which has become an integral part of people’s lives these days. Some of the famous and most popular among them are:

- Facebook
- Twitter
- Linkedin
- Quora
- GooglePlus
- Youtube

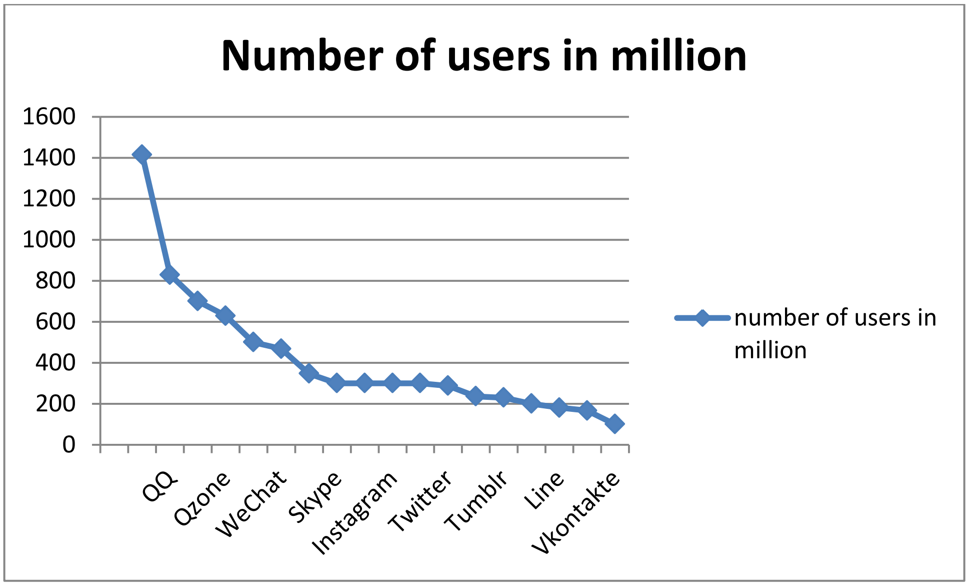


Figure 2 : Depicts the Number of users using Social Media World Wide (2015 statistics)

1.2.Advantages of Social Media

Social Media plays an important role in our life. Several merits of social media use include:

- Compelling and relevant content finds the attention of future customers and increase brand visibility

- Response facility to almost instantly to industry developments and become famous in your field
- It is very cheaper than traditional promotional and advertising activities
- Social content can indirectly encourage links to website content by appearing in general search results, improving search traffic and online sales
- Deliverance improved customer service and respond effectively to feedback
- Customers can find the seller easily through new channels, generating more leads
- Improved loyalty and advocacy from the contacted customers

1.3.Need for analyzing Social media

The use of social media is increasing day by day and this is represented by the no of monthly users as shown in Fig. 1. Increasing growth of social media users over internet has also increased their participation in various discussions and activities simultaneously. In case of a product, reviews of users will help to take many important decisions about the services of the product [12]. But manually reading such a bulk amount reviews is a very difficult task. So there is a need of an automatic system which will lead to automatically extract the positive and negative features of the product and make the decision making process easier. There are many sites and companies which perform these activities.

1.4.Sentiment Analysis

Sentiment analysis is a text classification problem which deals with extracting information present within the text. This extracted information can be then further classified according to its polarity as positive, negative or neutral. It can be defined as a computational task of extracting sentiments from the opinion. Some opinions represent sentiments and some opinions do not represent any sentiment.

- **Sentiments:** Opinions or in other sense can be recognized as someone's linguistic expressions of emotions, beliefs, evaluations etc.
- **Analysis:** To capture the opinions from a pool of users whether the opinion is positive, negative or neutral.
- **Benefit:** Provide efficient information in decision making

Example:

User's Opinion: Person a: it's a great movie (positive statement)

Person b: the new iphone is awesome..!!! (Positive statement)

Person c: Nah!! I didn't like it at all.. (Negative statement)

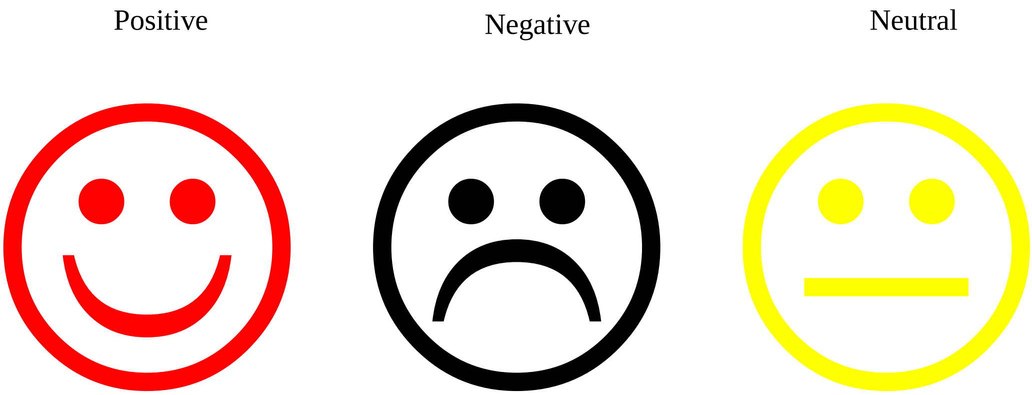


Figure 3: Polarities of Text

Fig. 3 depicts the polarities of sentiment analysis i.e. the text can be classified as Positive, Negative and Neutral. Sentiment Analysis field is interrelated with many other fields such as emotion detection, opinion mining etc. Sentiment analysis is a natural language processing and information extraction task. This technique aims to extract writer's feelings expressed in comments or reviews. Sentiment analysis does not only deal with extracting polarity but also deals with extracting features from the text. The different definitions of sentiment analysis are as follows:

- Opinion mining as a computational task, is defined as follows: given a set of evaluative text documents D that contains opinions or sentiments about an object (person, organization, product etc.), opinion mining aims to extract attributes and components of the object that have been commented on each document d in the set D and to determine whether the comments are positive, negative or neutral [19].

- Another definition of the opinion proposed by the author Bing Liu who defined "feature based sentiment analysis". According to him: an opinion on a feature f is a positive or negative view, attitude, emotion or appraisal on f from an opinion holder.

Sentiment analysis is also about finding subjectivity or objectivity of the opinion. What is subjectivity and objectivity? Subjectivity is about someone's personal review whereas objectivity is the opinion given by an expert. For example: doctor's opinion about the patient on the basis of observed symptoms comes under the objectivity.

Reading huge amount of reviews and discussions over internet is not an easy task and finally to take decision. But these discussions and reviews help in many sectors such as improving e-learning environment, providing personalization in e-learning environment, for getting public response to governmental activities [27].

In the Fig. 4 flowchart of sentiment analysis is represented which gives the general flow of process of sentiment analysis. From the given dataset, what we have to do is to extract the data and segment that data according to parts of speech. After that we will check the sentiments and assign tags to the extracted tokens. In the last step overall polarity of the text is calculated. If the polarity of data is positive, it is positive sentence and if polarity is negative it is negative sentence.

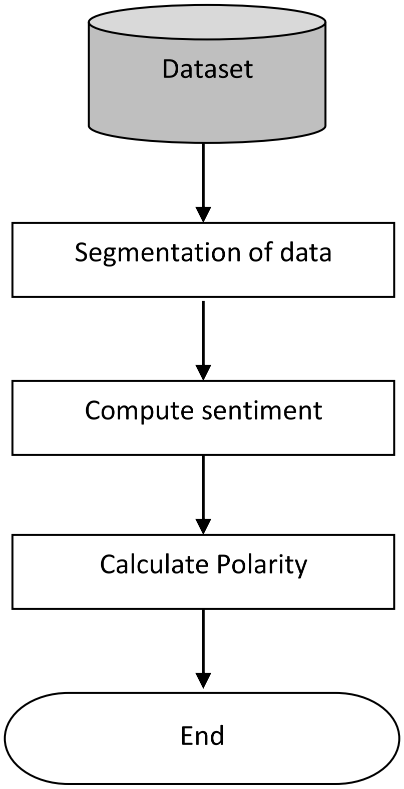


Figure 4 : General Process of Sentiment Analysis

1.5.Applications of Sentiment Analysis

There are many applications of sentiment analysis. Some of the applications are in business and government intelligence.

- Business intelligence seems to be one of the main factors behind corporate interest in the field. A major computer manufacturer, disappointed with unexpectedly low sales, find itself confronting with the question; “why are not consumers buying our laptops?” while concrete data such as the laptop’s weight or the price of a competitor’s model are obviously relevant, answering this question requires focusing on people’s personal views as such objective characteristics. Moreover, subjective judgments regarding intangible qualities examples: “the design is tacky” or even misperceptions- example: “updating device drivers are not available”. When such device drivers so in fact exist- must be taken into account as well [5]. Sentiment analysis technologies for extracting, it would be difficult to directly survey laptop purchasers who have not bought the company’s product. Rather, we could employ a system that

1. Finds reviews or other expressions of opinion on the web- newsgroups, individual blogs, and aggregation sites.
2. Then creates condensed versions of individual reviews or a digest of overall consensus points.

This would save an analyst from having to read potentially dozens or even hundreds of versions of some complaints. Note that internet sources can vary widely in form, tenor, and even grammatically; the fact underscores the need for robust techniques even when only one language is considered [16]. By tracking public viewpoints, one could perform trend prediction in sales or other relevant data.

- Government intelligence is another application that has been considered for example: it has been suggested that one could monitor sources for increases in hostile or negative [21]. Opinions matter a lot in politics.

A major computer manufacturer, disappointed with unexpectedly low sales, find itself confronting with the question;”why are not consumers buying our laptops?” while concrete data such as the laptop’s weight or the price of a competitor’s model are obviously relevant, answering this question requires focusing on people’s personal views as such objective characteristics. Moreover, subjective judgments regarding intangible qualities examples: “the design is tacky” or even misperceptions- example:”updating device drivers are not available”. When such device drivers so in fact exist- must be taken into account as well. Sentiment analysis technologies for extracting, it would be difficult to directly survey laptop purchasers who have not bought the company’s product. Rather, we could employ a system that

3. Finds reviews or other expressions of opinion on the web- newsgroups, individual blogs, and aggregation sites.
4. Then creates condensed versions of individual reviews or a digest of overall consensus points.

This would save an analyst from having to read potentially dozens or even hundreds of versions of some complaints. Note that internet sources can vary widely in form, tenor, and even grammatically; the fact underscores the need for robust techniques even when only one language is considered.

By tracking public viewpoints, one could perform trend prediction in sales or other relevant data. Government intelligence is another application that has been considered for example: it has been suggested that one could monitor sources for increases in hostile or negative. Opinions matter a lot in politics. E-Rule makers allowing the automatic analysis of the opinions that people submit about pending policy or government regulation proposal.

1. Named Entity Recognition - What is the person actually talking about, e.g. is 300 Spartans a group of Greeks or a movie?
2. Anaphora Resolution - the problem of resolving what a pronoun, or a noun phrase refers to. "We watched the movie and went to dinner; it was awful." What does "It" refer to?
3. Parsing - What is the subject and object of the sentence, which one does the verb and/or adjective actually refer to?
4. Sarcasm - If you don't know the author you have no idea whether 'bad' means bad or good.
5. Twitter - abbreviations, lack of capitals, poor spelling, poor punctuation, poor grammar.

1.6. Classification Levels of Sentiment Analysis

Sentiment analysis is also known as opinion mining. Sentiment analysis is a natural language processing and information extraction task that aims to obtain writers feelings expressed in positive or negative comments, questions by analyzing a large number of documents.

An opinion is a quadruple (g, s, h, t) [1]

Where g is the opinion (or sentiment) target, s is the sentiment about the target, h is the opinion holder and t is the time when the opinion was expressed.

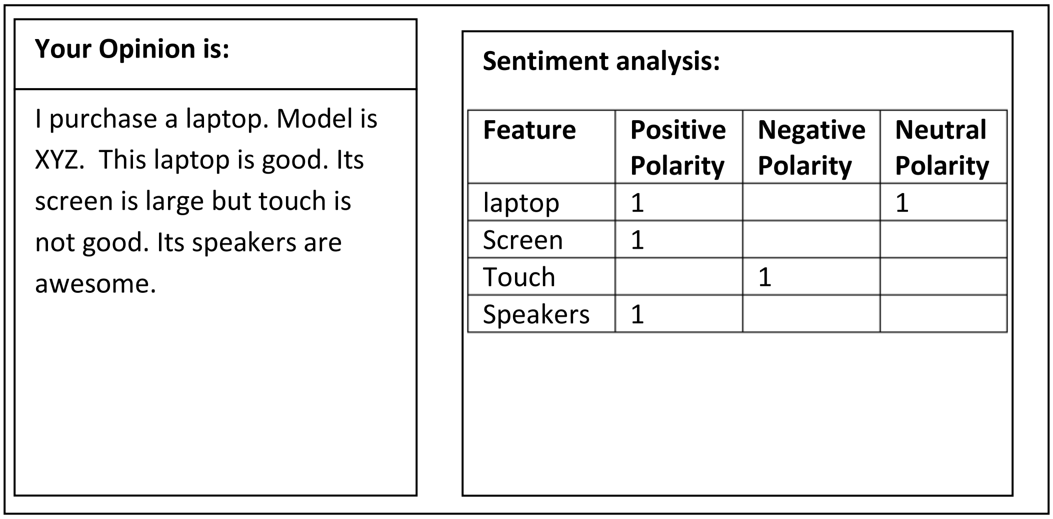
Sentiment analysis is an ongoing field of research in text mining field. SA is the computational study of Opinions, sentiments, subjectivity toward an entity. The entity can represent individuals, events or topics. The two expressions sentiment analysis and opinion mining are interchangeable.

They express a mutual meaning. But also in some contexts they have different meaning. Opinion mining extracts and analyzes people’s opinion about an entity while sentiment analysis identifies the sentiment expressed in a text then analyzes it. Therefore, the target of sentiment analysis is to find opinions, identify the sentiments they express, and then classify their polarity. Sentiment analysis can be considered as a classification process.

Sentiment analysis is considered as a classification process. There are main three classification levels in sentiment analysis:

- Document Level
- Sentence Level
- Aspect Level

1. **Document Level:** Document Level Sentiment Analysis aims to classify an opinion document as expressing a positive or negative opinion or sentiment. It considers the whole document a basic information unit (talking about one topic).
2. **Sentence Level:** Sentence Level aims to classify sentiment expressed in each sentence. The first step is to identify whether the sentence is subjective or objective. If the sentence is subjective, the sentence level sentiment analysis will determine whether the sentence expresses positive or negative opinions. Sentiment expressions are not necessarily subjective in nature. However, there is no fundamental difference between document and sentence level classifications because sentences are just short documents. Classifying text at the document level or at the sentence level does not provide the necessary detail which is needed in many applications, to obtain these details; we need to go to the aspect level.
3. **Aspect Level:** Aspect level sentiment analysis aims to classify the sentiment with respect to the specific aspects of entities. The first step is to identify the entity and their aspects. The opinion holder can give different opinions for different aspects of the same entity like this sentence: The voice quality of this phone is not good, but the battery life is long.



Your Opinion is:

I purchase a laptop. Model is XYZ. This laptop is good. Its screen is large but touch is not good. Its speakers are awesome.

Sentiment analysis:

Feature	Positive Polarity	Negative Polarity	Neutral Polarity
laptop	1		1
Screen	1		
Touch		1	
Speakers	1		

Figure 5 Example of Classification levels

1.7.Approaches of Classification

Text mining is branch of NLP (Natural Language Processing), i.e. used to extract automatically meaningful information from unstructured information which is usually textual data. This extracted information is transformed into numeric values and thereafter used by different data mining algorithms [8]. It can also be said that the purpose of text mining is to "transform text in to numeric form" to incorporate textual information in the application of predictive analysis. It is believed that commercial potential value of text mining is high as most of the information around 80 % is stored in textual format. There are different sentiment classification techniques. Sentiment classification has several important characteristics including the various tasks, features, techniques, and application domains. These are summarized in the taxonomy presented in Table 2

Table 2: Characteristics of sentiment analysis

	Tasks	
Category	Description	Label
Classes	Positive/negative sentiments or objective/subjective texts	C1
Level	Document or sentence/phrase-level classification	C2
Source/Target	Whether source/target of sentiment is known or extracted	C3
	Features	
Category	Examples	Label
Syntactic	Word/POS tag n-grams, phrase patterns, punctuation	F1
Semantic	Polarity tags, appraisal groups, semantic orientation	F2
Link Based	Web links, send/reply patterns, and document citations	F3
Stylistic	Lexical and structural measures of style	F4
	Techniques	
Category	Examples	Label
Machine Learning	Techniques such as SVM, naïve Bayes, etc	T1
Link Analysis	Citation analysis and message send/reply patterns	T2
Similarity Score	Phrase pattern matching, frequency counts, etc.	T3
	Domains	
Category	Description	Label
Reviews	Product, movie, and music reviews	D1
Web Discourse	Web forums and blogs	D2
News Articles	Online news articles and Web pages	D3

Lexcon- based approach[2]: In lexicon based approach, pre-built lexicons of words with the assignment of sentiment orientations are used to determine the overall polarity of the text. These methods result in good accuracy when used in the well-known domains. However these approaches have two main limitations. One is, number of words is limited in the lexicon, so it is difficult to extract polarity for a dynamic domain such as face book, twitter etc. the other is in lexicon based approach, a fixed orientation and score is assigned to the words, but the same words can be used to represent different meaning in different context.

Machine Learning approach[3]: In general, sentiment analysis is concerned with analyzing direction based text, determining whether a text is subjective or objective and whether a sentence contains positive or negative is a two class problem that involves classifying sentence as positive or negative. Machine learning approaches consider the sentiment analysis as topic based text classification problem. There are many algorithms that can be employed in machine learning approach i.e. Naïve Bayes or support vector machines.

