Harnessing Twitter for Automatic Sentiment Identification

Amiya Kumar Dash

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Harnessing Twitter for Automatic Sentiment Identification

Thesis submitted in

May 2015

to the department of

Computer Science and Engineering

of

National Institute of Technology Rourkela

in partial fulfillment of the requirements

for the degree of

Master of Technology

in

Computer Science and Engineering

by

Amiya Kumar Dash

[Roll No. 213CS1141]

under the guidance of

Prof. Sanjay Kumar Jena

Department of Computer Science and Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India

Department of Computer Science & Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India. www.nitrkl.ac.in

Declaration by student

I certify that

- I have complied with all the benchmark and criteria set by NIT Rourkela Ethical code of conduct.
- The work done in this project is carried out by me under the supervision of my mentor.
- This project has not been submitted to any other institute other than NIT Rourkela.
- I have given due credit and references for any figure, data, table which was being used to carry out this project.

Amiya Kumar Dash May 30, 2015

Place: Rourkela

Department of Computer Science & Engineering
National Institute of Technology Rourkela
Rourkela-769 008, Odisha, India. www.nitrkl.ac.in

Dr. Sanjay Kumar Jena
Professor

May 30, 2015

Certificate

This is to certify that the work in the thesis entitled ***Harnessing Twitter for Automatic Sentiment Identification*** by ***Amiya Kumar Dash***, bearing Roll No. 213CS1141, is a record of an original research work carried out by him under my supervision and guidance in partial fulfilment of the requirements for the award of the degree of *Master of Technology* in *Computer Science and Engineering*. Neither this thesis nor any part of it has been submitted for any degree or academic award elsewhere.

Sanjay Kumar Jena

Acknowledgment

karmaṇy evādhikāras te mā phaleṣu kadācana
mā karma-phala-hetur bhūr mā te saṅgo 'stv akarmaṇi

***BHAGAVAD-GĪTĀ**, Chapter-2, Verse-47*

Thank You God for giving me the courage to believe in the above philosophy...

Foremost, I would like to express my earnest gratitude to my thesis guide, Prof. Sanjay Kumar Jena for believing in my ability to work on the challenging domain of sentiment analysis. His profound insights has enriched my research work. The flexibility of work he has offered me has deeply encouraged me producing the research.

I am very much indebted to Dr. S. K. Rath , Head of the Department, Computer Science engineering, National Institute of Technology, Rourkela and other faculty members for his support during my work.

My special thanks go to Jitendra Kumar Rout, Soubhagya Sankar Badapanda and Santosh kumar Sahu for providing immense help and encouragement during my thesis work.

I would conclude with my deepest gratitude to my parents and all my loved ones. My full dedication to the work would have not been possible without their blessings, unconditional love, trust, and moral support. This thesis is a dedication to them who did not forget to keep me in their hearts when I could not be beside them.

Amiya Kumar Dash

Abstract

Sentiment Analysis is a motivating space of research because of its applications in different fields. Gathering opinions of individuals about products, social and political events, and problems through the web is turning out to be progressively prevalent consistently. People's opinions are beneficial for the public and for stakeholders when making certain decisions. Opinion mining is a way to retrieve information through search engines, web blogs, micro-blogs, Twitter and social networks. User generated content on Twitter gives an ample source to gathering individuals' opinion. Due to the gigantic number of tweets as unstructured text, it is difficult to outline the information physically. Accordingly, proficient computational strategies are required for mining and condensing the tweets from corpuses which, requires knowledge of sentiment bearing words. Many computational methods, models and algorithms are there for identifying sentiment from unstructured text. Most of them rely on machine-learning techniques, using Bag-of-Words (BoW) representation as their basis. In this study, we have used lexicon based approach for automatic identification of sentiment for tweets collected from twitter public domain. We have also applied three different machine learning algorithm (Naive Bayes (NB), Maximum Entropy (ME) and Support Vector Machines (SVM)) for sentiment identification of tweets, to examine the effectiveness of various feature combinations. Our experiments demonstrate that both NB with Laplace smoothing and SVM are effective in classifying the tweets. The feature used for NB are unigram and Part-of-Speech (POS), whereas unigram is used for SVM.

Keywords: Bag-of-Words (BoW), Lexicon, Machine Learning Algorithms, Laplace Smoothing, Part-of-Speech (POS)

Contents

List of Figures

List of Tables

Chapter 1

Introduction

"What people think" has forever been a very essential for many of us throughout the decision-making method. Before the familiarity of the World Wide Web, our friends were asked to suggest an automobile mechanic or to explain who they were aiming to vote for in local elections,or consulted client Reports to determine what product to purchase. However the net and the web have currently (among alternative things) created it attainable to search out opinions from the vast pool of individuals that neither belongs to personal contacts nor to well-known skilled critics i.e. individuals we never knew about. Furthermore, alternately, more individuals are making their opinions accessible to outsiders by means of the Internet. From two surveys carried on more than 2000 American adults, each 81% of Internet users (or 60% Americans) have accomplished research on a product on-line, at least once and 20% (15% Americans) prefer it on a specific day. We can say that for people's seeking out or expressing opinions on-line, consuming products and services can not be considered as the only criterion. Another important variable is the requirement for political information. Presently people can use the email for election campaign by gathering of information and exchanged of views about elections on-line. The user relies on on-line advice and suggestion because the information directly deals with opinion as primary object. But, according to Horrigan [1] report although experiences of majority of American Internet users during online product research is positive, reporting of 58% users about missing, difficult to find, confusing, and/or overwhelming of online information is surprising. Thus, the need for better information-access systems to help consumers of products and

information is of high demand. With the explosion of *Web 2.0* platforms such as blogs, discussion forums, and various other types of social media consumers share their brand experience, opinions regarding different products or services. Companies are progressively realizing that opinions of other consumers and reputation of their brand loyalties can be influenced by such opinions. So they starts responding to the consumer insights by social media monitoring and modifying their marketing messages, positioning of brand, product development and other activities accordingly. But industry analyst found that near about 80,000 new blogs and 2 millions new posts created daily. Due to the maximum use of Internet and gradual change in consumer behavior, the traditional monitoring methods have been crippled. Therefore, for monitoring purpose, advanced technologies related to product image is demanded. Subsequently, apart from individuals, a separate audience group for systems capable of analyzing consumer sentiment automatically, as described in no small part in online venues, are companies keen to realize, how their products and services are being perceived. This chapter presents a systematic overview of research trends, advances, challenges in Opinion Mining and Sentiment Analysis.