- **Machine Learning Approaches**

  There are various approaches to design machine learning algorithms. The purpose of ML algorithms is to use observations as input and this observation can be a data, pattern and past experience. Thus Ml algorithms use to improve the performance of instances, which can be done by any classifier by trying to classify the input pattern into set of categories or to cluster unknown instances. As the nature of ML algorithms it enhances its performance from past experience or by receiving feedback. It can be divided into two categories supervised and unsupervised approach [29].

  - Supervised: In supervised learning, the instances are labeled with known or target classes labels. Here before classification the dataset knows the target class. Thus it is very helpful for the problems which have known inputs.
  - Unsupervised: In unsupervised learning, the algorithm groups the instances by their similarities in values of features and makes different clusters. In it no prior class or clusters are given, the algorithm itself defines their clusters automatically and statistically.

Some algorithms are:

1) KNN Algorithm

KNN is type of instance based learning or lazy learning. In this learning the function is approximately locally and all computation is deferred until classification. It is simplest of all machine learning algorithms. In KNN classification, the output is class membership. An object is classified by majority votes of its neighbors by the object being assigned to class most common among its k nearest neighbor (k is positive small integer). The nearest neighbor is determined using similarity measure usually distance functions are user. Following are the distance function used by KNN [15].

2) Chi-square(X2)

Let n be the total number of documents in the collection,pi(w) be the conditional probability of class i for documents which contain w, Pi be the global fraction of documents containing the class I, and F(w) be the global fraction of documents which contain the word w. Therefore, the X2-statistic of the word between word w and class i is defined as:

$$Xi\ 2=\frac{n.F(w)2.(pi(w)-Pi)2}{F(w).\left(1-F(w)\right).Pi(1-Pi)}$$

Equation 1

Hybrid Approach[4]: Examining the strengths and limitations of each approach, a combined approach was designed. In the combined approach, two stages are used. In first stage, lexicon based approach is used and in second stage, the labeled messages obtained from first stage are used as the training set for a machine-learning based classifier.

1.8. Goals of the Work

To develop a framework which will remove unrelated comments, extract features, classify text and calculate the polarity of the text.

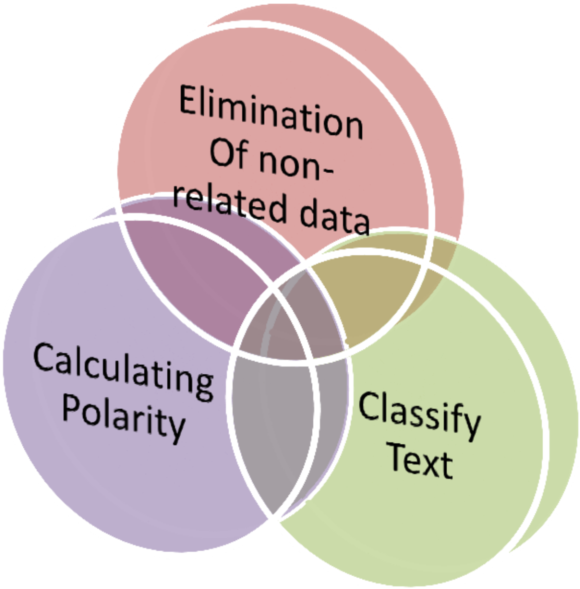


Figure 6: Goal of the work

Scope of Thesis

The thesis aims to mine Indian tweets to find sentiments of people regarding food price crisis using twitter data. The twitter data will be obtained through tweepy API using twitter secret key and access token. The data will be filtered in real time and will be subjected to preprocessing. The relevant tweets will have to be categorized and a dictionary of tweets will be created. Using those tweets, KNN algorithm will be applied on it for clustering those tweets as positive or negative tweets or neutral tweets.

Thesis Outline

The remainder of the thesis is as follows. Chapter 2 deals with the various literatures which are present in this field and related fields. It discusses the amount of work which are currently being carried out in the scientific community. Chapter 3 proposes our problem statement and gives a mathematical shape to it so that various algorithms could be applied on it. Chapter 4 introduces our methodology and discuss the implementation proposal for various algorithms and our own algorithm. Finally Chapter 5 concludes the report with a discussion on the various pros and cons of the methods.

2. Literature Review:

In previous chapter we have discussed different levels of sentiment analysis and different approaches used to extract the sentiments. In this chapter we have briefly depict the several approaches available in literature for extracting sentiments and improving accuracy. The current e-commerce environment, social media networks are providing different reasons for extracting the data and using that data for analyzing different issues. One of the reasons to analyze this text is to extract hidden issues. If these hidden issues are efficiently extracted, it will helpful in various ways. Due to this, a lot of research is ongoing in this field. Further, sentiment analysis is a broad area which deals with text or particularly we can say that with big data. Each and every approach is developed in order to improve the accuracy. Results of manual analysis and computational analysis are compared to check the accuracy. In the continuation of this topic the first paper which I have referred is a survey paper i.e.

2.1.Sentiment analysis algorithms and applications: A survey [1]

2.1.1. **Feature Selection in sentiment classification***:*

Sentiment analysis is considered as Sentiment Classification Problem. The first step in the sentiment classification problem is to extract and select text features. This is the survey paper which described the sentiment analysis in detail. The survey paper gives brief explanation to famous feature selection and sentiment classification algorithms. Some of the current features are:

- **Term presence and frequency:** These features are individual words or word n-grams and their frequency counts. It either gives the words binary weighting or uses term frequency weights to indicate the relative importance of features.
- **Parts of speech (POS):** finding adjectives, as they are important indicators of opinions.
- **Opinion words and phrases:** these are words commonly used to express opinions including good or bad, like or hate. On the other hand,

some phrases express opinions without using opinion words. For example: cost me an arm and a leg.

- **Negations:** the appearance of negative words may change the opinion orientation like not good is equivalent to bad.

Further in this paper author had discussed the feature selection methods.

2.1.2. Feature selection methods

2.1.2.1. ***Point-wise Mutual Information:*** the measure of mutual information offers an official way to model the mutual information among the classes and the features. Information theory was used to derive this measure. The definition of point-wise mutual information (PMI) $M_{i(w)}$ among the word w and the class I is given on the basis of the level of co-occurrence among the class I and word w. The estimated co-occurrence of class I and word w, from the root of mutual independence, is given by $\mathbf{p_i .F(w)}$, and $\mathbf{F(w).p_i(w)}$ is the true co-occurrence. The definition of mutual information is well-defined in terms of the ratio of these two values and is presented by the following equation: $\mathrm{Mi(w)} = \left(\frac{\mathrm{F(w).pi}}{\mathrm{F(w).Pi}}\right) = \log\left(\frac{\mathrm{pi(w)}}{\mathrm{Pi}}\right)$. The positive correlation of the word w to the class I is given when Mi(w) is greater than 0 and the word w is negatively correlated to the class I when Mi(w) is less than 0. Feature Selection methods can be divided into lexicon-based methods that need human annotation, and statistical methods which are automatic methods that are more frequently used. Lexicon based approaches usually begin with a small set of 'seed' words. Then they bootstrap this set through synonym detection or on-line resources to obtain a larger lexicon. Statistical approaches, on the other hand, are fully automatic. The feature selection techniques treat the documents either as group of words (Bag of words (BOWs)), or as a string which retains the sequence of words in the document. BOW is used more often because of its simplicity for the classification process. The most common feature selection step is the removal of stop-words and stemming (returning the word to its stem or root i.e. flies→fly).

In the next subsections, the author presents three of the most frequently used statistical methods in feature selection and their related articles.

2.1.2.2. ***Chi-square(X2):***let n be the total number of documents in the collection,pi(w) be the conditional probability of class i for documents which contain w, Pi be the global fraction of documents containing the class I, and F(w) be the global fraction of documents which contain the word w. Therefore, the X^2-statistic of the word between word w and class i is defined as:

$$X_i{}^2 = \frac{n.F(w)2.(pi(w)-Pi)2}{F(w).\left(1-F(w)\right).Pi(1-Pi)}$$

Equation 2 : Chi Squrare Equation

2.2. Serendio: Simple and Practical lexicon based approach to Sentiment Analysis [2]

- *Abstract:* In this paper a lexicon based approach for discovering sentiments is used. Lexicon is built from the Serendio taxonomy which consists of negative, positive, stop words, negation, and phrases. A usual tweet comprises word variations, hashtags, emoticons etc. Some processing steps which are used in the whole process are emotion detection, stemming and normalization, hashtag detection and exaggerated word shortening are used in the whole process. The lexicon based classifier classifies the tweets in terms of positive and negative on the basis of contextual sentiment orientation of the words.
- Introduction: Social media websites like Twitter, Facebook etc. are a major hub for users to express their opinions online. On these social media sites, users post comments and opinions on various topics. Hence these sites become rich sources of information to mine for opinions and analyze user behavior and provide insights for: User behaviors, Product feedback, User intentions, Lead generation.
- Approach: Serendio Sentiment engine Extracts and analyzes sentiments for a given product and feature set. Serendio sentiment engine currently works for eight different domains such as banking, tablets, smartphones, televisions, apparel, gaming, automobiles and e-readers. The lexicon is manually created in this approach. Two types of lexicons are created.

  Common lexicon: this contains data that would have the same semantic meaning or sense across different domains and categories.

Common or default sentiment word: positive and negative words that have the same sentiment value or sense across different domains. For e.g. sentiment word “good” always represents a positive sentiment and it is independent of any category. Positive or negative sentiment words have a sentiment score of +1 or -1 to indicate the respective polarity.

Negation Words: Negation Words are the words which reverse the polarity of sentiment. For example, “the battery life is not good” has negative sentiment.

Blind Negation words: in the sentence,” The T.V needs a better remote”, “needs” is a blind negation word. Blind negation words operate at a sentence level and points out the absence or presence of some sense that is not desired in a product feature.

Split words: Split words are the words used for splitting sentences into clauses. The split words list consists of conjunctions and punctuation marks. For example the complex sentence,” Camera is good but the battery is bad” is split into two clauses “Camera is good” and “Battery is bad”.

- Category specific lexicon: Category specific lexicon contains the 1) Product Catalog which identifies all the products that we are interested in. 2) Feature Catalog which is a list of attributes that the product has. This enables the Serendio engine to do analysis at the feature level. 3) Sentiment words (Positive and negative) that are specific to the category. For example, for a category such as Televisions, a product would be Samsung TV. The feature would be LCD screen and the word “glare” would be the category specific negative sentiment word.

    A typical tweet comprises of emoticons, word variations, and hashtags etc. The objective of the preprocessing step is to normalize the text into an appropriate form to extract the sentiments. Below are the preprocessing steps used :

    POS tagging

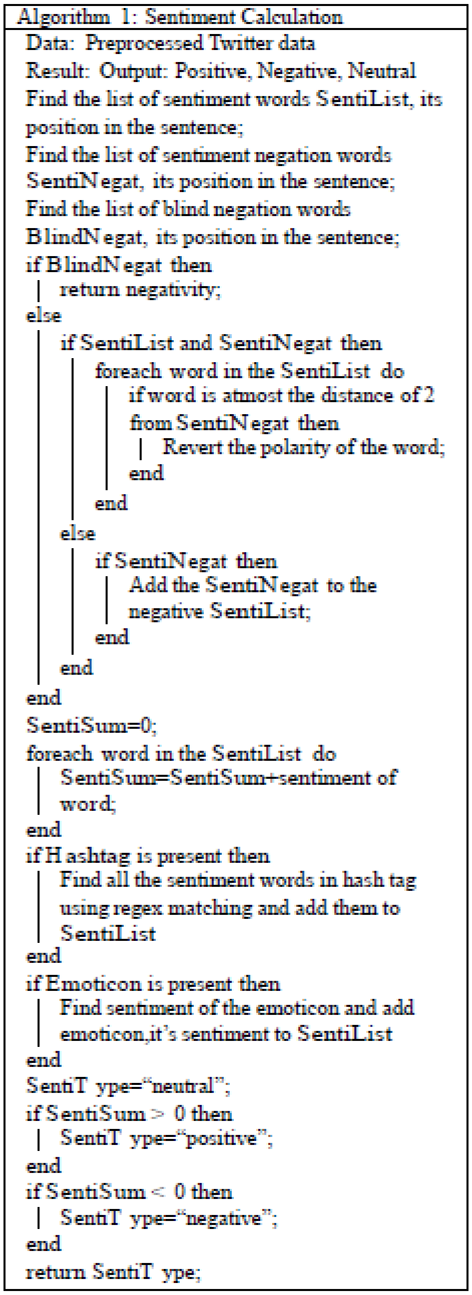
    Stemming

    Exaggerated word shortening

Emoticon detection

Hashtag Detection

Preprocessing: the following algorithm is used in this paper.



```
Algorithm 1: Sentiment Calculation
Data: Preprocessed Twitter data
Result: Output: Positive, Negative, Neutral
Find the list of sentiment words SentiList, its
position in the sentence;
Find the list of sentiment negation words
SentiNegat, its position in the sentence;
Find the list of blind negation words
BlindNegat, its position in the sentence;
if BlindNegat then
    return negativity;
else
    if SentiList and SentiNegat then
        foreach word in the SentiList do
            if word is atmost the distance of 2
            from SentiNegat then
                Revert the polarity of the word;
            end
        end
    else
        if SentiNegat then
            Add the SentiNegat to the
            negative SentiList;
        end
    end
end
SentiSum=0;
foreach word in the SentiList do
    SentiSum=SentiSum+sentiment of
    word;
end
if Hashtag is present then
    Find all the sentiment words in hash tag
    using regex matching and add them to
    SentiList
end
if Emoticon is present then
    Find sentiment of the emoticon and add
    emoticon,it's sentiment to SentiList
end
SentiType="neutral";
if SentiSum > 0 then
    SentiType="positive";
end
if SentiSum < 0 then
    SentiType="negative";
end
return SentiType;
```

Figure 7 Algorithm used for Preprocessing

2.3.Sentiment analysis in Facebook and its application to e-learning [3]

In stage 1, a dictionary of words is used, in which each word has its sentiment orientation (positive/negative emotional polarity). Each message is classified by following a number of steps. These steps are as follows:

1. **Preprocessing:** In the first step, message is preprocessed by converting all the words into lower-case. Then detection of the idioms is done and they are joined so as to consider as a unique word. For example, as good as is converted into as-good-as.
2. **Segmentation into sentences:** Then, the message is fragmented into sentences. Dots are considered as only punctuation marks that act as a separator at this step. As other punctuation marks such as commas or semicolons can be part of the emoticons.
3. **Tokenization 1(partial):** In this step, tokens are extracted from each sentence. In this only white spaces are considered to separate the tokens as other punctuation marks such as semi colon, hyphen can be the part of emoticons.
4. **Emoticon detection:** Next is the detection of emoticons. Consecutive occurrence of symbols is considered as presence of emoticons which are compared with text files containing emoticons, extracted from Wikipedia.
5. **Tokenization 2(complete):** in this step, the final set of tokens are extracted by removing all the punctuation marks such as commas, hyphen etc which are left after the removal of emoticons.
6. **Interjection detection:** this step deals in detecting the interjections. The interjections such as hehehe, lolz etc are marked as positive whereas interjections such as uff (tiredness) are marked as negative. This detection of interjection is implemented by the use of regular expressions. Because interjections are observed as set of repeated letters in the word. Such as long sequence of hehehehehe can be considered as strong happy sentiment.

7. **Token score assignation:** the next step is the assignment of the score to each token. 1 is assigned if the token represents positive polarity, -1 is assigned if the token represent the negative polarity; and 0 is assigned if the token is having neutral polarity. To assign a score, the classifier checks if the token is positive/negative emoticons, positive/negative interjection, or whether the word is present in the predefined dictionary. Also repetitive letters are removed in this step if the word does not have any match in the predefined dictionary. As the language written in the social networking sites such as face book is very casual. As in greaaaaat repetition of a leads to undetected word. So removal of a leads to match of the word. Sometimes spelling mistake also lead to the failure of detection of the word, so spelling checker is also used so that word can be corrected and accordingly polarity can be assigned.
8. **Syntactical analysis:** in this stage, syntactical analysis is done, where each token is checked for whether its polarity can be reversed or not which is because of negations. Negations are detected and polarity is reversed for that token.
9. **Polarity calculation:** For calculating polarity of a sentence, the numbers of tokens that are considered for conveying sentiment after the removal of determinant, articles, prepositions etc. are calculated. Such words are considered as stop words. Then each token is assigned a scored and polarity score for the sentiment of a sentence is calculated as the sum of the scores divided by the sum of all the candidates to receive a score. The score will lie between -1 to +1.

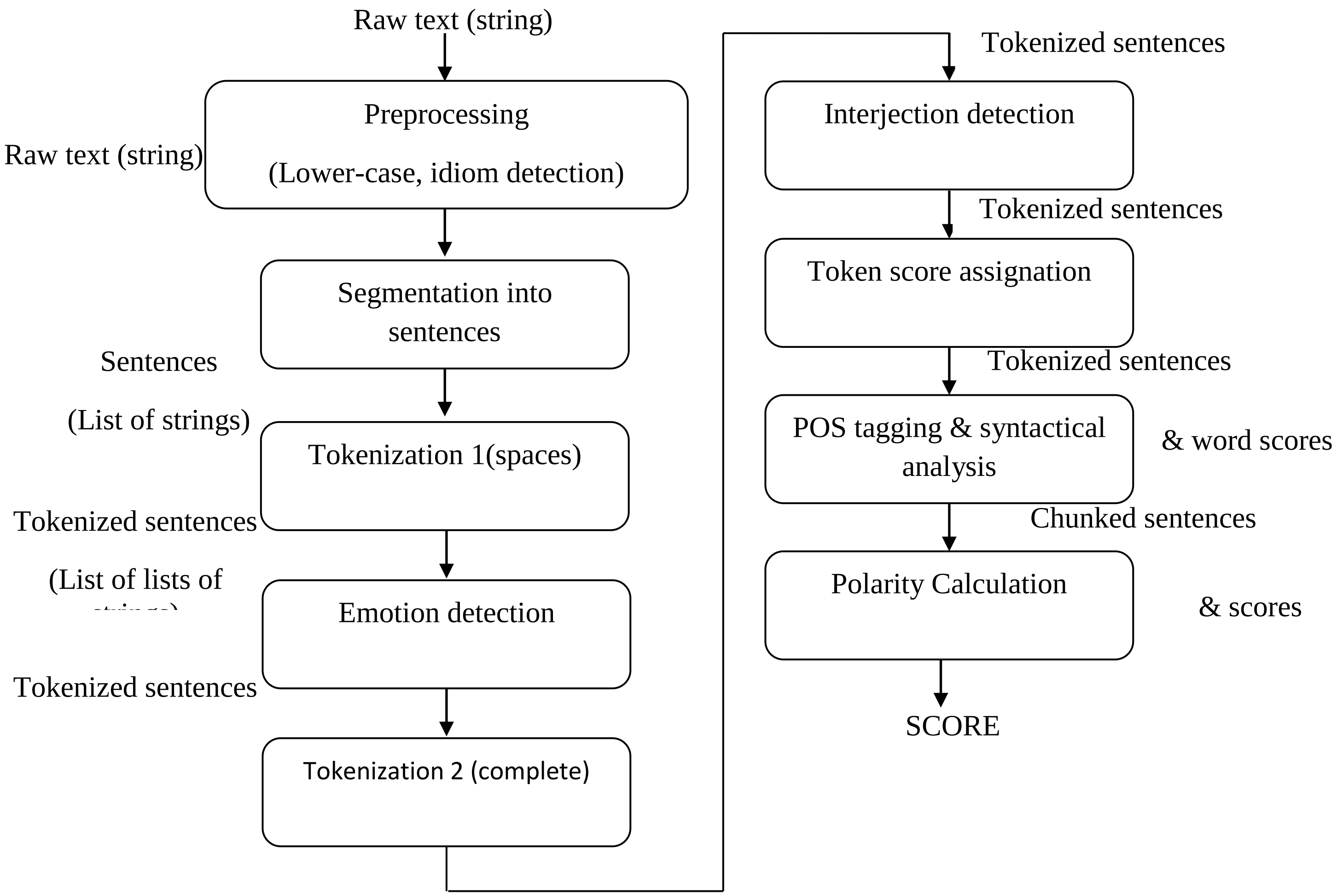


Figure 8 : A flowchart to represent the complete process

2.4.Sentbuk: Sentiment analysis for e-learning environments [4]

Sentiment change analysis: the main goal of this approach is not only extracting sentiment of user at a certain time but also to capture the habits of user's activities which will help in detecting the change of sentiments or the user sentiment state. It can also be explained as capturing user's regular pattern. The first point to be noticed is which information of user should be collected regarding which regular pattern is created. Collecting information about user's action will help to determine the sentiment changes of the user. As in face-book application sent-buk given in the paper, number of parameters is taken into consideration apart from simply checking the number of messages. The different parameters which are considered are:

- Mean of the sentiment showed by the messages published by the user(s).
- Number of messages written (m).
- Number of comments to messages made (c).
- Number of likes made to messages on his/her wall (l).
- Number of likes made to comments to messages on his/her wall (k).

Using all these parameters, user's patterns of interactions in different weeks will be compared.

One more but very important parameter to note is minimum amount of time necessary to track a user activity.

We use vector comparisons. Each vector, which we call profile (P), contains the data collected for each user (u) along a week (w).

P(u,w)= (m,c,l,k,s)

Through a set of weekly profiles, changes in the user's behavior could be detected. For example, if a user writes two or three messages per week and one week writes twenty messages with a lot of comments, then this may be a sign of something different may be happening to user. Similarly, if a user interacts with facebook daily, writing a lot of messages, commenting on other's wall and so on, and shows an extremely low activity for a while, this silence can also represent an emotional change.

Emotional changes will be detected by comparing vectors. In order to set the user's regular pattern, the median of each attribute, considering a set of weekly vectors, is calculated. Once this regular pattern is built (P), it is compared with last week profile (Q).

The easiest way to compare vectors is the Euclidean Distance

$$D_E(P,Q)= \sqrt{}(p1 - q1)2\ +\ (p2 - q2)2\ +\ (p3 - q3)2\ + \cdots +\ (pn - qn)2$$

Equation 3

$$=\sum_{i=1}^{n}(pi - qi)^2$$

Equation 4

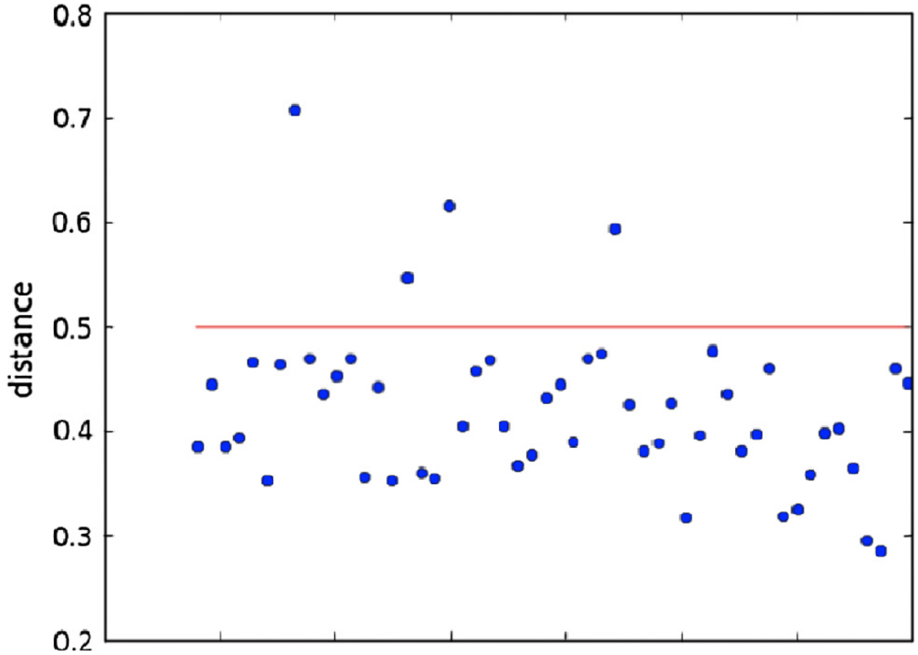


Figure 9: A graph of user profiles

Table 3 : Comparison of system with manual analysis

	Positive	**Neutral**	**Negative**	**Accuracy(Spanish)(%)**
Positive	920	23	19	95.63
Neutral	74	704	65	83.51
Negative	89	109	760	79.33

2.5Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications [5]

2.5.1 *Abstract:* This article offers a short-term overview of the modern trends in the field and defines the manner in which the objects or entities contained in the distinct issue contribute to the development of the area. Finally, we remark on the recent challenges and envisioned improvements of the sentiment analysis fields and the subjectivity, also with their application to other related domains and Natural Language Processing tasks.

2.5.2 Introduction: In computational linguistics, the automatic detection of affect in texts is becoming increasingly important from an applicative point of view. Consider for example the tasks of opinion mining, market analysis, or natural language interfaces such as e-learning environments or educational/edutainment games. For instance, the following represent examples of applicative scenarios in which affective computing could make valuable and interesting contributions:

- Sentiment analysis Text categorization according to affective relevance, opinion exploration for market analysis, etc., are examples of applications of these techniques. While positive/negative valence annotation is an active area in sentiment analysis, a fine-grained emotion annotation could also contribute to the effectiveness of these applications.
- Computer assisted creativity. The automated generation of evaluative expressions with a bias on certain polarity orientation is a key component in automatic personalized advertisement and persuasive communication.
- Verbal expressivity in human–computer interaction Future human–computer interaction is expected to emphasize naturalness and effectiveness, and hence the integration of models of possibly many human cognitive capabilities, including affective analysis and generation. For example, the expression of emotions by synthetic characters (e.g., embodied conversational agents) is now considered a key element for their believability. Affective words selection and understanding is crucial for realizing appropriate and expressive conversations

2.5.3 Recent trends in subjectivity and sentiment analysis: the following are some trends.

- Multilingual subjectivity and sentiment analysis
- Subjectivity and sentiment analysis in Social Media

2.6SentiWordNet: A publicly Available Lexical Resource for Opinion Mining [6]

2.6.1 *Abstract:* Opinion mining (OM) is a recent sub discipline at the intersection of computational linguistics and information retrieval which is concerned not only with what the topic a document is about, but also with what opinion it expresses. OM has an ironic set of applications, ranging from tracking users' opinions about political candidates or about products or about policies as conveyed in online forums, to customer relationship management. With the purpose of the extraction of opinions from text, current research has tried to automatically determine the "PN-polarity" in the terms of subjectivity, i.e. recognize whether a term which is an indicator of opinionated content has a positive or a negative association. Research on finding and formulating whether a term is indeed an indicator of opinionated content (a subjective term) or not (an objective term) has been, in its place, much scarcer. In this work we define SENTIWORDNET, a lexical resource in which every WORDNET synset s is connected to three numerical scores Pos(s), Obj(s) and Neg(s), unfolding how positive, objective, and negative the terms contained in the synset are. The process used to grow SENTIWORDNET is created on the basis of if quantitative analysis of the glosses connected to synsets, and also on the use of the resulting vectorial term illustrations for semi-supervised synset classification. The three scores are extracted form merging the results formed by a group of eight ternary classifiers; all categorized with almost similar accuracy levels but with different classification behavior. SENTIWORDNET is spontaneously and freely available for the purposes of research, and is awarded with a user interface such as Web-based graphical interface.

2.6.2 Introduction: Opinion mining (OM – also recognized as "sentiment classification") is a recent sub discipline at the intersection of computational linguistics and information retrieval which is concerned not with what the topic a text is about, but also with what the opinion it expresses. Opinion-driven content management has

several important applications, such as determining critics' opinions about a given product by classifying online product reviews, or tracking the shifting attitudes of the general public towards a political candidate by mining online forums or blogs. Within OM, several subtasks can be identified, all of them having to do with tagging a given text according to expressed opinion:

1. Determining text SO-polarity, as in deciding whether a given text has a factual nature (i.e. describes a given situation or event, without expressing a positive or a negative opinion on it) or expresses an opinion on its subject matter. This amounts to performing binary text categorization under categories Subjective and Objective (Pang and Lee, 2004; Yu and Hatzivassiloglou,2003);

2. Determining text PN-polarity, as in deciding if a given Subjective text expresses a Positive or a Negative opinion on its subject matter (Pang and Lee, 2004; Turney, 2002);

3. Determining the power of text PN-polarity, as in determining e.g. whether the Positive opinion conveyed by a text on its subject matter is Strongly Positive, Mildly Positive, or Weakly Positive (Pang and Lee, 2005; Wilson et al., 2004).

2.7Sentiment Miner: A prototype for Sentiment Analysis of Unstructured Data and Text [7]

2.7.1 Abstract--This paper presents a method to apply opinion mining on unstructured text for polarity extraction and classification at sentence level within a document. The generation of massive unstructured information about individuals makes the task of progress tracking and monitoring almost impracticable which results in the quest to find some way for automated text analysis and tagging. The proposed solution in this work is the development of a System (Sentiment Miner). It will provide features to process and classify text files (reviews and appraisals) for opinion mining at sentence level using Natural language Processing techniques and Opinion Mining algorithms. The prototype of a final product; a Semantic Search Engine will facilitate in document retrieval for analysis whenever required.

2.7.2 Introduction-- The explosion of Internet has not only led to the generation of gigantic volumes of unstructured information in the form of web documents, but also a huge amount of text is produced in the form of evaluations, appraisals and reviews etc. This generated text acts as potential gold mines for extracting important information from unstructured documents. The important knowledge that we can gain from such gold mines can be in the form of summarization of text to extract the real essence or it can be in the form of predictions that we can get from this text analysis. Another type of work that we mostly do is document classification and it can be on the basis of writer, subject or topic. One such application of this approach is document classification on the basis of sentiments or opinion, which is covered under a relatively newer search area called Sentiment Analysis. Sentiment analysis or opinion mining refers to a broad area of natural language processing, text mining and computational linguistics. Commonly speaking, it targets to determine the attitude of a presenter or a writer regarding some topic. The attitude may be their evaluation or judgment, their affective state (e.g., the emotional state of the writer or poet or the author when writing) or the intentional emotional communication (e.g., the emotional effects the writer, the poet or the author wishes to have on the reader). Opinion mining research considers the computational treatment of subjective information contained in text. With the rapid growth of available subjective text on the Internet in the form of reviews, blog posts and comments, it can assist in a number of potential applications in areas such as search engines, recommender systems and market research.

2.7.3 **Procedure—**

Document Uploading and Initial Tagging: Documents collected from different users have been uploaded on a file server after converting into text format (*.txt files). While uploading, these documents have been tagged properly with a unique identifier and the information regarding author and subject has also been stored in a database table against that document identifier.

Document Processing for Sentiment Extraction Once documents are uploaded onto file server, these documents are ready for processing. Document Processing is the core module (use case) of Sentiment Miner, which involves different subtasks like Text Tokenization, Part of Speech Tagging, Part of Speech Filtering, Polarity Calculation, Feature Extractions and Text Classification. The system snapshot (below) describes "Process Document" module which is capable to perform these tasks. It enables us to process one-tomany files or a complete text files repository at a time. It also gives control to select parts of speech that will be counted while polarity calculation.

Step 1: Text Tokenization (Tokenizer) In the first step of document processing every text file is divided into paragraphs which further broken down into individual sentences for part of speech tagging. The tokenizer module in Sentiment Miner performs these tasks as follows. Conjunction which impacts the meaning of different parts of sentences has also been handled in the first step.

Step 2: Part of Speech Tagging (POS Tagger) It's important to determine the grammatical class, a token (output of tokenizer) belongs to. This can be done by fixing tags to tokens, representing the part of speech being used by a word in the bag of tokens. A part of speech tagger is an application that performs this task. Taggers are usually built by statistical analysis of patterns from large corpus of documents with annotated parts of speech. The Penn Treebank

(Marcus et al, 1993) and Brown Corpus (Garside, 1987) being admired examples of available taggers. The Brill part of speech tagger (Brill, 1992) is one which is frequently used algorithm based on building tagging rules from annotated documents. Other approaches to part of speech tagging have been proposed by using maximum entropy techniques (Toutanova et al, 2000) and by building statistical Markov models (Brants, 2000). We used OpenNLP API in this work, an implementation of Eric Brill Tagger that uses Penn Tree Bank Tags, a set of parts of speech tags for tagging.

Step 3: Part of Speech Filtering (POS Filter) Most of the past work on determining the strength of subjective expressions within a sentence or a document uses specific parts of speech such as Adjectives, Verbs and Nouns in which adjectives are commonly used to extract the sense of a sentence (Bo Pang L. L., 2002)(Farah Benamara, 2007). The lexicon selected in this research contains polarity scores for, Verbs, Adverbs and Adjectives. So it can be one idea to filter all the words with these four tags: Nouns, Verbs, Adverbs and Adjectives and second idea can be “only Adjectives” to make things simple.

Step 4: Polarity Extraction (Extractor) The fourth and last step involved in sentiment analysis of a text is polarity extraction. In this step we submit the filtered tokens to Sentiment Miner Extractor module which finds the score of each lexical unit by looking into a custom build lexicon, resolves the word sense ambiguity whenever required, determine the impact of Adverbs on Adjectives, detect negation and apply negation rules if required and finally combine all these to calculate sentence score.

2.8 Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon [8]

A Multi-aspect sentiment analysis for Chinese online Social Reviews (MSA-CORs) is performed. The unsupervised learning proposed by this paper is used to automatically find out the aspects conferred in Chinese Social Review. A LDA (Latent Dirichlet Allocation) model is applied to find out multi-aspect global topics of social reviews. After that based on the sliding window context above the review text, extracts the local topic. A trained LDA model is modeled to identify the local topic and the use of HowNet lexicon is done to extract the associated polarity. The accuracy obtained in the terms of accuracy is 91.23%.

2.9Lexicon-based comments-oriented news sentiment analyzer system [9]

This paper presents a Lexicon based Comments oriented News Sentiment Analyzer(LCN-SA) that consists of two modules that are automatic focus detection module and sentiment analyzer module. The authors in this paper present a technique that is able to analyze the three aspects. These three aspects are:

1. The ability of analyzer to make the user's able to expree their views in non-standard languages.
2. In multi-domain scenario, the ability of analyzer to detect the target from user's opinions.
3. The analyzer also deals with the design of a lingustic modularization knowledge model with low cost adaptability.

2.10. Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews [10]

This paper addresses the reviews of restraurant and proposed a new senti lexicon for the sentiment calculation. Using supervised learning algorithm for classifying the review document as a positive and as a negative polarity, there comes a difference of 10% classification accuracy as positive classification accuracy appears higher than the negative classification accuracy. This creates a problem which decreases the average accuracy when accuracies of two classes are expresesed as an average case. To remove such problem, the authors proposed an improved Naïve Bayes algorithm. Accuracy is improved in the terms of gap between positive classification accuracy and negative classification accuracy. The gap is reduced to 3.6% when a unigram+bigram was used as a feature. As compared to SVM, the reduced gap is 28.5%.