1.1 A Note on Terminology: Opinion Mining and Sentiment Analysis

Sentiment Analysis(SA), conjointly referred to as Opinion Mining is that the field of study that analyzes people's opinion, sentiment, attitude, evaluation and emotions towards a entity. An opinion is the private state of an individual, and as such, it represents the individual's ideas, beliefs, assessments, judgments and evaluations about a specific subject/topic/item. Liu *et al.*[2] conclude that others' opinions have a great impact on and provide guidance for individuals, organizations and social communities during the decision making process. During this process, human beings require fast, accurate and concise information so they can make quick and accurate decisions. Individuals usually ask their companions, relatives, and specialists for information during the decision-making process and their opinions and views are based on experiences, observations, concepts, and beliefs which can be either positive or negative about a subject.

Opinion Mining (OM) is a procedure used to extract opinion from text. According to Pang *et al.* [3] "OM is a recent discipline at the crossroads of information retrieval, text mining and computational linguistics which tries to identify the opinion expressed in natural language texts". Opinion mining is a field of Knowledge Discovery and Data mining (KDD) that uses Natural Language Processing(NLP) and statistical machine learning techniques to separate opinionated text from factual text. Opinion mining tasks involve opinion identification, opinion classification (positive, negative, and neutral), source & target identification, and opinion summarization. The main concern in Opinion Mining task is how to automatically identify opinion components and summarize the opinion about an entity from a huge volume of unstructured text.

Sentiment Analysis is concerned with automatically extracting sentiment related information from a text and aims to categorize text as positive or negative on the premise of the positive or negative sentiment (opinion) expressed in the document/sentence towards a topic. A document/sentence with positive or negative sentiment is also said to be of positive or negative polarity respectively[2]. The granularity of polarity can be up to the level of words. That is textual information can be classified as either objective or subjective. Objective (non-polar) sentences and words represent facts, while subjective (polar) sentences and words represent perceptions, perspectives or opinions. It is important to make distinction between subjectivity detection and Sentiment Analysis as they are two separate task in natural language processing. Sentiment analysis can be dependently or independently done from subjectivity detection. Pang and Lee [4] state that to get better result subjectivity detection performed prior to Sentiment Analysis.

The task of SA is very challenging, not only due to the syntactic and semantic variability of language, but also because it involves the extraction of indirect or implicit assessments of objects, by means of emotions or attitudes. That is why automatic identification of sentiment requires fine grained linguistic analysis techniques and substantial efforts to extract features for machine learning or rule-based approaches.

1.2 Sentiment Analysis Tasks

In this section we present the key tasks of Sentiment Analysis[5]. These tasks are derived from the definition of sentiment which is a quintuple defined as follows[2]:

Definition(sentiment)

An opinion is a quintuple, ($e_i, a_{ij}, s_{ijkl}, h_k, t_l$)
where,

- e_i is the name of the entity
- a_{ij} is an aspect of e_i
- s_{ijkl} is the sentiment on aspect a_{ij} of entity e_i
- h_k is the opinion holder
- t_l is the time when the opinion is expressed by h_k

The principle segments of a SA issue are the source of the opinion, the target of the opinion and the evaluation expressions or remarks made by opinion holder. According to Liu [6] SA problem is defined as "Given a set of evaluative text documents D that contain opinions (or sentiments) about an object, Sentiment Analysis aims to extract attributes and components of the object that have been commented on in each document $d \in D$ and to figure out whether the comments are positive, negative or neutral". Usually, an opinion is expressed by an individual (opinion holder) who conveys a perspective (positive, negative, or neutral) about an entity (target object, e.g. person, item, organization, event, service, etc.). The following subsections describe the key tasks and approaches to each sub-problem of Opinion Mining.

1.2.1 Subjectivity and Polarity Classification

The core task of Sentiment Analysis is to automatically identify opinionated text in a document. These mined text are then classified as subjective and objective [7]. Most of the existing research agrees that objective text constitutes factual info whereas subjective text

represents individual views, beliefs, opinions or sentiments. Hence most of the Sentiment Analysis frameworks utilize the subjective text for opinion hood determination. While different methodologies have been adopted for this subtask of SA, the most common include heuristics and discourse structure, coarse and fine-grained analysis, key word and concept analysis [8]. Determination of opinionated text in a document is divided into two subtask: subjectivity classification and polarity classification. Subjectivity classification methods are used to classify classify terms, sentences and documents into opinion and non-opinion, where as polarity classification techniques are used to classify opinionated terms into positive, negative and neutral statements. Some of the work on subjectivity classification use weighting techniques to identify the strength of subjectiveness, i.e. weakly negative and strongly negative or weakly positive and strongly positive [9].

1.2.2 Sentiment Target Identification

Sentiment(opinion) target identification is an extremely important feature of SA task. Here the target refers to the person, object, feature, event or topic about which the sentiment is expressed. It is extremely important for the manufacturers, public and the merchants to do in–depth analysis of every aspect of a product based on customer opinion. In order to compare reviews it is necessary to automatically determine and extract those features that are mentioned within the reviews. Hence, feature mining of product is vital for opinion mining and opinion summarization [10]. Sentiment identification is a very challenging task. This is because to identify opinion targets in a sentence or document the system should be able to determine evaluative expressions and also, certain features that are not explicitly present and need to identify from the term semantic called implicit features. Existing research on Sentiment target identification reveals that the process of sentiment target extraction requires various Natural Language Processing(NLP) techniques such as pre–processing, Part–of–Speech tagging, noise removal, feature selection and classification.

Several techniques have been proposed for automatic identification of sentiment target. These techniques can be broadly divided into two major categories: unsupervised and supervised. Supervised learning strategies are based on manually labeled text. During this

approach, machine-learning techniques are used to train the model on manually labeled data to classify and predict features within the reviews. Though supervised techniques offer better results for feature extraction, it needs manual work for the preparation of the training sets. Consequently, this method is time overwhelming, skill-oriented, and, sometimes, domain dependent. The most widely used machine-learning techniques for sentiment identification are Naive Bayes classifier, Support Vector Machine(SVM), Neural Networks and K-Nearest Neighbor(KNN). Interestingly, unsupervised approaches don't need training data and they automatically predict product features based on syntactic patterns and semantic relatedness.

1.2.3 Sentiment Source Identification

The source of sentiment or sentiment holder is the person or medium who expresses the sentiment. Sentiment source is extremely important during authentication and classification of sentiment, because the quality and reliability of a sentiment is greatly dependent on the source of that opinion. For example an expert sentiment has greater strength in comparison to an ordinary person and a sentiment is reliable when it is produced from an authentic source. So the process of Sentiment Analysis task needs to identify the source of the sentiment. Determining sentiment holder is also a NLP problem that has been the subject of numerous studies over the years.

1.3 Literature Survey

Sentiment Analysis is an emerging topic of research now-a-days. In past few years, numerous research has been conducted in this area to develop a system on Opinion Mining which is more reliable and gives better accuracy. Existing research used both rule based and statistical machine learning approaches for Opinion Mining and Sentiment Analysis. In this section, we briefly discussed some of the techniques on Sentiment Analysis and their applications.