2.11. Sentiment miner: A prototype for sentiment analysis of unstructured data and text [11]

A model for opinion mining on unstructured text for extracting polarity and for classifying the document at sentence level is presented. The production of enormous unstructured information about different topics makes the task of improvement tracking and monitoring almost unfeasible. This approach is used to find a safe way automatic text analysis and tagging. Sentiment miner will classify the text files by providing features to process at Sentence Level by the use of natural language processing techniques and opinion mining algorithms. 61% accuracy is achieved when five different levels for sentiment orientation are used. But when number of levels of sentiment orientation is reduced accuracy improves. As 75% accuracy is achieved when three levels of sentiment orientation are used.

2.12. Sentiment analysis: Capturing favorability using natural language processing. [12]

This paper presents a sentiment analysis approach to extract sentiments in the association of polarities of positive or negative for particular subjects from a document, in spite of classification of the complete document into positive or negative class. The main problems of sentiment analysis are:

1. Identification of how sentiments are expressed in texts.
2. Whether the expressions indicate favourable or unfavourable opinions toward the subject.

To improve the accuracy of sentiment analysis, it is necessary to appropriately identify the semantic relationships between the sentiment expression and the subject. Semantic analysis is applied with a syntactic parser and sentiment lexicon. In this paper, about 95% precision and roughly 20% recall is achieved in the initial experiment. But the prototype achieved a low precision of 75%, when the domains and datatypes are expanded.

3. Proposed Framework

In this chapter, I have elaborated the broad problem statement (Online Text Mining Model).

- A stream of tweeter comments is extracted for food price crisis in India. The problematic task in sentiment analysis is in sorting the polarity of a certain text at various levels i.e. sentence level, aspect level and document level..
- Whether the stated opinion in a document, a sentence or an entity feature/aspect is +ve, -ve or neutral.

Firstly we have described the architecture of the proposed framework followed by its flowchart. After that we have discussed the other methodology with which we are comparing the performance of our system.

3.1. Online Hybrid Text Mining Model *(OHTM)* : Architecture[10]

In order to detect the sentiments in online text and use these results for the improvement purposes in different contexts such as for improving services by online shopping sites, for the product improvement by different companies, used as reviews for movies and also to extract the impact of government policies etc. This model can be considered as a layered architecture as different tasks are done at different levels. The glimpse of each layer is shown in Figure 1 which shows it contains three modules i.e. Users, Online Text Extractor and Classifiers. The working of these modules is defined below.

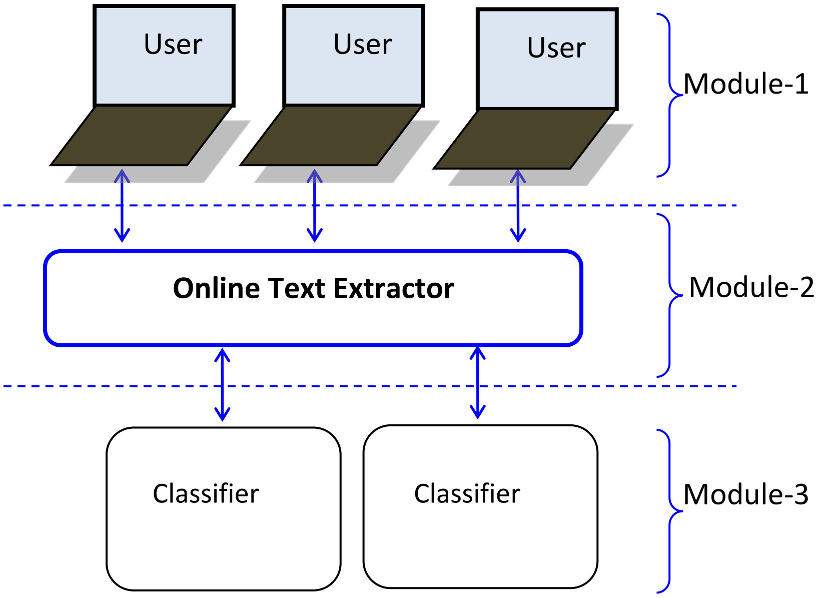


Figure 10 : Online Text Mining Architecture

1. **Users**: Internet users can post their reviews for any topic they are interested in. online users have freedom to express their views and the views inform the owners about the services they are provided. This feedback is used to improve the services by the organizations.

2. **Online Text Mining module**: it the module which extracts the views or comments from the web sites. The views or comments can be extracted as topic based, location based etc. various properties can be applied on the extractor module. In the twitter comments can be extracted on the basis of hashtags etc.

3. **Classifiers**: Different classifiers can be used to classify the text as positive, negative or neutral. In the proposed approach, we have used the KNN classifier and the Naïve Bayes classifier. In the end we compare the results of both the classifiers in terms of accuracy achieved. The other machine learning approaches can also be used to classify the text.

3.2. Online Hybrid Text Mining Model Life Cycle *(OHTMLF)*

The main idea behind Online Text Mining *(OTM)* is to minimize the possibilities of faulty services by any organization. By minimizing faulty services both the resources and time which can occur as overhead in terms of repair can be minimized. *OTM* is mainly design for the computation of the most popular websites in case of online product purchases or as the best policies implemented by the government which are accepted by public and in many more areas in can be proved helpful.

Further *OTM* Life Cycle *(OHTMLF)* works for three modules, i.e. Users, Online text Extractor and Classifiers. (Figure). *(OHTMLF)* combines the Natural Language Processing *(NLP)* techniques with Machine Learning approach to perform the desired task. *(OHTMLF)* starts with the input from the user interface.

Now this user interface defines the application for which the sentiments are going to be calculated. This online user interface can be for any shopping website, can be for any Restaurant website, news channel’s website, any blog and can be any social media network. Next is online text Extractor module. This module defines the methodology which is used to extract comments and reviews from the different sites. For example for extracting tweets from tweeter tweepy app is used. The third module is classifier. Here we can use any classifier used in the Machine Learning techniques for classification. In the proposed approach, we implement KNN and Naïve Bayes classifiers and compare the accuracy of the system using both the systems. Apart from calculating the sentiments *OTM* model can also be useful for extracting features or aspects with respect to particular entity. *OTMLF* is responsible for extracting the comments or reviews, removing the unnecessary data, creating the lexicon resource with respect to the context of the application area and then classifying the comments as Positive, negative or neutral. Both User and Online text extractor comes in the first phase and the classifier comes in the second phase in which any classifier can be implemented. The basic steps of *OHTMLF* are shown in the table.

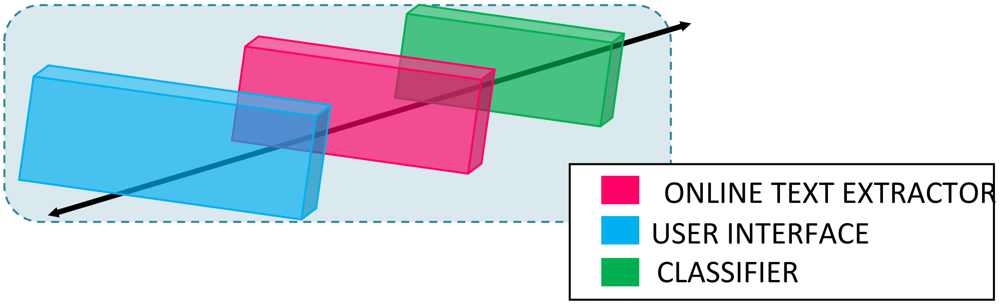


Figure 11 : Online Hybrid Text Mining Model Life Cycle *(OHTMLC)*

In the first step of *OHTMLC*, comments or reviews are posted by the user for a particular topic. In the second step, these comments or reviews are extracted on the basis of topic. Means topic wise extraction is done in this step. In the third step, noisy data is removed by the preprocessing module which includes stemming, removal of stop words, spam words etc. after that a tfd matrix is formed. In the next step, classifier is applied and in the last step overall score is calculated and the performance is measured.

Table 4 OHTMLC Steps

Step	**Input**	**Output**
Comments or Reviews	User personal opinions	Feedback or Review
Text Extraction	Authentication mechanism for extracting comments	Maintaining a topic specific dataset of comments
Preprocessing	Applying Stemming, Stop word removal etc.	Removal of Noisy data
Applying Classifier	KNN Classifier	Classifying the text
Calculating Polarity	Calculate Sentiments	Calculating Overall Score

3.3. OHTM Approach

We classify the OHTM approach in two parts: One is for creating Application Specific Lexicon and second is for classifying text based on the lexicon using classifier.

3.3.1. OHTM for building Lexicon***:*** This approach provides the methodology to build the application specific Lexicon. Application Specific Lexicon is required because some words have different meanings in different contexts. This phase contains the following tasks that are to be performed for building lexicon.

3.3.1.1. *Dataset Extraction:* dataset is extracted from the online sites. As our application includes working with Twitter, so extracting dataset from Twitter includes various attributes to be included.

3.3.1.1.1. Keywords: as the search is done on the basis of topic. So topic specific tweets have to be extracted from the Twitter. Intersection of keyword is done using the different API's available.

3.3.1.1.2. Location: for performing area wise sentiment extraction, extracted tweeters must belong to the specific region. So Location must be specified. We are extracting sentiments for India.

3.3.1.2. Preprocessing: preprocessing of twitters after extracting dataset is necessary. As extracted dataset contains a lot of unnecessary information such as url from which the data is taken, time and date when these comments were published etc. so to remove such type of unnecessary data is the earliest and most important step. Removal of URL's can be done by mentioning the pattern in the regular expressions.

3.3.1.3. Tokenization: extracted and preprocessed data is then divided into tokens. These tokens are nothing but the keywords used to build the lexicon. These steps are performed by using NLTK toolkit.

3.3.1.4. Removal of Short End URLs: these are the URL's which are posted within the review posted by the user for the advertisement purpose.

3.3.1.5. Removal of SPAM Words: a predefined database is used to remove the SPAM keywords which are not allowed to be there in the system because of some banned policies.

3.3.1.6. Creating Dictionary: A dictionary is formed on the basis of term frequency. A matrix is formed with associated scores to these keywords on the basis of how important that word is.

*3.3.2.*OHTM for applying classifier: a classification algorithm or classifier is implemented in this phase. In our purposed approach KNN classifier is applied on the system and a Naïve Bayes classifier is also applied and performance is measured which shows that KNN classifier is much more efficient as compared to the Naïve Bayes classifier.

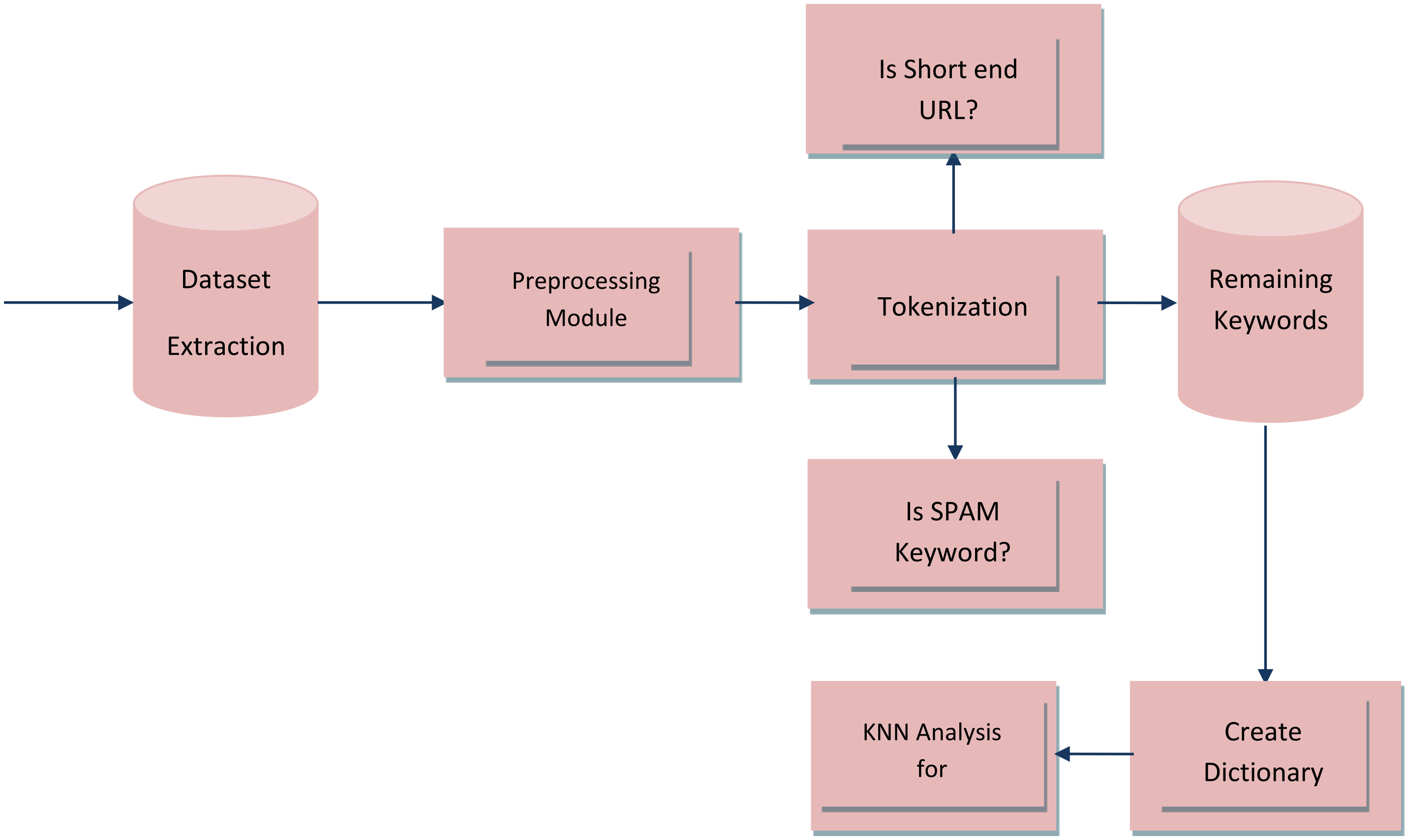


Figure 12: The Hybrid Model

3.4.Algorithm for Lexicon formation

This algorithm builds an application specific lexicon resource which is used by the classifier to classify the text. Basically this algorithm provides the second phase with the training data. As supervised machine learning approaches require training and testing phase.

```
Initial Input :      A set of tweets;
Local variable:      L: A list of tweets.
                     L1: A List Of Spam Keywords
                     $A_i$: Array list for Tokens
                     S: Set of rules
                     TF:Term Frequency
                     IDF:inverse Document Frequency
Output of Phase1:    Training Set (TDM matrix)

Training function {
            a.  Input:   L:A set of extracted tweets for food price;
                /*extracted by data extraction module */

      b.  Preprocessing:
                Repeat For each  tweet in L1
                  Perform:  Convert into Lowercase
                            Perform Stemming/* Extract root words*/
                            Remove Short Words/*Articles*/
                            Remove Conjunctions /*and, then etc */
                            Update L1
      c.  Tokenization:
                Repeat For each tweet in L1
                Perform: store Tokens in arrayList $A_i$ /*Apply Tokenization
                Module (for performing From Sentence Level to Aspect level
                analysis)*/
      d.  Removal of Spam Words :
                Repeat for each Token in $A_i$
```

```
                    Match with L1
                    If(word in L1==Token in $A_i$)
                    {
                    Remove from $A_i$
                    }

e.  Removal of Short end URLs:
                    Repeat for each Token in $A_i$
                    Match with S
                    If(Rule in S==Satisfied)
                    {
                    Remove from $A_i$/*Mark as Short end URL*/
                    }
f.  Form TDM matrix:/*Term Document Frequency*/\
                    For each Token t in $A_{i,}$
                         For each ($A_i$==Token)
                     ▪  Calculate **TF**
                    TF(t) = (Number of times term t appears in a document) /
                    (Total number of terms in the document).
                     ▪  Calculate **IDF**
                        IDF(t) = log_e(Total number of documents / Number of
                        documents with term t in it).
                    TF(t) = (Number of times term t appears in a document) /
                    (Total number of terms in the document).
                     ▪  Create **TDM** matrix and store in database
```

3.5. Algorithm for text classification

Input:

K- no of neighbors

Comments or reviews

Output:

Classify the text as positive, negative or neutral

Step by step algorithm for computing K nearest neighbors KNN algorithm:

1. Determine the parameter K which is number of nearest neighbors
2. Calculate the distance between the tokens of comment and all the training values from the lexicon
3. Sort the distance calculated in last step and find out nearest neighbors based on the K_{th} minimum distance
4. Find the category of the nearest neighbors
5. Simply Using majority of the class of nearest neighbors as the prediction value for the comments.
6. Add the sum and calculate the polarity

Next we are representing the flowchart for the phase 1 and phase 2.

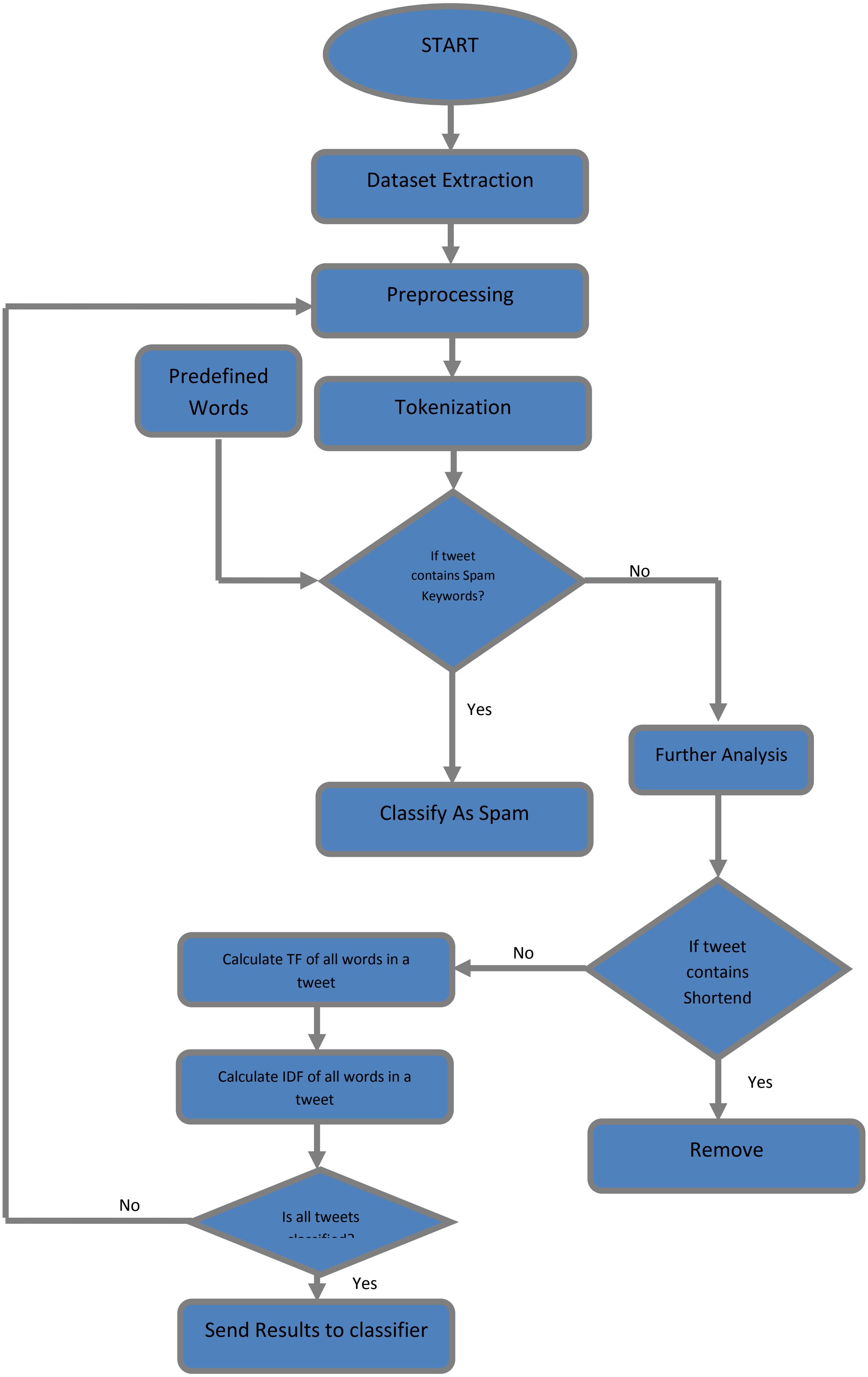

START
Dataset Extraction
Preprocessing
Predefined Words
Tokenization
If tweet contains Spam Keywords?
No
Yes
Further Analysis
Classify As Spam
If tweet contains Shortend
No
Yes
Calculate TF of all words in a tweet
Calculate IDF of all words in a tweet
Remove
No
Is all tweets classified?
Yes
Send Results to classifier

Figure 13 : Flowchart for Phase 1

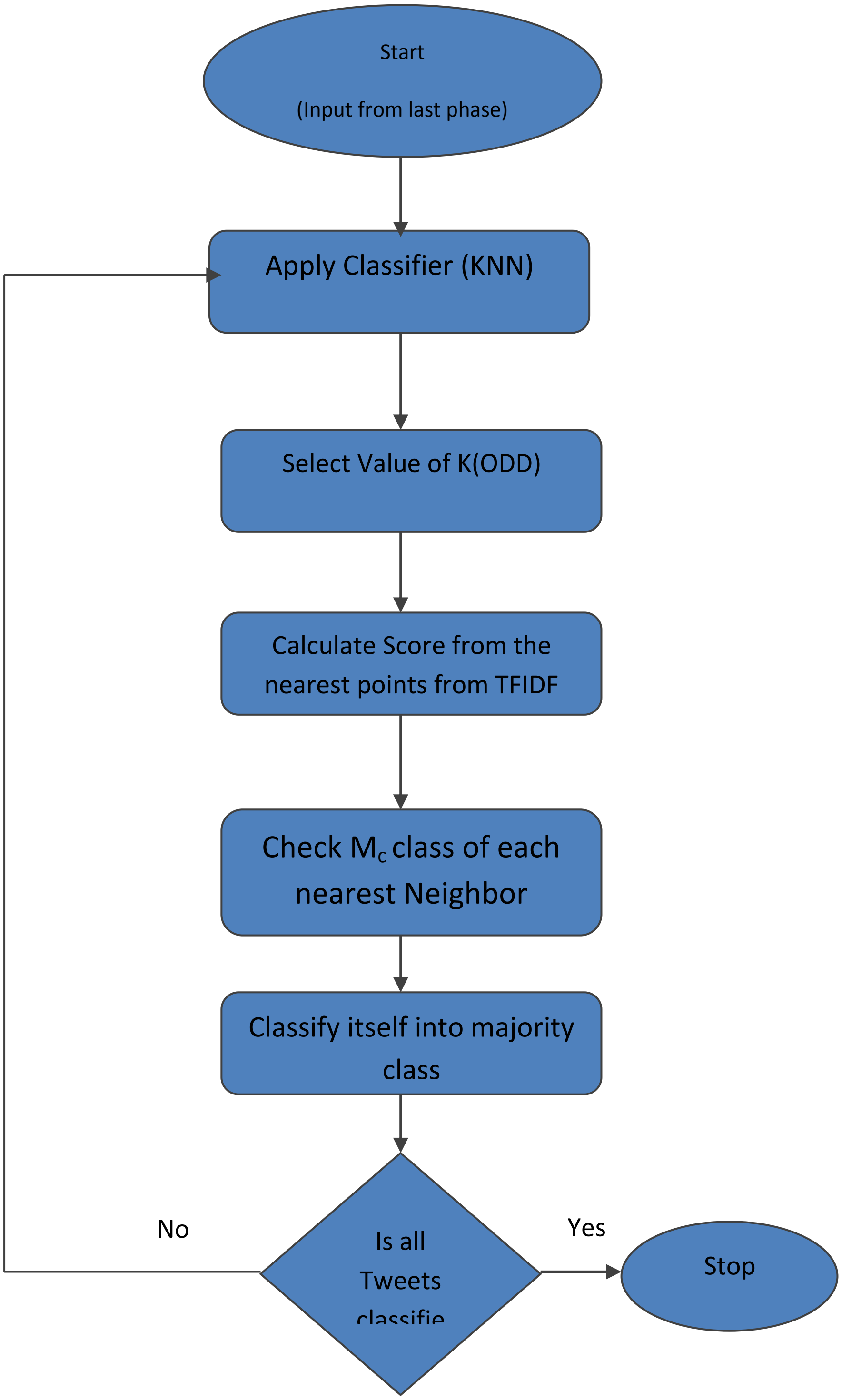

Start
(Input from last phase)
Apply Classifier (KNN)
Select Value of K(ODD)
Calculate Score from the nearest points from TFIDF
Check M_c class of each nearest Neighbor
Classify itself into majority class
No
Is all Tweets classifie
Yes
Stop

Figure 14 : A flowchart for the Phase 2

3.6. Context for OHTM model: Mining Indian Tweets to Understand Food Price Crisis

Food prices in country impose a direct impact on the purchasing power of population. Despite, India has seen impressing economic growth in last few years; the country is still struggling with widespread poverty and hunger. From world's total hungry population, India points to 25% of the total. According to the international food policy research institute, India is at 55th position as per 2014 global hunger index. The consumer price index means estimation of price changes in India is increased to 145.20 index points in October of 2014 from a145 index points in September. CPI in India averaged 125.20 index points for 2011 until 2014, reaching an all-time high of 145.20 index points in October 2014, according to the ministry of statistical and program implementation (MOSPI), India. (Fig. 4) The poor and vulnerable were significantly left behind. Rising food prices would further undermine the food security and livelihoods of the most vulnerable by eroding their already limited purchasing power. Poor people spend 60 to 70% of their income on food and they have little capacity to adapt as prices and wages may not adjust accordingly. Thus the situation in India can still pose a threat to food and nutrition security of the country. Food price rise impacts a lot of sections in the country. One of them is increase in poverty and many more.

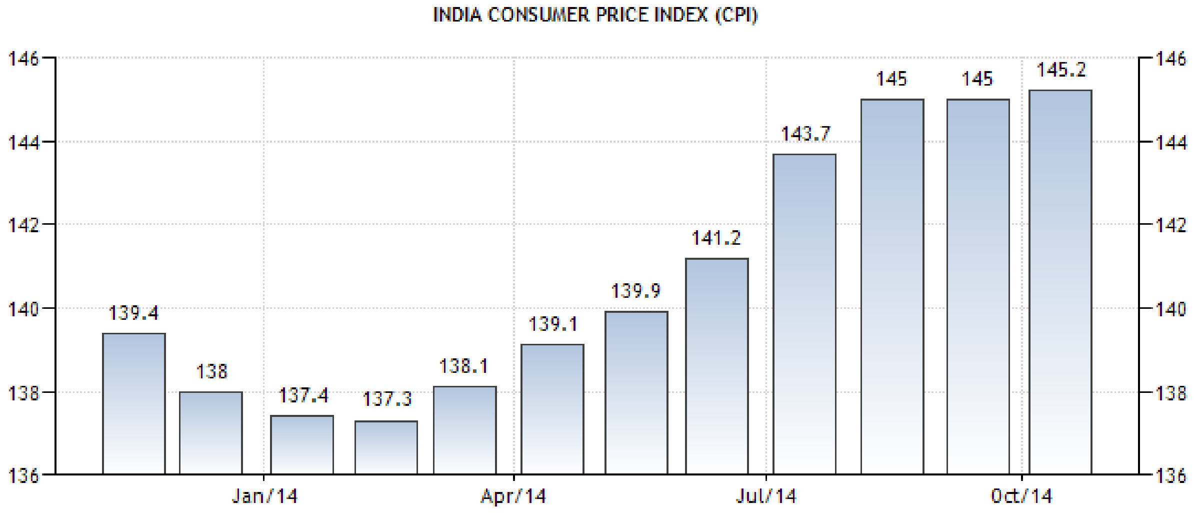


Figure 15: India Consumer Price

As the use of social media is increasing day by day, consumers react toward government actions. Total 110 million users in India use social networking sites. The analysis of consumer behavior thus become the inevitable and critical part of the overall planning and decision making functions for any organization that is helpful to match the core competencies and capabilities.

This research work will help to monitor the food security in the country. Analysis of twitter conversations related to food price increases amongst Indians. This will represent a new source of information and also helps to explore the new relationships between such conversations, food price inflation and external trends. Table 2 is representing the global monthly food price indices by FAO i.e. Food and Agricultural Organizations of United Nations.

Table 5: MONTHLY FOOD PRICE INDICES (2012-2015)[50]

Date	Food Price Index	Meat Price Index	Dairy Price Index	Cereals Price Index	Oils Price Index	Sugar Price Index
1/2012	214.7	181.4	211.3	217.7	235.0	334.3
2/2012	217.2	184.2	207.1	220.9	240.8	342.3
3/2012	217.0	183.7	200.3	221.9	247.4	341.9
4/2012	213.9	185.0	187.9	218.6	253.4	324.0
5/2012	205.0	180.0	178.4	216.2	233.4	294.6
6/2012	200.6	174.0	176.2	217.0	219.2	290.4
7/2012	213.0	172.4	176.3	253.0	224.4	324.3
8/2012	213.8	177.7	181.7	253.6	223.6	296.2
9/2012	217.8	184.5	195.4	255.4	221.4	283.7
10/2012	217.1	187.2	201.0	254.3	201.9	288.2

11/2012	215.6	186.5	202.8	255.2	195.8	274.5
12/2012	213.8	187.2	204.9	249.1	190.7	274.0
1/2013	212.9	184.3	208.5	244.0	200.3	267.8
2/2013	212.6	186.4	209.7	241.1	201.8	259.2
3/2013	214.8	185.2	228.8	240.5	196.7	262.0
4/2013	216.9	186.6	258.8	230.7	194.0	252.6
5/2013	214.6	180.0	253.5	234.8	194.3	250.1
6/2013	211.9	179.7	246.2	232.3	193.5	242.6
7/2013	207.5	179.4	243.6	222.3	186.7	239.0
8/2013	204.5	182.4	247.6	206.8	181.8	241.7
9/2013	203.7	186.1	250.2	195.0	184.3	246.5
10/2013	206.6	187.3	251.1	196.6	188.0	264.8
11/2013	205.7	185.7	250.8	194.3	198.5	250.6
12/2013	206.2	185.6	264.1	192.9	196.0	234.9
1/2014	203.2	182.2	267.7	191.4	188.6	221.7
2/2014	208.6	181.8	275.4	198.6	197.8	235.4
3/2014	213.8	185.5	268.5	208.9	204.8	254.0
4/2014	211.5	190.4	251.5	209.2	199.0	249.9
5/2014	210.4	194.6	238.9	207.0	195.3	259.3
6/2014	208.9	202.8	236.5	196.1	188.8	258.0
7/2014	204.3	205.9	226.1	185.2	181.1	259.1
8/2014	198.3	212.0	200.8	182.5	166.6	244.3
9/2014	192.7	211.0	187.8	178.2	162.0	228.1
10/2014	192.7	210.2	184.3	178.3	163.7	237.6
11/2014	191.3	206.4	178.1	183.2	164.9	229.7
12/2014	185.8	196.4	174.0	183.9	160.7	217.5
1/2015	178.9	183.5	173.8	177.4	156.0	217.7
2/2015	175.8	176.9	181.8	171.7	156.6	207.1
3/2015	173.1	175.0	184.9	169.8	151.7	187.9
4/2015	171.0	178.0	172.4	167.6	150.2	185.5

Food prices in country have a straight impact on the purchasing power of a large amount of Indian population. In spite of impressive economic growth in last time span, a large portion of population is still under attack of scarcity and starvation. Some of the facts are given by World Food Programme [1] is an estimated 32.7% of the total population lives on less than US$ 1.25 per day. According to UNDP Human Development Index 2014, India is on 135th position among 187 countries and on 55th position among 76 countries in Global Hunger Index. [2] India FoodBanking Network states that all of the world, India is largest residence of leading undernourished and hungry population. About 1/6th of our total population is undernourished. Almost 190 million citizens walk off starving daily and many more such facts are there which need quick actions to compete with poverty, hunger and also to increase food security across the country. Objective of this research is to extract hidden issues which can help to monitor food security across the country. Operating on the principle that conversations over social media can stand for a new source of information to supervise food security. Sentiment analysis of such conversations can provide the feedback to all the policies which are applied by Indian government and many more facts. Sentiment analysis deals with a lot of challenges such as subjectivity analysis, finding blind negation, recognising sarcasm etc.

Sentiment analysis of the data set containing relevant data to food price rise helps to extract factors that affect food price and effects of food price rise on population. This research will also help to analyze the actions that the government has taken to decline the food prices across the country and explore the other factors that can help to reduce the food prices.

We can found a relationship been official food inflation and number of tweets speaking about food price increases. We can further find the relationship with other commodities or fields. This research work will try to automate the monitoring of public sentiments on social media, combined with contextual knowledge, and has the potential to be a real time proxy for food-related economic indicators. This automation will also help to uncover the public's reaction towards government actions. Main challenges are to collect data related to food prices rise. Topic filtering will be the one of the step in this research work. After that context based sentiment extraction will be done.

The major steps which need to be fulfilled in this thesis are given below:

- Development of a framework to extract tweets from twitter
- Streaming of tweets using Tweepy and creation of a database
- Preprocessing of the tweets using various NLP pre-processing steps
- Creation of a dictionary of important terms and tweet id
- Creation of TDM matrix
- Application of Classification algorithm
- Comparative analysis of performance

Our model combines NLP and Machine Learning techniques to classify the text. NLP techniques are used in preprocessing step which is compulsory in every field to make the data noise free.

3.7.Objective of our research

- Extraction of tweeter dataset for food prices in India
- Implementation of the system using hybrid approach
- Comparing the system with Naïve Bayes approach

3.8.Benefits

- To monitor the food security in the country.
- Analysis of twitter conversations related to food price increases amongst Indians.
- Helps to explore the new relationships such as food price inflation and external trends.

3.9.Performance Matrices

Performance of selected classifier for our work is compared from the given metric described below:

a) Accuracy: Accuracy is the whole correctness of the model. It is calculated as the sum of correct classifications of class x divided by the total number of classifications of class x. It is defined as:

$$\text{Accuracy}(\text{x}) = \left(\frac{\text{sum of correct classification}}{\text{total number of classification}}\right) \times 100$$

Equation 5

b) Precision: Precision is the proportion of the examples which truly have class x among all those which are classified as class x. This is defined as:

$$\text{Precision}(\text{x}) = \left(\frac{\text{number of correctly classified instances of class x}}{\text{number of instances classified as belonging to class x}}\right) \times 100$$

Equation 6

c) Recall: It is a measure of the ability of a prediction model to select instances of a certain class from a data set also called as sensitivity and corresponds to the true positive rate. It is defined as:

$$\mathrm{Recall}(x) = \left(\frac{\text{numbers of true positive predictions}}{\text{numbers of true positive predictions} + \text{numbers of false negative predictions}}\right)100$$

Equation 7

3.10. Hardware /Software Requirements

Hardware Requirements

- Processor: Intel core 2 Duo with CPU clock rate 2.10 GHZ.
- RAM : Memory two DDR2 with 3 GB
- HDD: SATA with capacity of 500 GB.

Software Requirements

- Operating System : Microsoft 7 x86
- Python: Python version 3.4 is used
- MS Word: MS Word is used for documentation purpose.
- MS Excel: MS Excel is used to generate bar graphs and .csv files for classifier

4. Implementation

This chapter elaborates the implementation of proposed Online Hybrid Text Mining Model *(OHTM)*. The various text mining algorithm and streaming of twitter api are given in this chapter. The first step starts with the extraction of tweets followed by preprocessing of the extracted tweets. Then Classifier algorithm has to be applied on it.