The business potential of Sentiment Analysis has resulted in an exceedingly important quantity of analysis and Pang [3] provides an overview. Ibrahim *et al.* [11] presents an in

depth survey about different techniques used for Opinion Mining and Sentiment Analysis. Pang and Lee use Naive Bayes, Maximum Entropy and Support Vector Machines for SA of movie reviews considering distinctive features like unigrams, bigrams, combination of both, including Parts–of–Speech and position information with unigram, adjectives etc. [4][12]. It was seen from their experiment that Feature presence is more important than feature frequency. It was also observed that for small feature space Naive Bayes performs better than SVM but when feature space is increased SVM performs better than Naive Bayes classifier. Turney [13] gives an unsupervised algorithm which uses semantic orientation of the phrases for classification of reviews. Esuli *et al.* [14] developed SentiWordNet lexicon which contains opinion strength for each term . Hamouda A. *et al.*[15], uses SentiWordnet Lexicon for classification of reviews. A dictionary Based techniques is proposed by Fei G., Bing Liu, Castellaons M. to identify aspects of a review by considering adjectives only.

Our work is to perform sentence-level sentiment identification, where we have classified tweets using three different machine learning algorithms: Multinomial Naive Bayes, Maximum Entropy and Support Vector Machine (SVM). To increase the accuracy of the classifier, we have pre-processed the tweets to remove the non-polar words and consider only the polar words that give sentiment.

1.4 Key Problems Addressed by Sentiment Analysis Research

As the Internet and web advancements keep on growing, the scope in the area of information retrieval also increasing. Presently researchers take interest in the area of Sentiment Analysis, which is a sub area of knowledge discovery and information retrieval. In spite of various research efforts, the present SA studies and applications still have limitations and provides space for development. According to pang *et al.* [3] classification and extraction are the two broad categories in the areas of Opinion Mining. Classification involves detecting whether a piece of text is subjective or objective and if it is subjective, what will be its polarity. Where as extraction involves information retrieval and to identify the key attributes of an opinion, for example the opinion holder or the entity it refers to etc.

Esuli *et al.* [14] categories the Opinion Mining task into three classes:(1) determining the degree in which a given text is objective or subjective; (2) if a text is subjective determine whether it gives positive or negative polarity; (3) For a subjective text determine the strength of its polarity.

One field of research that relate to Opinion Mining is to develop a computational model that can detect human emotions like anger, fear, surprise etc. But developing such a model is very closely linked to the problem of subjective detection, as both relate to the expression of human emotions. Major challenges that have made SA or OM very difficult, related to Natural language processing like semantic relatedness, context dependency and ambiguity. Some of the important challenges in Sentiment Analysis are discussed in this section.

The First is subjectivity detection i.e. to identify the sentiment or opinion containing sentences. For example consider two sentences in a review of city Singapore."Singapore's economy is heavily dependent on tourism and IT industry. It is an excellent place to live in". the first sentence is an objective or factual one and does not convey any sentiment towards Singapore. So the objective sentence needs to be removed as it doesn't have any role to determine the polarity of a sentence. The Second is Word sense disambiguation (WSD) that is a word considered to be positive in one situation may be considered negative in another situation. Take the word "low" for instance. If someone said "low price", that would be a positive opinion. If someone said "low quality", that would be a negative opinion. This difference in meaning indicates that a system trained to collect opinion on one type of product feature may not perform very well on another. A Third challenge is that people can be contradictory in their statements. It has been found that in most of the reviews both positive and negative sentiments are present, which can be feasible by scanning the sentences one at a time. Where as in social media sites such as twitter or blogs people are combining different opinions in same sentences which is very difficult to analyze. For example: "the movie bombed even though the lead actor rocked it" is easy for a human to understand, but more difficult for a machine to parse. The fourth challenge is to handle the negation, as presence of negator can change the orientation of the sentence. Fifthly, keeping the target in focus can be a challenge. Consider the statement "Windows 7 is much better than Vista!". For a target Windows 7 above statement gives positive polarity whereas for a

target Vista it gives negative polarity.

1.5 Summary

In this chapter the research area of Sentiment Analysis are surveyed. We briefly discuss the subtask of Sentiment Analysis and the approaches that were used till now. The primary objective of Sentiment Analysis is divided into two categories: first to identify whether a given text is subjective or objective and second is if it is subjective find its polarity. This chapter has highlighted the challenges in Sentiment Analysis and the techniques used to over come it.

1.6 Organization of the Thesis

The thesis is organized as follows.

Chapter 2— Data Collection and Preprocessing will discuss the procedure we have used to collect data from Twitter public domain and the preprocessing needs to be done before applying any of the Sentiment Analysis techniques.

Chapter 3— Sentiment analysis using Lexicon Based Approach will discuss the detail structure of the SentiWordNet lexicon, with the objective to best use of this lexical resource to build a Sentiment Analysis system for tweets.

Chapter 4— Sentiment analysis using machine learning techniques will describe how Machine learning algorithm have been applied to Sentiment Analysis.

Chapter 5— Conclusions will describe the analysis and conclusion of our experiment.

Chapter 2

Data Collection and Preprocessing

With the swift upgradation of Communication and Information based Technology, immense complexities have been added to information broadcasting. Specifically, prevalentness of the social networks like facebook, twitter, myspace in controlling the information transmission and bussiness intelligence can not be overlooked. Tweets from twitter were used to continue the research activity. In spite of the small size of the messages (confined to 140 character long, unlike other social networking sites), it is of high importance, since a lot more can be discovered from this small space. Moreover, from the photos, videos and conversations, the whole story can be percepted at a glance, all at a single palce. Another reason is availability of data. Using *Twitter API* we can collect millions of tweets to train our model where as in past research, tests only consisted of thousands of training items.

Collection of data for research purpose is not that simple as it appears, since suitable and significant presumptions and conclusions are to be made. There are three differently collected datasets: test data, subjective training data, and objective (neutral) training data. Before discussing them, Twitter API will be discussed.

2.1 Twitter API

In general, two APIs are offered by Twitter: REST and Streaming. REST API further comprises of two other APIs i.e., REST API and Search API (whose difference is due to their history of upgradation). Streaming API varies from REST APIs in the sense that its

connection is longlived and offers data in almost realtime. In contrast, the REST APIs support short-lived connections and are ratelimited (one can download a restricted amount of data per day). The Twitter data like status updates and user info are accessed by the REST APIs regardless of time. However, availability of data older than a week is not facilitated by Twitter. Thus, the access of REST is limited to data tSwittered not prior to more than a week. Consequently, where REST API approves accessing these accumulated data, Streaming API facilitates access to data as it is being twittered.

For our present research, we relied on the Streaming and Search REST API to collect the data. Where the Streaming API was used to accumulate the training samples, Search REST API was engaged for the test data. These two datasets i.e., training and test data were to be collected in different methods. Why the Streaming API is preferred for collection of training data is because having a large amount of tweets (size of training data is a large) demands a non-rate-limited long-lived connection. Correspondingly, the Test data were to be collected using the Search REST API for specific reasons that will be revealed soon.