Thus, I have developed Python based modules to implement the proposed framework. In order to firstly implement the *OHTM* for building lexicon I have developed the module to extract the tweeter comments based on the keywords and location.

The twitter API named as 'tweepy' has been used in this thesis for the extraction step. The major steps involved in development of the framework for live streaming of tweets begin with setting up an account on twitter.

4.1. Implementation of OHM-Lexicon Creation

4.1.1.Setting up the twitter account

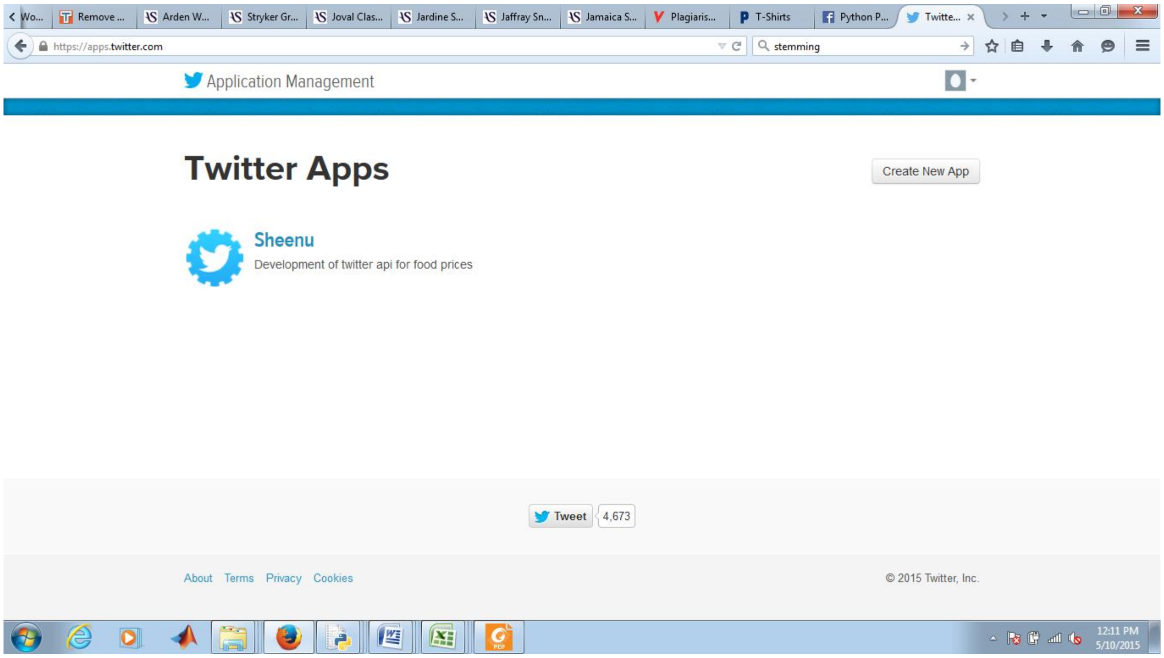
Following steps are followed to set up a twitter account and snapshots are given for step by step registration process for creating an app for accessing twitter comments. An app named 'Sheenu' is created. This will be utilized for streaming.

- Set up an account on twitter
- Go to dev.twitter.com
- Create a new app and register for it
- Change access level to Read, write and access messages
- Generate security id and secret number
- Generate access token id and secret token number
- Save them to be utilized for streaming
- OAuth handler is used for streaming the tweets.

4.1.2. Filtrations settings

Filters are applied on it using the track filter. The tweets are filtered by two ways.

- Filter by content
- Filter by location



Application Management

Twitter Apps

Create New App

Sheenu

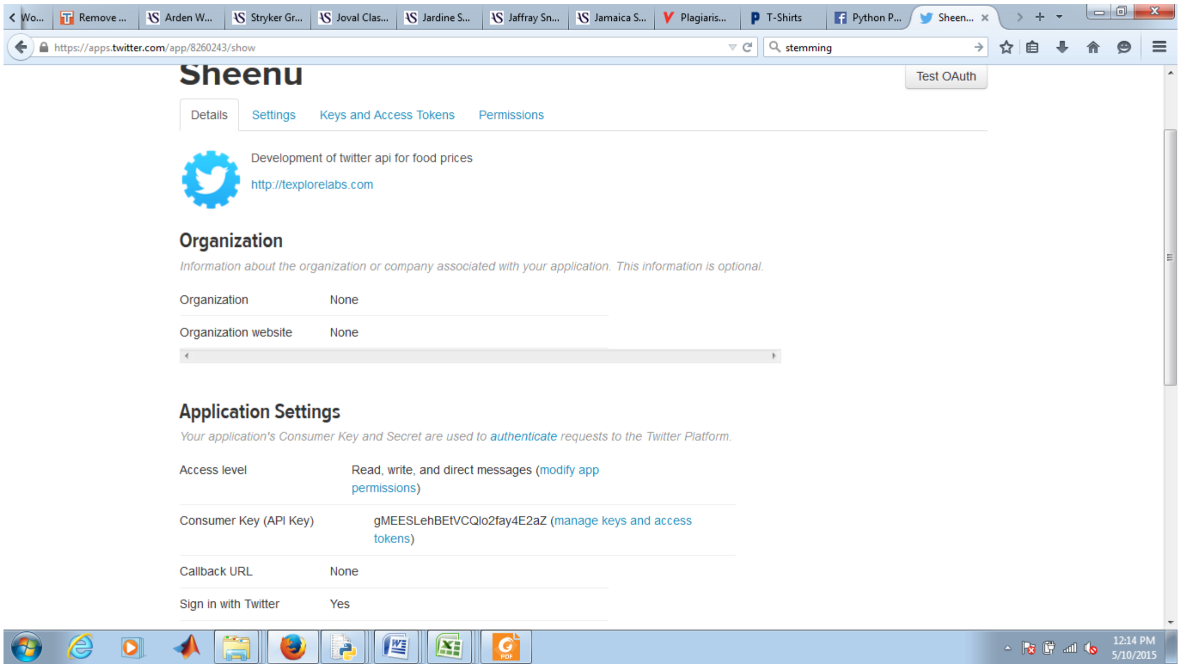
Development of twitter api for food prices

Tweet 4,673

About Terms Privacy Cookies

© 2015 Twitter, Inc.

Figure 16 : Setting up account for twitter app



Sheenu

Test OAuth

Details Settings Keys and Access Tokens Permissions

Development of twitter api for food prices

http://texplorelabs.com

Organization

Information about the organization or company associated with your application. This information is optional.

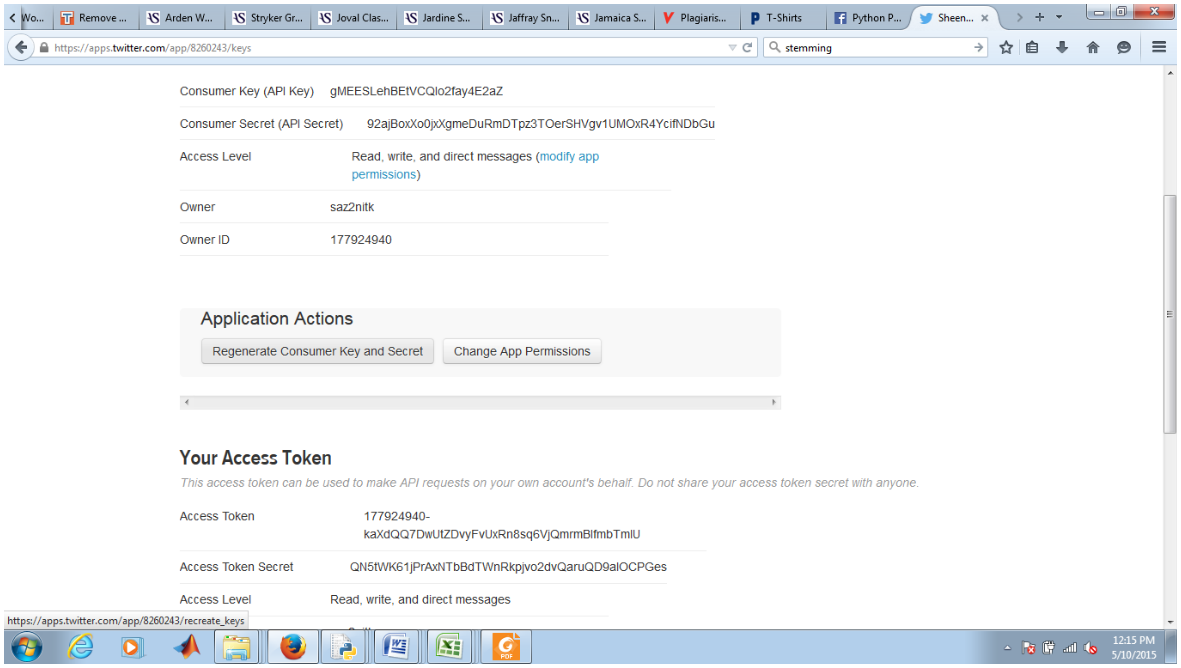
Organization	None
Organization website	None

Application Settings

Your application's Consumer Key and Secret are used to authenticate requests to the Twitter Platform.

Access level	Read, write, and direct messages (modify app permissions)
Consumer Key (API Key)	gMEESLehBEtVCQlo2fay4E2aZ (manage keys and access tokens)
Callback URL	None
Sign in with Twitter	Yes

Figure 17: Getting the access tokens and authentication



Consumer Key (API Key)	gMEESLehBEtVCQlo2fay4E2aZ
Consumer Secret (API Secret)	92ajBoxXo0jxXgmeDuRmDTpz3TOerSHVgv1UMOxR4YcifNDbGu
Access Level	Read, write, and direct messages (modify app permissions)
Owner	saz2nitk
Owner ID	177924940

Your Access Token

This access token can be used to make API requests on your own account's behalf. Do not share your access token secret with anyone.

Access Token	177924940-kaXdQQ7DwUtZDvyFvUxRn8sq6VjQmrmBlfmbTmlU
Access Token Secret	QN5tWK61jPrAxNTbBdTWnRkpjvo2dvQaruQD9alOCPGes
Access Level	Read, write, and direct messages

Figure 18: Access Tokens

4.1.3. Filter by content

Due to the policies of twitter the filtering is not absolutely correct and there might be a similar tweet which doesn't lie in the filtered bandwidth. The content filtering is done using the following keywords:

- Food price Crisis
- Food price
- Food Security
- Inflation
- Vegetable Prices
- Tomato Prices
- Onion prices
- Price Rise
- Onion Price

4.1.4.Filter by Location

The location is done using a 'location ' filter available with tweepy. The location filter works on the basis of latitude and longitude of the place. A bounding box has to be formed in which the location filter works. Any tweets sent from that bounding box is streamed.

This thesis has utilized the following settings.

South West Longitude=73 degrees

South West Latitude=15 degrees

North East longitude=85 degrees

North East Latitude=27 degrees

Using these settings the tweets are extracted and saved in a database.Text mining is applied on the filtered tweets for further processing.

Further to implement the *OHTM-Classifier* approach I have simulated the KNN model and Naïve Bayes model over PYTHON platform. To implement these interfaces we have used following software specifications (shown in Table 6):

Table 6: Software specification for OHTM

Sr. No.	Specification	Description
1	Platform	Python 3.1
2	Programming Language	Python
3	Development Tool	Python idle or shell, or jetbrains pycharm
4	Operating System	Windows 7

A consumer Key is generated. Next figure shows the Keys generated which will be used for streaming. Four Keys are required. For this thesis , Consumer Key, Consumer Secret, Access Token and Access token Secret is shown in Fig. These Keys are utilized and streaming is done as shown in the fig below.

4.2.Steps of OHTM Model: phase 1

Pre-processing steps on textual description of bug reports are performed. It includes tokenization, stop word removal and stemming. Tokenization divides textual description into tokens by removing punctuation marks. Then stop words are performed that remove unnecessary information (conjunctions, interjections and articles) from datasets. Stemming on reduced datasets are performed to reduced terms into their root terms. Porter's stemming algorithms are used to perform stemming.

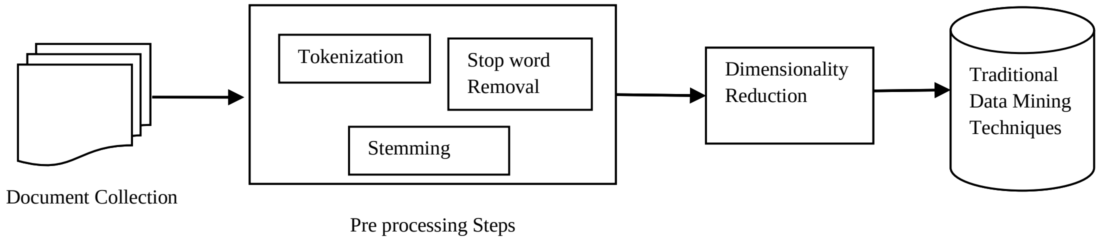


Figure 19: Text Mining

Steps in text mining: The different steps performed in text mining are as follows:

Preprocessing- It is used to distill unstructured data to structured format. There are different preprocessing steps performed in Text mining such as tokenization, stop word removal and stemming. These algorithms are discussed below.

i. **Tokenization:** The purpose of tokenization is to remove all the punctuation marks like commas, full stop, hyphen and brackets. It divides the whole text into separate tokens to explore the words in document.

ii. **Stop word removal:** The purpose of this process is used to eliminate conjunction, prepositions, articles and other frequent words such as adverbs, verbs and adjectives from textual data. Thus it reduces textual data and system performance is improved.

iii. **Stemming**: Stemming is used to reduce the words to their root words e.g. words like ”computing” ,”computed” and “computerize” has it root word “compute”. The purpose of stemming is to represent the words to only terms in their document. There are different algorithms to perform stemming such as Lovins Stemmer, Porters Stemmer, Paice/Husk Stemmer ,Dawson Stemmer, N-Gram Stemmer, YASS Stemmer and HMM Stemmer.

iv. **Weighting Factor**: - Features are extracted from overloaded large datasets.TF-IDF (Term frequency- Inverse document frequency) [47] score is generally is used to give weight to each term. TF-IDF is multiplication of term frequency and inverse document frequency.

$$\mathrm{TF-IDE} = n_w^d \, log_2(\frac{N}{N_w})$$

Equation 8

Where n_w^d = frequency of word w in document d.

N= total document and N_w= document congaing word w.

v. Term - document matrix – After initial steps of preprocessing text in documents is converted into term- document matrix. Rows in matrix represents document in which word appears and columns represent the words that are extracted from documents. The cell of matrix is filled with TF-IDF score.

Classification, clustering and predictive methods are applied to the reduced datasets using data mining techniques to analyze the pattern and trends within data.

vi. **Term -Document matrix**- The Term- Document Matrix (TDM) is created. Each column in matrix represents the terms occurring in documents and row represents id of each bug report. The cells of matrix are filled with TF-IDF score. If term is not present in the particular bug reports then cell is filled with zero.

vii. **Dimensionality Reduction** – After preprocessing steps, dimensionality reduction is performed. Here original TDM (term document matrix) is replaced with smaller matrix by using a SVD (singular value decomposition technique). This technique discards unimportant word and relevant and important word are filtered out. The new matrix is generated of terms and documents.

viii. **Feature Selection-** Feature selection methods are used to retrieve the most informative terms from corpus of datasets. In our research, we have used two feature selection methods info gain and CHI square methods. These methods are applied on TDM matrix to reduce matrix.

ix. **Creating dictionary of terms-** The terms obtained after applying feature selection are sorted in descending order according to their weights. The top m- terms are used for creating dictionary. The dictionary contains the terms that help in specifying the severity levels of each bug report.

4.3.Machine Learning Approaches

There are various approaches to design machine learning algorithms. The purpose of ML algorithms is to use observations as input and this observation can be a data, pattern and past experience. Thus Ml algorithms use to improve the performance of instances, which can be done by any classifier by trying to classify the input pattern into set of categories or to cluster unknown instances. As the nature of ML algorithms it enhances its performance from past experience or by receiving feedback. It can be divided into two categories supervised and unsupervised approach [49].

Supervised: In supervised learning, the instances are labeled with known or target classes labels. Here before classification the dataset knows the target class. Thus it is very helpful for the problems which have known inputs.

Unsupervised: In unsupervised learning, the algorithm groups the instances by their similarities in values of features and makes different clusters. In it no prior class or clusters are given, the algorithm itself defines their clusters automatically and statistically.

4.5.1.KNN Algorithm

KNN is type of instance based learning or lazy learning. In this learning the function is approximately locally and all computation is deferred until classification. It is simplest of all machine learning algorithms. In KNN classification, the output is class membership. An object is classified by majority votes of its neighbors by the object being assigned to class most common among its k nearest neighbor (k is positive small integer). The nearest neighbor is determined using similarity measure usually distance functions are user. Following are the distance function used by KNN [17].

Euclidean distance function $\sqrt{\sum_{i=1}^{N}(a_i - b_i)^2}$

Manhattan distance function $\sum_{i=}^{N}|a_i - b_i|$

Where {(a_1, b_1), (a_2,b_2) ,(a_3,b_3)…… (a_N, b_N)} is training datasets.

In KNN algorithm all the distance from testing point to training point are computed. Then these all testing points are sorted ascending order. Then class labels are added for each K nearest neighbors and sign of sum are used for prediction. The value of k in k-nearest neighbor is challenging task. As choosing smaller value of k. e.g. by choosing k=1 may lead to risk of over fitting and choosing larger value of k e.g. k=N may lead to under fitting. Therefore optimal value of k has been chosen between the values 3-10, which gives better result.