There prevails a language parameter both for the streaming and Search REST API, which can be set to a language code, e.g. 'en' to collect English data. However, type of the collected data samples is not only limited to English. In other words, there also exists tweets in other languages, thus making the overall data noisy. Therefore, we have decided to collect tweets that contains some specific emoticons without regard to language and keep the task of seprating data into English and non-English for later purpose. The approach of accumulating the data is discussed below.

2.2 Data Collection

2.2.1 From Twitter using Third Party API (Twython)

As the objective of the thesis is to identify the sentiment (positive,negative and neutral) of tweets with respect to a particular product or a movie, so only tweets about that particular product or a movie should be collected. However, this is not a simple task. There seems to be no way of obtaining all and only tweets that are posted w.r.t. a particular object. So for retriving tweets from Twitter we have used a third party API (Twython). Twython

is a pure python wrapper for the Twitter API which supports both normal and streaming Twitter APIs. To work on this API we have to download the twython-master from `https://github.com/ryanmcgrath/twython/archive/master.zip` and install it.

For implementation, we have collected 6000 tweets from our Twitter account by running a script that uses twython API . The corpus contains tweets about Apple, Google, Microsoft and Twitter.

For emotion analysis we have also retrieved tweets from Twitter by using *hashtag* and form a large emotion dataset. It has been found from literature survey that basically there are seven categories of emotions *(anger, love, fear, joy, sadness, surprise, thankfulness)* are present. In our work we have harnessed twitter to handle the emotion identification problem. We have used these seven emotions as key words to collect tweets from twitter. A snapshot of the tweets retrieved using *hashtag* are shown in the Figure 2.1. Figure 2.2, Figure 2.3 and Figure 2.4 represent the volume of tweets retrieved w.r.t time for *#anger*, *#sad* and *#surprise* respectively.

Service	Term	Username	Name	Update	Link	Location	Followers	Friends	Time(PDT)
twitter	#joy	mariellabella	mariella	#pasta #meatballs #food #lunch #happy #mom #joy #yummy #handmac	http://twitter.	Trinidad	105	445	07-04-2015 09
twitter	#joy	KariJoys	Kari Joys MS	RT @DiRiseborough: #Letgo of the weight & find your #JOY on the	http://twitter.	Spokane, WA	23425	24309	07-04-2015 09
twitter	#joy	AllGodsThings	Omnia Dei	Alleluia! "@RadiateLA: #Easter is all about #joy. Radiate it! #RadiateLA	http://twitter.	Caribbean	266	338	07-04-2015 09
twitter	#joy	Rastamon86	fiestas EDM madrid	#Zoologicoclub #capital #joy #marcoAldani #madriz #salir #wekeend #d	http://twitter.	Madrid	1714	1366	07-04-2015 09
twitter	#joy	debsie301261	Deborah Myerscough	RT @ramblingsloa: " #Joy attracts #Joy" - Rhonda Byrne @thesecret @S	http://twitter.		525	581	07-04-2015 09
twitter	#joy	upfortomorrow	Tamara	RT @headquarters: I'm gonna root root root for BOTH teams! #Opening	http://twitter.		18	82	07-04-2015 09
twitter	#joy	wtjohnson01	Whitney Johnson	The little spaces we create for peace. #peaceful #spring #littlethings #l	http://twitter.		292	620	07-04-2015 09
twitter	#joy	ashrafxd	ashraf	RT @pixodentist: #FriFotos Blossom the #joy of Spring. http://t.co/7pA	http://twitter.	God`s own country / ma	314	1046	07-04-2015 09
twitter	#joy	LessysWorld	Lessy	And you should too! ⭐⭐ #Joy #TrueJoyComesThroughChrist ⭐ https://t.c	http://twitter.	Brooklyn, NY	717	650	07-04-2015 09
twitter	#joy	eraivy	eraivy	RT @JohnSilkauskas: Start #singing start humming, start #worshiping rig	http://twitter.	kualalumpur	936	810	07-04-2015 09
twitter	#joy	NanaJenLaw	Nátalie Tavares	RT @Newsin01887: @JenLawUS @olv #Joy is filming RIGHT NOW at the	http://twitter.	RJ, Brasil	471	566	07-04-2015 09
twitter	#joy	patlejarde	Patricia Lejarde	luv u girlz @erikalaylo @ivymancilla @AsiRaihana @_coleenqb @jshxa	http://twitter.	PH	743	285	07-04-2015 09
twitter	#joy	TheJoyTrain	#JoyTrain	RT @DiRiseborough: #Letgo of the weight & find your #JOY on the	http://twitter.	Global	2269	2414	07-04-2015 09
twitter	#joy	Iwrenci	amendoa	RT @Newsin01887: @JenLawUS @olv #Joy is filming RIGHT NOW at the	http://twitter.		1944	776	07-04-2015 09
twitter	#joy	bydefault904	Drizzy	@adamcarolla @Jokoy you don't understand Jo. We have a serious bro	http://twitter.	NE Florida	109	318	07-04-2015 09
twitter	#joy	kkax_	ลาพดอ๊amine	RT @parkkim9699: [CAP] 150404 #JOY ⭐http://t.co/pyalllp09N ©Studio,	http://twitter.	s k r 2 1 ₀	6039	131	07-04-2015 09
twitter	#joy	mathewdennis	Dennis Mathew	RT@Samsamum Focus for today Making peace with my complexity - #lo	http://twitter.	Canada eh!	26284	26089	07-04-2015 09
twitter	#joy	TheJoyTrain	#JoyTrain	RT @KariJoys: @adservio Transform your #Anxiety to #Joy!#Spokane #J	http://twitter.	Global	2269	2414	07-04-2015 09
twitter	#joy	TheJoyTrain	#JoyTrain	RT @KariJoys: Real, Inspiring #Stories~People You May Know~Found #J	http://twitter.	Global	2269	2414	07-04-2015 09
twitter	#joy	xolovebeam	บีมเสล็อต	RT @babyvelvet_th: [HQ] 150328 #JOY at Fansign Event In Busan ©man	http://twitter.		625	1546	07-04-2015 09
twitter	#joy	TheJoyTrain	#JoyTrain	RT @KariJoys: Welcome to the #JoyTrain community! Just RT other twe	http://twitter.	Global	2269	2414	07-04-2015 09
twitter	#joy	TheJoyTrain	#JoyTrain	RT @KariJoys: #NewBook> "Who Stole Your #Joy?" Inspiring #hope!	http://twitter.	Global	2269	2414	07-04-2015 09
twitter	#joy	xolovebeam	บีมเสล็อต	RT @babyvelvet_th: [OFFICIAL] 150407 #JOY & #JIMIN at After Sch	http://twitter.		625	1546	07-04-2015 09

Figure 2.1: Snapshot of the tweets collected viz. third party API

2.2.2 Tweets About Product and Manually Annotation

In order to apply our machine learning classifier the corpus containing tweets about different products are manually examined and annotated as positive, negative, neutral and irrelevant. Here is the procedure, we have followed to manually annotate the tweets.

- A Twitter post that contains factual words about a product was annotated as neutral.
- A Twitter post that contains subordinating conjunctions was annotated the sentiment of the main clause.

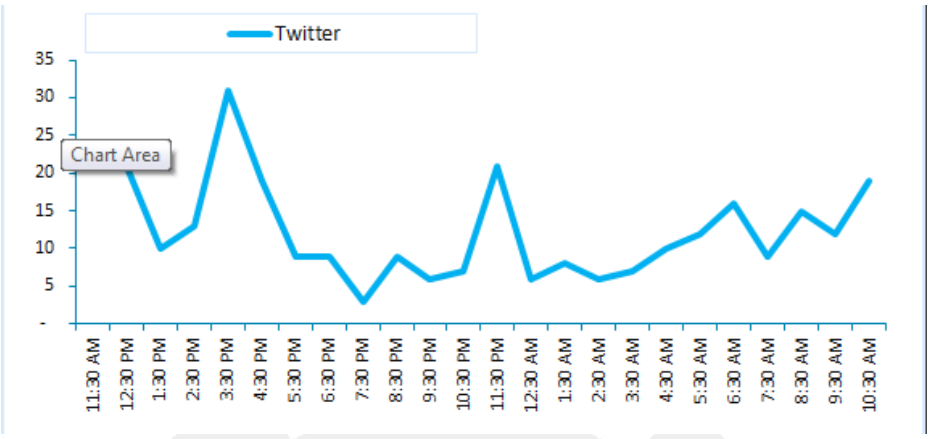

Figure 2.2: Snapshot of volume of tweets w.r.t time for # anger

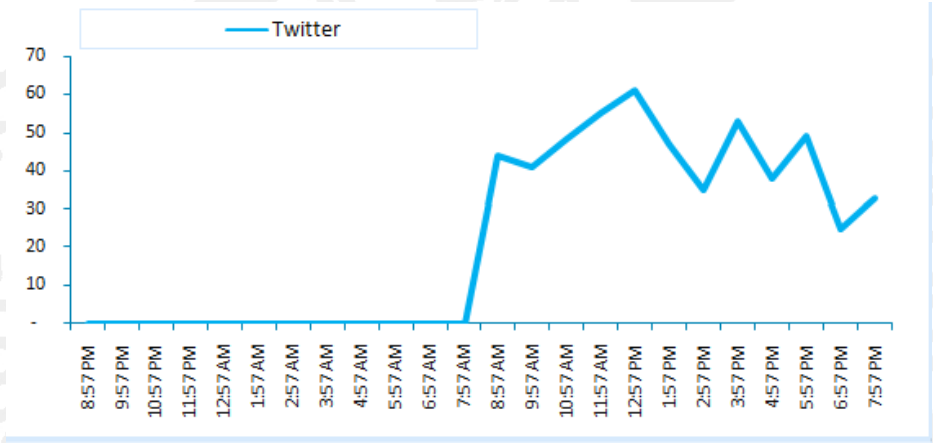

Figure 2.3: Snapshot of volume of tweets w.r.t time for #sad

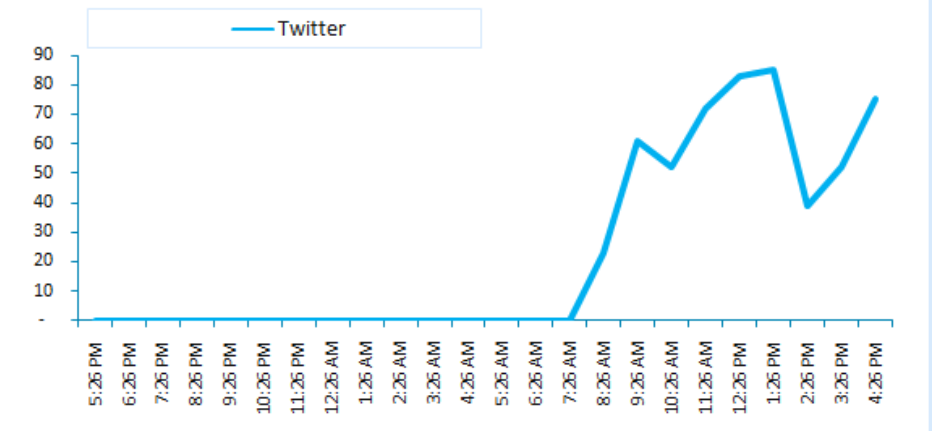

Figure 2.4: Snapshot of volume of tweets w.r.t time for #surprise

- A Twitter post that contains subtleties was annotated as one of the three sentiment classes only if it was clearly determinable.

- A Twitter post that was difficult to give a sentiment, was annotated as neutral.
- A Twitter post that were not in English language or not related to the topic was annotated as irrelevant.

Out of the 6000 Twitter posts about four different products that have been annotated following the above procedure, 1906 Twitter posts are irrelevant i.e. not related to topic, 774 posts are negative, 697 posts are positive and 2623 Twitter posts are neutral. The table below shows the break down of the topic data.

Table 2.1: Break down of topic data

Topic	**# Positive**	**# Neutral**	**#Negative**	**#Irrelevant**	**Twitter search term**
Google	248	634	91	528	# google
Apple	221	611	407	194	# apple
Twitter	103	677	108	641	# twitter
Microsoft	125	701	168	543	# microsoft

2.3 Training Data

There are two datasets that are used for the training of a classifier: subjective data and neutral data. Subjective data are data that involve positive and/or negative sentiment while neutral data is data that does not show sentiment. The following data was collected to be used to train a classifier.

2.3.1 Subjective Data

Subjective data in this context are data that contain negative and/or positive sentiment or emoticons. While it is possible to collect enough negative and positive data in one or two consecutive days by running our script using emoticons, we used our manually annotated positive and negative tweets as subjectivity dataset(1100 Twitter posts out of 1471 posts)

to train the classifier. As can be seen from the Table 2.1 , out of 6000 tweets a total of 774 tweets are negative and 697 are positive. The fact that there are more negative tweets than positive tweets shows that more people use negative emoticons than positive emoticons . It is also noteworthy that there are substantial amount of tweets that contain both negative and positive emoticons. Tweets that contain both negative and positive emoticons are confusing because they contain both sentiments, so we have annotated these tweets as neutral.