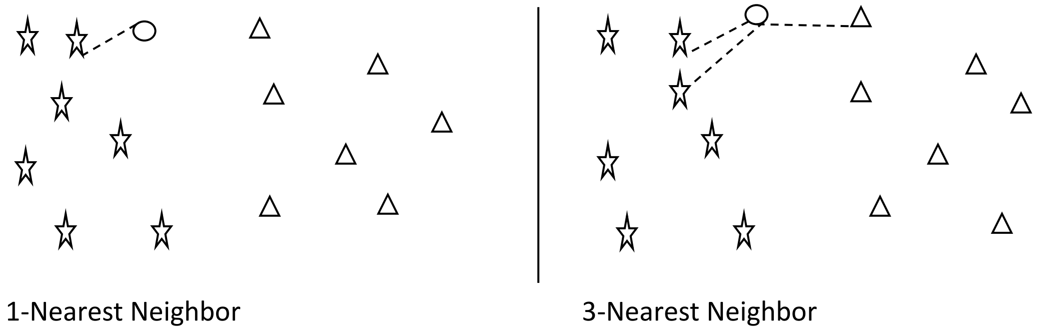


Figure 20: Working of KNN Algorithm

Algorithm: KNN (D, k, $\hat{x}$)

D is training dataset, N training examples are paired as (x_1, y_1), (x_2, y_2) … (x_N, y_N).

[] $\longrightarrow$ an empty list and $\oplus$ $\longrightarrow$ used to append in list.

Prediction on $\hat{x}$ (testing data point) is called $\hat{y}$

```
1.   S $\longleftarrow$ []
```

```
    for n=1 to N do
        $S \longleftarrow S \oplus \langle d(x_n, \hat{x}), n \rangle$        // store distance to training example n
    end for
    $S \longleftarrow$ SORT(S)        // put lowest-distance objects first
    $\hat{y} \longleftarrow 0$
    For K=1 to K do
        $\langle dist, n \rangle \longleftarrow S_K$        // n this is the kth closest data point
        $\hat{y} \longleftarrow \hat{y} + y_n$        // vote according to the label for the nth training point
    end for
    return SIGN($\hat{y}$)        // return +1 if $\hat{y} > 0$ and 1 if $\hat{y} < 0$
```

Application of K nearest neighbor

1. Nearest Neighbor based Content Retrieval- It is one of the important applications of K-Nearest neighbor e.g. if the content is video and it is used for retrieving videos that is closest to given video [18].
2. Protein-Protein interaction and 3D structure prediction- KNN is used to predict the structure of Gene and graph based KNN is used to predict the interaction of protein.

4.5.2.Naïve Bayes Algorithm

The algorithm is named after famous statistician Thomas Bayes who proposed Bayesian theorem. The Naïve bayes algorithm is also based on Bayesian theorem. This theorem assumes that all the attributes are conditionally independent to each other. In this algorithm, conditional probability for each attribute with respect to certain class level is calculated. The new document is classified using sum of probabilities for each class [17] . The classifier is easy to build and useful when there is large datasets.The classification framework is briefly discussed as follows:

Suppose we have D set of tuples and each tuple has attribute vector X(x1, x2, x3 , ... xn) of n dimensions. Let there are k number of classes C1, C2, C3... Ck. The classifier predicts X belongs to Ci if

P $(\frac{C_i}{X})$ = P $(\frac{C_j}{X})$ for 1<=j <=k, j <> i.. (1)

Posterior probability is calculated as

P $(\frac{C_i}{X}) = \frac{P\ (X/Ci)\ \ P(c_i)}{P(X)}$.. (2)

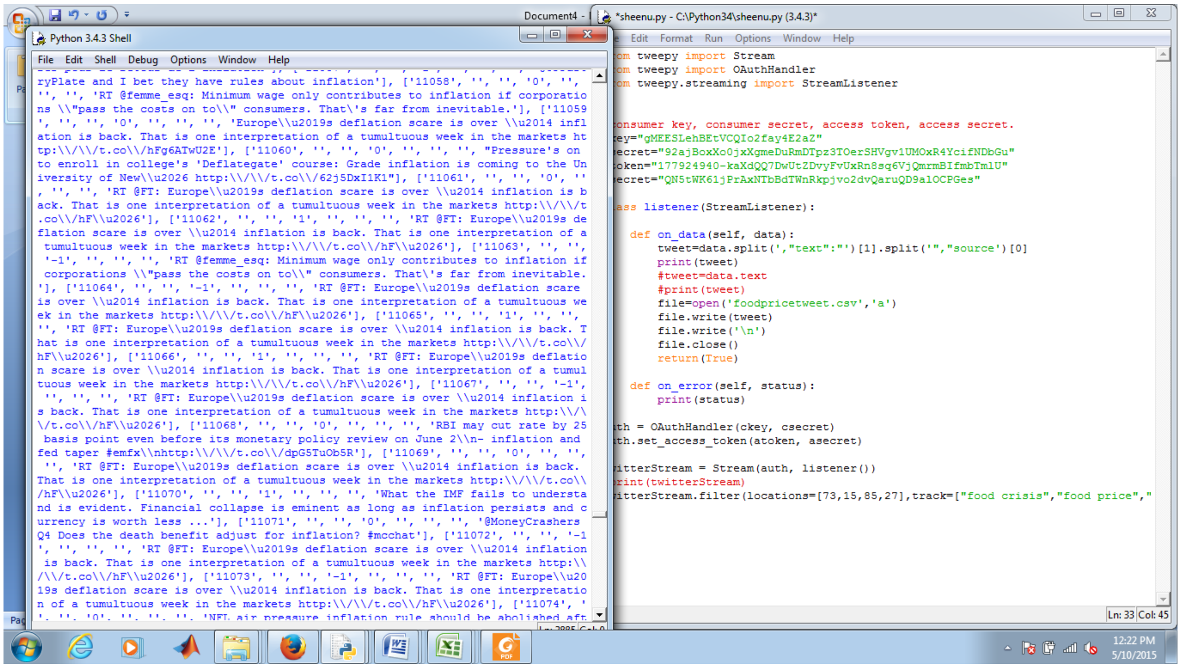
Application of Naïve Bayes

1. Text Classification- The classifier is well known for its most efficient learning capability for classification of text document [16].
2. Spam filtering –Spam filtering makes use of the classifier to identify spam mails and filter out them from legitimate mail. E-mail filter such as SpamBayes, SpamAssassin and Bogofilte are example of filter that uses Naïve bayes classifier.
3. Hybrid Recommender System- It is proposed a unique switching hybrid recommendation approach by combining a Naïve Bayes classification approach with the collaborative filtering.

The results of the proposed method will be shown and discussed in the next chapter. The methodology has been designed for development of a framework to automatically provide public feedback for decision making regarding food related issues in India. The rise in food prices has been dealt with the public opinion regarding this will be shown in next Chapter.

5. Results

This chapter elaborates the results which are obtained by applying the proposed model. Results for both the KNN and Naïve Bayes are calculated and compared. The accuracy of KNN algorithm is much more accurate.


Document4 -
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
sheenu.py - C:\Python34\sheenu.py (3.4.3)
e Edit Format Run Options Window Help
Ln: 33 Col: 45
12:22 PM
5/10/2015

```
ryPlate and I bet they have rules about inflation'], ['11058', '', '', '0', '',
'', '', 'RT @femme_esq: Minimum wage only contributes to inflation if corporatio
ns \\"pass the costs on to\\" consumers. That\'s far from inevitable.'], ['11059
', '', '', '0', '', '', '', 'Europe\\u2019s deflation scare is over \\u2014 infl
ation is back. That is one interpretation of a tumultuous week in the markets ht
tp:\\/\\/t.co\\/hFg6ATwU2E'], ['11060', '', '', '0', '', '', '', "Pressure's on
to enroll in college's 'Deflategate' course: Grade inflation is coming to the Un
iversity of New\\u2026 http:\\/\\/t.co\\/62j5DxI1K1"], ['11061', '', '', '0', ''
, '', '', 'RT @FT: Europe\\u2019s deflation scare is over \\u2014 inflation is b
ack. That is one interpretation of a tumultuous week in the markets http:\\/\\/t
.co\\/hF\\u2026'], ['11062', '', '', '1', '', '', '', 'RT @FT: Europe\\u2019s de
flation scare is over \\u2014 inflation is back. That is one interpretation of a
 tumultuous week in the markets http:\\/\\/t.co\\/hF\\u2026'], ['11063', '', '',
 '-1', '', '', '', 'RT @femme_esq: Minimum wage only contributes to inflation if
 corporations \\"pass the costs on to\\" consumers. That\'s far from inevitable.
'], ['11064', '', '', '-1', '', '', '', 'RT @FT: Europe\\u2019s deflation scare
is over \\u2014 inflation is back. That is one interpretation of a tumultuous we
ek in the markets http:\\/\\/t.co\\/hF\\u2026'], ['11065', '', '', '1', '', '',
'', 'RT @FT: Europe\\u2019s deflation scare is over \\u2014 inflation is back. T
hat is one interpretation of a tumultuous week in the markets http:\\/\\/t.co\\/
hF\\u2026'], ['11066', '', '', '1', '', '', '', 'RT @FT: Europe\\u2019s deflatio
n scare is over \\u2014 inflation is back. That is one interpretation of a tumul
tuous week in the markets http:\\/\\/t.co\\/hF\\u2026'], ['11067', '', '', '-1',
 '', '', '', 'RT @FT: Europe\\u2019s deflation scare is over \\u2014 inflation i
s back. That is one interpretation of a tumultuous week in the markets http:\\/\
\/t.co\\/hF\\u2026'], ['11068', '', '', '0', '', '', '', 'RBI may cut rate by 25
 basis point even before its monetary policy review on June 2\\n- inflation and
fed taper #emfx\\nhttp:\\/\\/t.co\\/dpG5TuOb5R'], ['11069', '', '', '0', '', '',
 '', 'RT @FT: Europe\\u2019s deflation scare is over \\u2014 inflation is back.
That is one interpretation of a tumultuous week in the markets http:\\/\\/t.co\\
/hF\\u2026'], ['11070', '', '', '1', '', '', '', 'What the IMF fails to understa
nd is evident. Financial collapse is eminent as long as inflation persists and c
urrency is worth less ...'], ['11071', '', '', '0', '', '', '', '@MoneyCrashers
Q4 Does the death benefit adjust for inflation? #mcchat'], ['11072', '', '', '-1
', '', '', '', 'RT @FT: Europe\\u2019s deflation scare is over \\u2014 inflation
 is back. That is one interpretation of a tumultuous week in the markets http:\\
/\\/t.co\\/hF\\u2026'], ['11073', '', '', '-1', '', '', '', 'RT @FT: Europe\\u20
19s deflation scare is over \\u2014 inflation is back. That is one interpretatio
n of a tumultuous week in the markets http:\\/\\/t.co\\/hF\\u2026'], ['11074', '
', '', '0', '', '', '', 'NFL air pressure inflation rule should be abolished aft
```

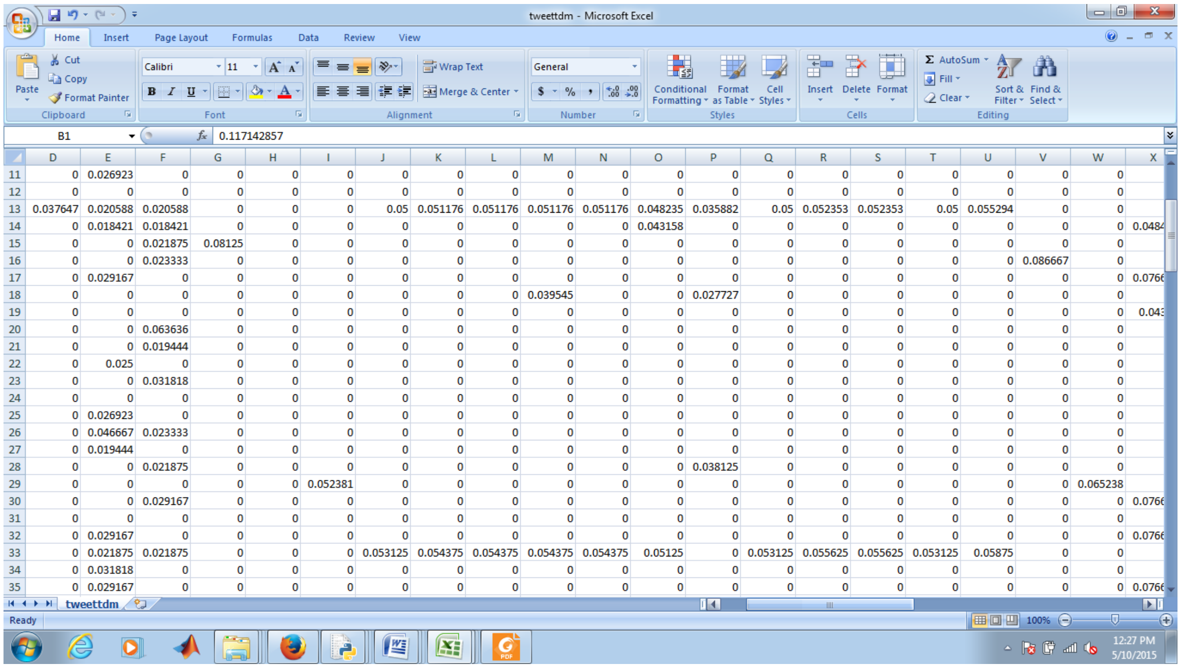
```
om tweepy import Stream
om tweepy import OAuthHandler
om tweepy.streaming import StreamListener


onsumer key, consumer secret, access token, access secret.
ey="gMEESLehBEtVCQIo2fay4E2aZ"
ecret="92ajBoxXo0jxXgmeDuRmDTpz3TOerSHVgv1UMOxR4YcifNDbGu"
oken="177924940-kaXdQQ7DwUtZDvyFvUxRn8sq6VjQmrmBIfmbTmlU"
ecret="QN5tWK61jPrAxNTbBdTWnRkpjvo2dvQaruQD9alOCPGes"

ass listener(StreamListener):

   def on_data(self, data):
       tweet=data.split(',"text":"')[1].split('","source')[0]
       print(tweet)
       #tweet=data.text
       #print(tweet)
       file=open('foodpricetweet.csv','a')
       file.write(tweet)
       file.write('\n')
       file.close()
       return(True)

   def on_error(self, status):
       print(status)

th = OAuthHandler(ckey, csecret)
th.set_access_token(atoken, asecret)

itterStream = Stream(auth, listener())
rint(twitterStream)
itterStream.filter(locations=[73,15,85,27],track=["food crisis","food price","
```

Figure 21: Extraction of Twitter comments

Figure 23 represents the extracted dataset by the twitter using tweepy API. Authentication key and access tokens are used to gain the access and work as the authorized user at the other end. These tweets are saved in a database and sentiment values are assigned to them based on manual interpretation. The sentiments are assigned as follows.

- ‘1’ for positive sentiment
- ‘2’ for negative sentiment
- ‘o ’ for neutral sentiment

An array of the tweets is created and term document matrix is created using TFIDF score as shown below.

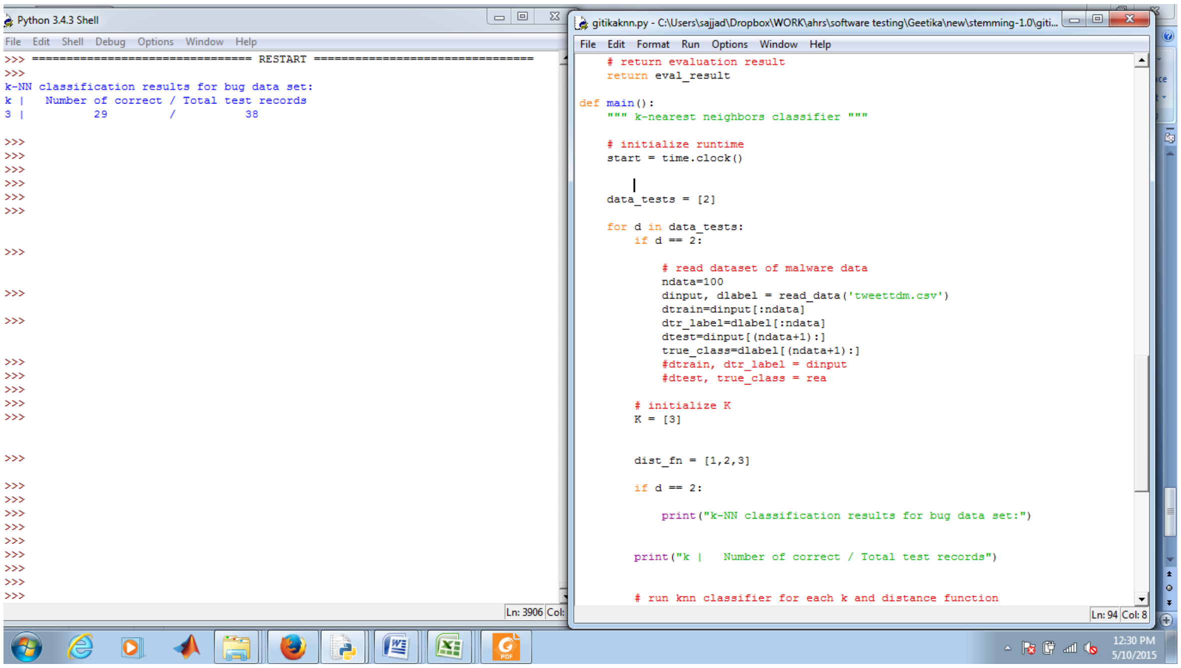


B1 0.117142857

	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X
11	0	0.026923	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
12	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
13	0.037647	0.020588	0.020588	0	0	0	0.05	0.051176	0.051176	0.051176	0.051176	0.048235	0.035882	0.05	0.052353	0.052353	0.05	0.055294	0	0	
14	0	0.018421	0.018421	0	0	0	0	0	0	0	0	0.043158	0	0	0	0	0	0	0	0	0.0484
15	0	0	0.021875	0.08125	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
16	0	0	0.023333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.086667	0	
17	0	0.029167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0766
18	0	0	0	0	0	0	0	0	0	0.039545	0	0	0.027727	0	0	0	0	0	0	0	
19	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.043
20	0	0	0.063636	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
21	0	0	0.019444	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
22	0	0.025	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
23	0	0	0.031818	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
24	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
25	0	0.026923	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
26	0	0.046667	0.023333	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
27	0	0.019444	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
28	0	0	0.021875	0	0	0	0	0	0	0	0	0	0.038125	0	0	0	0	0	0	0	
29	0	0	0	0	0	0.052381	0	0	0	0	0	0	0	0	0	0	0	0	0	0.065238	
30	0	0	0.029167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0766
31	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
32	0	0.029167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0766
33	0	0.021875	0.021875	0	0	0	0.053125	0.054375	0.054375	0.054375	0.054375	0.05125	0	0.053125	0.055625	0.055625	0.053125	0.05875	0	0	
34	0	0.031818	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	
35	0	0.029167	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0.0766