2.3.2 Removing non-English Tweets

The tweets we have retrieved contain both English and non-English tweets. As our objective is to find the sentiment of tweets that are in English language, we have eliminated the non-English tweets from the dataset and train the classifier using English tweets. This was possible by using Google's language detection web service, which requires a reference website against which strings are compared to determine their language. The strings in this case are the tweets. The web service enables one to specify a confidence level that ranges from 0 to 1. Since Twitter data contains a lot of slang and misspellings, we set the confidence level at 0 not to get rid of many English tweets.

2.3.3 Neutral Tweets

Neutral tweets in this context are tweets that contain factual words about a product or contain both positive and negative sentiment. Out of 6000 tweets, a total of 2623 tweets are annotated as neutral and 1906 tweets are irrelevant. As we have considered three classes positive, negative and neutral, we converted all the irrelevant class to neutral classes. A total of 4529 tweets are present in our dataset out of which 2529 neutral tweets are used to train the classifier.

2.4 Data Preprocessing

Preprocessing of data is essential to remove the incomplete, noisy and inconsistent data. Preprocessing must be done in order to apply any of the data mining functionality. We have

employed the following preprocessing task before applying any of the sentiment analysis approach(lexicon based or machine learning).

- **Removing URLs**

  In general, for analysis of sentiment of the tweets, URLs can not be held responsible. For example, take a look of the sentence "She has logged into www.ecstasy.com as she is bored". Although the sentence is negative, due to presence of the word 'ecstasy', it may appear neutral resulting in a false prediction. To overcome this draw back, we must remove the URLs.

- **Filtering**

  Usually, people use words with repeated letters like 'coooool' or 'happppyyyy' to reveal their intensity of expression. Since, these words do not exist in English language, there arises the need for eliminating the extra letters. So, we adopt to the rule that a letter can't repeat more than three times.

- **Removing Question words and Stop words**

  Question words e.g. what, which, how etc. do not contribute to polarity. Hence, such words can be removed to ensure complexity reduction. We also discarded words like *for, above, about* etc. called stop words as they hardly contribute to detection of polarity.

- **Removing specail character**

  In order to resolve the discrepancies during assignment of polarity, special characters like [] , {}, () should be avioded. For example "It's bad:". Unless the special characters are removed, they may concatenate and make those words unavailable in the dictionary. In order to overcome this, we also remove the special character.

- **Removal of retweets**

  Retweeting can be defined as the process of copying the tweet of another user and posting to another account. Usually, this takes place when a user like another user's tweets. Retweets are commonly abbreviated with \ RT. For example, following tweet

may be considered: RT @Anum3288: Finally made a full paper box \U0001f49c # ArtLovers # paperbox # creativity # giftbox.

- **Removing Hash symbol**

  A hashtag is a type of label used in social networking sites or microblogging services which makes it easier for users to find messages with a specific theme or content. For example, if you search on #LOST (or #Lost or #lost, because it's not case-sensitive), we'll get a list of tweets related to #lost. Generally this symbol is used to indicate nouns. We stripped hash symbol (#*tomorrow* → *tomorrow*) as it is not needed for polarity detection.

2.5 Summary

This chapter has looked at the data collection and annotation schema that have been used for sentiment analysis of tweets. It has also highlighted the preprocessing techniques that we have employed before applying any of the sentiment analysis approach (lexicon based or machine learning).

Chapter 3

Sentiment Analysis using Lexicon Based Approach

To accomplish detection of subjectivity and classification of sentiments, use of key words are considered to be indicative of positive or negative bias. This approach is governed by the logic that words can be defined as an ensemble of opinion contents. In literature, multiple methods based on this approach exist with prominent success: a subjectivity detection method is proposed by Turney *et al.* [13], which relies on a list of seed words decided by the proximity measure to other common terms.

A test was performed by Pang *et al.* [3] with a manually created positive and negative words list to classify the sentiments of movie reviews. Likewise, a lexicon of positive, negative and valence shifter terms was prepared by Kennedy *et al.* from various sources to perform document-level sentiment differentiation. Interestingly, for the approaches based on word lists, training data are not necessary for making predictions, as it depends exclusively on a pre-defined sentiment lexicon. These methods are suitable for cases, where no demand of training data persists. Consequently, such methods can be categorized under unsupervised learning techniques. In this chapter, we have discussed about SentiWordNet database structure in detail. We have used this lexicon to perform sentence-level sentiment classification. As outcome, a specification for a perticular features is derived, for which tweets plain text is considered as the starting point and sentiment information is captured on terms using SentiWordNet. In our experiment we have found that though using

SentiWordNet we can identify sentiment of tweets it is better to generate a lexicon from our training corpus to perform sentiment analysis. So in this chapter we have discussed how to generate a lexicon from the training corpus itself.

3.1 SentiWordNet

SentiWordNet is a lexical resource designed to assist in sentiment analysis and opinion mining tasks [14]. SentiWordNet provides a term level opinion polarity which is derived from the WordNet database of English terms and relations in semi-automatic fashion .

3.1.1 WordNet

WordNet is a lexical database for English language developed at Princeton University to realize the nature of semantic relations of terms in English language, where retrieving and exploration of terms is executed according to concepts and their semantic relationships. It has been widely applied to in various natural language processing task.

Presently WordnNet version 3.0 is available which can be searched via a variety of software APIs or web interface. It offers an all-inclusive database of over 150000 distinct terms mapped into more than 117000 varied meanings (WORDNET, 2006). The growth of WordNet was with expansion of its structure subjected to variety of different languages (WORDNET, 2009).

Key Term Relationships

In WordNet, similarity of meaning drives the key relation between the terms. Here, sets of synonyms (synsets) are prepared by assembling the terms together. The general rules that applies to grouping of terms unitedly to form a synset is regardless of whether a term utilized inside of a sentence on a particular connection can be supplanted by another term on the same synset while not altering the sentence's significance. As a straight implication of this approach, terms must be classified by syntactical classes. It is because nouns, adjectives, verbs and adverbs don't seem to be exchangeable inside of a sentence. Moreover, Synsets additionally contain a brief enlightening text shaping its terms – or gloss

– for helping in specifying. This implies, it becomes especially valuable on synsets with solely one term, or with a little number of relations.

Similarly, antonimity is another useful term relationship present in WordNet, which indicates fundamentally conflicting terms. In WordNet there's a difference between direct and indirect antonyms for exceptional instance of adjectives. For example words like dry/wet are considered as direct antonyms, words like weightless/heavy are qualified as indirect antonyms, since they are conceptually opposites. As they belong to synsets, wherever a direct opposite word exists between the terms (light/heavy) however aren't directly correlated.

Another term relations that is frequently present in WordNet is super–subordinate relation also known as ISA relation or hyperonymy, hyponymy. It interfaces additional synsets like {furniture, piece_of _furniture} to progressively particular ones like {bed} and {bunkbed}. In this way, WordNet expresses that the class furniture incorporates bed, which thus incorporates bunkbed; then again, ideas like bed and bunkbed make up the class furniture. Hierarchies of all noun eventually go up the root node { entity}. The relation Hyponymy is transitive: for example, if perennial is a kind of plant, and if plant is a kind of organism, then perennial is a kind of organism. WordNet also differentiates among types (common nouns) and instances (specific persons, geographic entities etc.). Thus perennial is a type of plant, Narendra Modi is an instance of a prime minister. In these hierarchies instances are always represented by terminal nodes. Another term relation exists between synsets such as {car} and {air_bag, bumper}, {plant} and {hood}. Parts are inherited from their superordinate but not inherited "upward" as they may be an attribute only of specific kinds of things rather than the whole class for example cars and motorvehicles have air_bags, but not all kinds of automobiles have air_bags.

In WordNet majority of relations connect words from the same POS. Thus, it comprises of four sub-nets for nouns, verbs, adjectives and adverbs, along with some cross-POS pointers. Cross-POS relations incorporate "morphosemantic" links which exist among semantically similar terms sharing a stem with constant meaning: observe (verb), observant (adjective) observation, observatory (nouns). The semantic role of the noun in relation with the verb has been specified in most of the noun-verb pairs for example, { sleeper,

sleeping_car} is the LOCATION for { sleep} and { painter} is the AGENT of { paint} , while { painting, picture} is its RESULT.

3.1.2 Building SentiWordNet

Expanding on the calibre of connections of WordNet's semantic, sentiment scores for synsets is decided by SentiWordNet utilizing a semi-supervised strategy. By using this strategy a few synset terms known as {paradigmatic} terms are manually labeled and the remaining database determined utilizing an automated tactics. The overall procedure is summarized below[14] :

1. The paradigmatic terms, obtained from the WordNet are manually labeled to positive and negative labels based on opinion polarity.

2. Iteratively, every level is extended by including terms from WordNet, which are connected to effectively marked terms by the following relationship:
   a. Direct antonym
   b. Attribute
   c. Hyponymy
   d. Similarity

3. For each newly added terms find its direct antonym relation and the terms containing directly opposite orientation are added to opposite level.

4. Step 2 and 3 are repeated for a fixed number of times N.
   After completion of step 1-4, a subset of WordNet synsets can have the label of positive or negative. Based on their synset glosses, a set of classifiers is trained or textual definitions of each synset meaning available on WordNet in order to find the score for all terms. The process of classifying the new entries continues according to the training data and generates score for each term, as detailed below:

5. A word vector is produced for each labeled synset from steps 1-4 along with a positive/negative label. The training dataset used to train the classifier is constructed

as below:

- The prediction e.g. positive/non–positive and negative/non–negative is done by the pair of trained classifier.
- Synset term that lies in both positive and negative class are assigned to "objective" class with and negative score as *zero–valued*.
- This process is repeated for different training set dimensions, which are achieved by variation of N in preceeding stages:0,2,4 and 6.
- Support Vector Machine and Rocchio algorithm are used for each training set.

6. When new terms are subjected to set of classifiers, a prediction score is generated by each resulting classifier. These scores are added and normalized to 1.0 for producing the final score(negative and positive) for each term.

The methodology outlined above for building SentiWordNet demonstrates the reliance of term scores on two specific parts: the choice of paradigmatic words that will create the starting arrangement of negative and positive scores which must be selected carefully, subsequently the rest of the wordNet term score depends on these paradigmatic terms for selecting a score for each term.

3.2 SentiWordNet Database

As discussed in previous section, SentiWordNet can be defined as a database having scores for words obtained from the WordNet database version 2.0 [16]. It is constructed employing a semi–supervised strategy to get polarity scores from a set of seed words that are famed to hold opinion polarity. Every set of terms having a similar meaning, or synsets, is related to three numerical scores starting from zero to one, each demonstrating the synset's objective, positive and negative score. One vital property of SentiWordNet is that, for any given term positive and negative scoring is ranked, and it is feasible for a term to have non-zero values for every negative and positive scores, in accordance with the subsequent principle:
For a synset s we define the following

- $Obj(S) \rightarrow$ *objective score for synset s*

- $Pos(S) \rightarrow$ *Positive score for synset s*
- $Nes(S) \rightarrow$ *negative score for synset s*

We apply the following rules :

$Obj(S) + Pos(S) + Neg(S) = 1$

In order to imply the objectiveness using following relation, value of positive and negative scores are always given:

$Obj(S)$= 1- $(Pos(S) + Neg(S))$

3.2.1 Database Structure

This lexical resource is supplied as a text file where scores for each term are categorized by synset and the relevant Part–of–Speech. Table 3.1 depicts the sections for a single entry in SentiWordNet database reflecting polarity information of a synset.

Table 3.1: Record Structure of SentiWordNet Database

Field	**Description**
POS	Part of speech associated with synset. Four possible values can be considered: a= adjective n= Noun v=Verb r=adverb
Offset	Numerical ID that is connected with Part–of–Speech uniquely distinguishes a synset in the database.
PosScore	Positive score for this synset.This can be a numerical worth starting from zero to one.
NegScore	Negative score for this synset. This can be a numerical value starting from zero to one.
SynsetTerms	List of all terms enclosed in this synset.

To outline how sentiment data shows up in SentiWordNet database, Table 3.2 Presents few sample rows obtained from the raw database file

Table 3.2: Sample SentiWordNet Data

POS	Offset	PosScore	NegScore	SynsetTerms
a(adjective)	00336831	0.25	0.5	sure, certain
n(noun)	13869547	0.25	0.0	hook, hook_shot, crotchet
v(Verb)	00155143	0.375	0.0	rise, go_up, climb
r(Adverb)	00090897	0.25	0.375	intently

3.3 Considerations on SentiWordNet Data

After examining the structure of SentiWordNet database, in this section we inspect the fundamental aspects that need to be consider for designing features that can be used for sentiment classification.

3.3.1 Part of Speech Tagging

Information in SentiWordNet is ordered by POS. Thus as seen on Table 3.2, there are significant variations within the level of objectivity a synset might convey, based on its grammatical role. For classifying a source document we need to extract the Part–of–Speech information, so that scores from SentiWordNet database is accurately applied. To accomplish this, automatic classification of the words into classes based on POS from the source documents is accomplished using a POS tagging algorithmic rule.

For a Part–of–Speech tagger, input is a text and output is a document where each word and punctuation mark is tagged with a label that shows the POS of a given term. For example, the input sentence:

"take care of my cat offers a refreshingly different slice of asian cinema"
produces the following outcome from a Part–of–Speech tagger:
('take', 'VB'), ('care', 'NN'), ('of', 'IN'), ('my', 'PRP'), ('cat', 'NN'), ('offers', 'NNS'), ('a', 'DT'), ('refreshingly', 'RB'), ('different', 'JJ'), ('slice', 'NN'), ('of', 'IN'), ('asian',

'JJ'), ('cinema', 'NN')

Every term is related to a appropriate tag demonstrating its characteristics within a sentence, for example, adjective, verb, noun, and so on. Numerous models exist for POS tag formats, out of which the most common are related to the Penn Treebank explained corpus [17] and the varied occurrences of the CLAWS tag sets, obtained from the original tag set for the brown corpus [18]. To show the above illustration, Table 3.3 focuses key tags from the Penn Treebank tag set pertinent to SentiWordNet. In our experiment we have used the following Penn Treebank tags to access the score from SentiWordNet.