Figure 22 : Assigned TFIDF Score

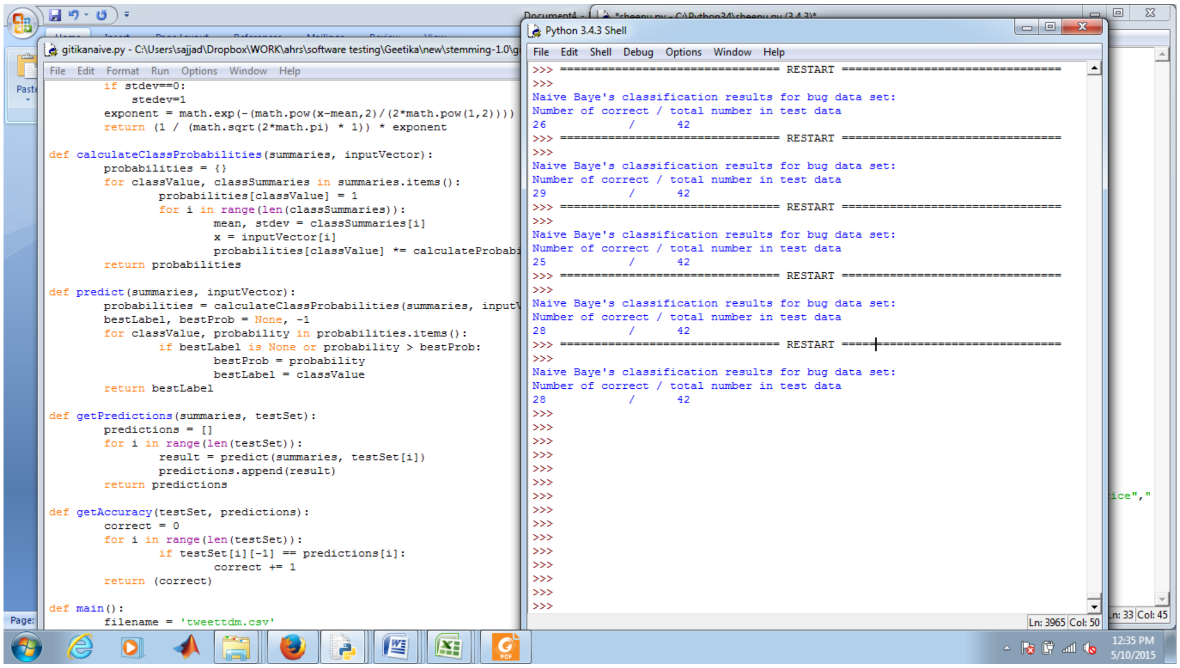
This tfidf score is used as training data for the KNN classifier and Naïve Bayes Classifier. Next figures show the results of both the classifiers i.e. KNN classifier and Naïve Bayes classifier. The accuracy of both the classifiers is calculated.



```
>>> ================================ RESTART ================================
>>>
k-NN classification results for bug data set:
k |   Number of correct / Total test records
3 |         29         /          38

>>>
>>>
>>>
>>>
>>>
>>>


>>>


>>>

>>>


>>>
>>>
>>>
>>>
>>>


>>>

>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```

```
    # return evaluation result
    return eval_result

def main():
    """ k-nearest neighbors classifier """

    # initialize runtime
    start = time.clock()


    data_tests = [2]

    for d in data_tests:
        if d == 2:

            # read dataset of malware data
            ndata=100
            dinput, dlabel = read_data('tweettdm.csv')
            dtrain=dinput[:ndata]
            dtr_label=dlabel[:ndata]
            dtest=dinput[(ndata+1):]
            true_class=dlabel[(ndata+1):]
            #dtrain, dtr_label = dinput
            #dtest, true_class = rea

        # initialize K
        K = [3]


        dist_fn = [1,2,3]

        if d == 2:

            print("k-NN classification results for bug data set:")


        print("k |   Number of correct / Total test records")


        # run knn classifier for each k and distance function
```

Figure 23: Applying KNN classifier



gitikanaive.py - C:\Users\sajjad\Dropbox\WORK\ahrs\software testing\Geetika\new\stemming-1.0\gi
File Edit Format Run Options Window Help
Python 3.4.3 Shell
File Edit Shell Debug Options Window Help
Ln: 3965 Col: 50
12:35 PM
5/10/2015

```
        if stdev==0:
            stedev=1
        exponent = math.exp(-(math.pow(x-mean,2)/(2*math.pow(1,2))))
        return (1 / (math.sqrt(2*math.pi) * 1)) * exponent

def calculateClassProbabilities(summaries, inputVector):
        probabilities = {}
        for classValue, classSummaries in summaries.items():
                probabilities[classValue] = 1
                for i in range(len(classSummaries)):
                        mean, stdev = classSummaries[i]
                        x = inputVector[i]
                        probabilities[classValue] *= calculateProbabi
        return probabilities

def predict(summaries, inputVector):
        probabilities = calculateClassProbabilities(summaries, inputV
        bestLabel, bestProb = None, -1
        for classValue, probability in probabilities.items():
                if bestLabel is None or probability > bestProb:
                        bestProb = probability
                        bestLabel = classValue
        return bestLabel

def getPredictions(summaries, testSet):
        predictions = []
        for i in range(len(testSet)):
                result = predict(summaries, testSet[i])
                predictions.append(result)
        return predictions

def getAccuracy(testSet, predictions):
        correct = 0
        for i in range(len(testSet)):
                if testSet[i][-1] == predictions[i]:
                        correct += 1
        return (correct)

def main():
        filename = 'tweettdm.csv'
```

```
>>> ================================ RESTART ================================
>>>
Naive Baye's classification results for bug data set:
Number of correct / total number in test data
26          /      42
>>> ================================ RESTART ================================
>>>
Naive Baye's classification results for bug data set:
Number of correct / total number in test data
29          /      42
>>> ================================ RESTART ================================
>>>
Naive Baye's classification results for bug data set:
Number of correct / total number in test data
25          /      42
>>> ================================ RESTART ================================
>>>
Naive Baye's classification results for bug data set:
Number of correct / total number in test data
28          /      42
>>> ================================ RESTART ================================
>>>
Naive Baye's classification results for bug data set:
Number of correct / total number in test data
28          /      42
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
>>>
```

Figure 24 : Applying Naïve Bayes classifier

Training to test ratio is kept as 3:1. A total of 140 tweets are finally selected after filtering and all and manual assignment of sentiments is done to be fed into the classifier. Two types of classifiers are implemented in this thesis.

- Naïve Bayes

  The result of Naïve Bayes Classifier is found to be 28 correct classified to that of total 42 tweets.

  The Accuracy is calculated as:

  Accuracy=28/42*100=66.66 %

- KNN

  Value of K is taken as three and the result of KNN is found to be 29 correct tweets as compared to 39 total tweets.

  The accuracy is calculated as:

  Accuracy=29/39*100=76.31%

6. Conclusion and Future Work

A methodology for the classification of sentiments was developed in this thesis for food price crisis in Indian market. Twitter API was used for streaming of tweets. The streamed tweets was filtered for relevant content and stored in a database. The several steps of pre-processing were applied on it and the tweets were removed from special characters, stop word, tokenized, etc. Stemming was done to all words in order to extract the root words.

TF-IDF score based approach was utilized and the score was calculated for each tweets. Feature Selection was applied on it using Chi Square method and information gain. The extracted features form a term document matrix which is utilized in the classification algorithm. Two classification algorithms are compared as shown in previous chapter.

The results are found to be satisfactory and when comparative analysis is done between them it is found that KNN outperforms Naïve Baye’s Algorithm. Thus an automated system is designed for opinion mining related to food price crisis using Indian tweets.

In future I will try to implement this mechanism with different classifiers and try to improve the accuracy of the system.

References

1. Medhat, Walaa, Ahmed Hassan, and Hoda Korashy. "Sentiment analysis algorithms and applications: A survey." *Ain Shams Engineering Journal* 5.4 (2014): 1093-1113.

2. Whitelaw, Casey, Navendu Garg, and Shlomo Argamon. "Using appraisal groups for sentiment analysis." *Proceedings of the 14th ACM international conference on Information and knowledge management*. ACM, 2005.

3. Nasukawa, Tetsuya, and Jeonghee Yi. "Sentiment analysis: Capturing favorability using natural language processing." *Proceedings of the 2nd international conference on Knowledge capture*. ACM, 2003.

4. Yi, Jeonghee, et al. "Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques." *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*. IEEE, 2003.

5. Balahur, Alexandra, Rada Mihalcea, and Andrés Montoyo. "Computational approaches to subjectivity and sentiment analysis: Present and envisaged methods and applications." *Computer Speech & Language* 28.1 (2014): 1-6.

6. Tan, Chenhao, et al. "User-level sentiment analysis incorporating social networks." *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 2011.

7. Lin, Chenghua, and Yulan He. "Joint sentiment/topic model for sentiment analysis." *Proceedings of the 18th ACM conference on Information and knowledge management*. ACM, 2009.

8. Feldman, Ronen. "Techniques and applications for sentiment analysis."*Communications of the ACM* 56.4 (2013): 82-89.

9. Kang, Hanhoon, Seong Joon Yoo, and Dongil Han. "Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews." *Expert Systems with Applications* 39.5 (2012): 6000-6010.

10. Shahbaz, Muhammad, and Aziz Guergachi. "Sentiment miner: A prototype for sentiment analysis of unstructured data and text." *Electrical and Computer Engineering (CCECE), 2014 IEEE 27th Canadian Conference on*. IEEE, 2014.

11. Kang, Hanhoon, Seong Joon Yoo, and Dongil Han. "Senti-lexicon and improved Naïve Bayes algorithms for sentiment analysis of restaurant reviews." *Expert Systems with Applications* 39.5 (2012): 6000-6010.

12. Moreo, Alejandro, et al. "Lexicon-based comments-oriented news sentiment analyzer system." *Expert Systems with Applications* 39.10 (2012): 9166-9180.

13. Xianghua, Fu, et al. "Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon." *Knowledge-Based Systems* 37 (2013): 186-195.

14. Ortigosa, Alvaro, José M. Martín, and Rosa M. Carro. "Sentiment analysis in Facebook and its application to e-learning." *Computers in Human Behavior* 31 (2014): 527-541.

15. Pak, Alexander, and Patrick Paroubek. "Twitter as a Corpus for Sentiment Analysis and Opinion Mining." *LREC*. Vol. 10. 2010.

16. Martin, José M., Alvaro Ortigosa, and Rosa M. Carro. "SentBuk: Sentiment analysis for e-learning environments." *Computers in Education (SIIE), 2012 International Symposium on*. IEEE, 2012.

17. Liu, Bing. "Sentiment analysis and opinion mining." *Synthesis Lectures on Human Language Technologies* 5.1 (2012): 1-167.

18. Esuli, Andrea, and Fabrizio Sebastiani. "Sentiwordnet: A publicly available lexical resource for opinion mining." *Proceedings of LREC*. Vol. 6. 2006.

19. Martineau, Justin, and Tim Finin. "Delta TFIDF: An Improved Feature Space for Sentiment Analysis." *ICWSM*. 2009.

20. O'Keefe, Tim, and Irena Koprinska. "Feature selection and weighting methods in sentiment analysis." *Proceedings of the Australasian document computing symposium*. 2009.

21. Baccianella, Stefano, Andrea Esuli, and Fabrizio Sebastiani. "SentiWordNet 3.0: An Enhanced Lexical Resource for Sentiment Analysis and Opinion Mining." *LREC*. Vol. 10. 2010.

22. Davidov, Dmitry, Oren Tsur, and Ari Rappoport. "Enhanced sentiment learning using twitter hashtags and smileys." *Proceedings of the 23rd International Conference on Computational Linguistics: Posters*. Association for Computational Linguistics, 2010.

23. Bakliwal, Akshat, et al. "Mining sentiments from tweets." *Proceedings of the WASSA* 12 (2012).
24. http://www.indiafoodbanking.org/hunger

25. https://www.wfp.org/countries/wfp-innovating-with-india/overview

26. Saif, Hassan, Yulan He, and Harith Alani. "Semantic sentiment analysis of twitter." The Semantic Web–ISWC 2012. Springer Berlin Heidelberg, 2012. 508-524.

27. Abel, Fabian, et al. "Analyzing user modeling on twitter for personalized news recommendations." User Modeling, Adaption and Personalization. Springer Berlin Heidelberg, 2011. 1-12.

28. Montoyo, Andrés, Patricio MartíNez-Barco, and Alexandra Balahur. "Subjectivity and sentiment analysis: An overview of the current state of the area and envisaged developments." Decision Support Systems 53.4 (2012): 675-679.

29. Agarwal, Apoorv, et al. "Sentiment analysis of twitter data." *Proceedings of the Workshop on Languages in Social Media*. Association for Computational Linguistics, 2011.

30. Balahur, Alexandra, et al. "Sentiment analysis in the news." *arXiv preprint arXiv:1309.6202* (2013).

31. Jebaseeli, A. Nisha, and E. Kirubakaran. "A Survey on Sentiment Analysis of (Product) Reviews." *International Journal of Computer Applications* 47.11 (2012).

32. Scholar, P. G. "Big-SoSA: Social Sentiment Analysis and Data Visualization on Big Data."

33. Horakova, Marketa. "Sentiment Analysis Tool using Machine Learning." *Global Journal on Technology* (2015).

34. Gupta, Aditi, et al. "Sentiment analysis for social media." *International Journal of Advanced Research in Computer Science and Software Engineering* 3.7 (2013): 216-221.

35. B. Pang and L. Lee. Opinion mining and sentiment analysis. Foundations and Trends in Information Retrieval, 2(1 -2):1{135, 2008.

36. B. Pang, L. Lee, and S. Vaithyanathan. Thumbs up? Sentiment classification using machine learning techniques. In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP), pages 79{86, 2002.

37. Twitter Sentiment Classification using Distant Supervision by Alec Go, Richa Bhayani, and Lei Huang.

38. J. Read. Using emoticons to reduce dependency in machine learning techniques for sentiment classification. Association for Computational Linguistics, 2005.

39. K. Nigam, J. Lafferty, and A. Mccallum. Using maximum entropy for text classification. In IJCAI-99 Workshop on Machine Learning for Information Filtering, pages 61{67, 1999.

40. Mikheev, 1999 Andrei Mikheev. Feature lattics and maximum entropy models. Machine Learning, 1999.

41. Nigam et al. , 1999 Kamal Nigam, Andrew McCallum,Sebastian Thrun, and Tom Mitchell. Text classification from labeled and unlabeled documents using EM. Machine Learning, 1999.

42. Csisz_ar, 1996 I. Csisz_ar. Maxent, mathematics, and information theory. In K. Hanson and R. Silver, editors, Maximum Entropy and Bayesian Methods. Kluwer Academic Publishers, 1996. [9]. [Rosenfeld, 1994] Ronald Rosenfeld. Adaptive Statistical Language Modeling: A Maximum Entropy Approach. PhD thesis, Carnegie Mellon University, 1994

43. A.Kowclka, Aditi Gupta, Karthick Sondhi, Nishit Shivhre, Raunaq Kumar "Sentiment analysis for social media" Volume 3, Issue 7,International Journal of Advanced Research in Computer Science and Software Engineering. pp.216-221,2013.

44. Ayushi Dalmia ,Mayank Gupta, Arpit Kumar Jaiswal Sunil and Chinthala Tharun Reddy [online] "Sentiment Analysis in twitter" Available at :http://researchweb.iiit.ac.in/. [4]. B. Pang and L. Lee, “Opinion mining and sentiment analysis,” Foundations and Trends in Information Retrieval,vol.2,no.1-2, pp.1–135,2008.

45. Bing Liu, Sentiment Analysis and Opinion Mining, Morgan and Claypool Publishers, May 2012.pp.18-19, 27-28,4445,47,90-101.

46. B. Mane, Yashwant Sawant, Saif Kazi, Vaibhav Shinde "Real Time Sentiment Analysis of Twitter Data Using Hadoop" (2014) Volume 3, International Journal of Computer Science and Information Technologies, pp-3098 – 3100.

47. Data Visualization [online] Available at:http://www.sas.com/en_us/insights/big-data/datavisualization.html.

48. Efthymios Kouloumpis , Theresa Wilson, and Johanna Moore, “Twitter sentiment analysis: the god the bad and the OMG!,” in Proceedings of the 5th International AAAI Conference on Weblogs and Social Media, pp. 538–541, 2011.

49. Flume, http://flume.apache.org/ [10]. G.Vinodhini, RM.Chandrasekaran. "Sentiment analysis and opinion Mining: A Survey " , Volume 2, Issue 6,International Journal of Advanced Research in Computer Science and Software Engineering.2012.

50. http://www.fao.org/worldfoodsituation/foodpricesindex