Table 3.3: Penn Treebank Tags

Part of Speech	**Penn Treebank Tags**
Verb	VB, VBP (Present tense), VBZ (Present tense 3rd person), VBG (Gerund), VBD (Past tense), VBN (Past participle).
Noun	NN, NNS(Plural), NNP (Proper noun), NNPS(Proper noun, plural).
Adjective	JJ, JJS (Superlative), JJR (Comparative).
Adverb	RB, RBS (Superlative), RBR (Comparative).

In order to use the POS information to extract score from SentiWordNet database the output of Part–of–Speech tagger must be parsed. This methodology demands an advance application, which reads a tagged text produce by POS tagger, effectively match terms and their Part–of–Speech tag to a SentiWordNet score. In our experiment our program reads a tagged text and exactly match words and their POS tag to a SentiWordNet score.

3.3.2 Word Sense Disambiguation

Word sense disambiguation is a classical NLP problem where a given term has multiple meaning. When we assessed scores for a word utilizing SentiWordNet, a problem emerges in deciding which WordNet synset the term fits in with and which score to consider for evaluation. Consider the case for the expression "break" as noun, with fourteen synsets in WordNet, some of its SentiWordNet score is shown in Table 3.4.

Table 3.4: Example of multiple scores for the same term in SentiWordNet

Synset	**Sentiwordnet Score(pos, obj, neg)**	**Gloss**
happy_chance,good_luck,break	(0.5,0.25,0.25)	"he finally got his big break"
interruption,break	(0.0, 1, 0.0)	"the telephone is an annoying interruption"; "there was a break in the action when a player was hurt"
suspension,pause,intermission	(0.125,0.875, 0.0)	a time interval during which there is a temporary cessation of something
fracture	(0.0,0.75,0.25)	"it was a nasty fracture"; "the break seems to have been caused by a fall"
shift,geological_fault, faulting, fracture	(0.0,0.1,0.0)	"they built it right over a geological fault"; "he studied the faulting of the earth's crust"

As there are different choices of meaning for the noun "break" is available, so determining which synset is best suitable for that particular situation is similar to the problem of WSD. There are no sophisticated techniques for handling word sense disambiguation. In our experiment we have used a strategy for handling WSD that is discussed in next section.

3.4 Proposed Model

In the past section of this chapter, the structure of the SentiWordNet database are explained in details and issues were created on challenges and limitations of what opinion data need to be gathered. Keeping those in mind we have proposed a model for making an arrangement of set of features for sentiment analysis utilizing SentiWordNet. The methodology for a list of features suggested in this section however begins from the rule that the features acquired through SentiWordNet catch different facets of tweets sentiment and are best suited to train any classifier algorithm. In our model we have used some strategy to eliminate two key problem of sentiment analysis task: handling WSD and negation.

3.4.1 Handling WSD

Word sense disambiguation is a classical natural language processing problem where a single term can have multiple meaning and we need to find out the exact meaning for that particular context. For example consider the sentence "Red Tape holds up the bridge" here the term "holds up" has two meanings: one is supporting to build the bridge another is delay in building the bridge. So in different context it has different meanings. Similarly the term "low price" in some context is positive but in other context it may be negative. So to handle WSD in our experiment Part–of–Speech tagging is executed for extracting scores from SentiWordNet. If a term has multiple scores then we have considered the average score of syn-sets for that particular term. So in our model we followed an easier methodology based on the following rules:

Average of Max of Pos and Neg Score

- For each synset of a given term find its score from SentiWordNet database.
- If for the same term, there prevails both positive and negative scores — compute the average of all positive and negative scores.
- Return the average score with maximum value.

$$score\,(t) = \max\left(\frac{\sum_N Pos\,(t_n)}{N}, \frac{\sum_N Neg\,(t_n)}{N}\right) \tag{3.1}$$

where,

$Pos(w_n)$ = positive score of SentiWordNet for N^{th} term t

$Neg(w_n)$ = Negative score of SentiWordNet for N^{th} term t

N = Number of synset for the given term

Weighted Average of all Synsets

It has been found that in WordNet, terms in a synset are organized according to their frequency of utilization in that specific sense. So the score that each sense provides will be

scaled by the position on the word within the synset and the total number of words within the synset [19]. This gives rise to the subsequent formula for scoring a term :

$$score\,(t) = \frac{\sum_N weight_n * \max\,(Pos\,(t_n)\,, Neg\,(t_n))}{\sum_N weight_n} \tag{3.2}$$

where the weight is given by:

$$weight_n = 1 - \frac{position\ of\ term\ t\ in\ N^{th} synset}{Number\ of\ words\ in\ N^{th} synset} \tag{3.3}$$

It has been found that in WordNet synset the most frequently occurring word is always present at first position. So weight of a most frequently occurring word is 1 as position of a term starts from zero.

3.4.2 Handling Negation

Negation is a grammatical category that allows the changing of the truth value of a propositions. It is often expressed through the use of negative signal or negators. Words like “isn’t”, “no” “never” and “not” can significantly affect the sentiment orientation of a term. Consider the following example:

- People like the change.
- People don’t like the change.

Clearly, both the sentence have the term “like” which carries positive sentiment and positive score in SentiWordNet. However second sentence gives a negative meaning. Therefore a evaluation methodology that merely adds scores for terms as they seem on text can result in poor results. Previously, for handling negation a conventional method is used called reversing hypothesis which is given by the following formula:

$$S\,core\,(n, w) = -S\,core\,(w) \tag{3.4}$$

where,

$S\,core\,(w)$ is the sentiment of word or phrase w

$S\,core\,(n, w)$ is the sentiment of the expression formed by concatenation of the negator n

and word w.

For example, if $Score\,(honest) = 0.9$ then $Score\,(not, honest) = -0.9$. For handling negation in our experiment we have used the *NegEX* algorithm proposed by Chapman *et al.*. This algorithm works by identifying three classes of expressions: There are certain negation expression which doesn't alter the orientation of sentiment bearing words called pseudo-negating terms; and certain expression are there that alter the orientation of previous and next term in a sentence. When a negation expression is found then the sentiment orientation of a sentence is inverted for all terms among a selected window, or till a punctuation is found. Here window size is a numeric parameter that indicates the scope of a negating term within a sentence. Output of this *NegEX* algorithm determines what number of terms are being negated and how often negating expressions are used as a narrative device. We can use this as a feature to find the sentiment orientation of a sentence.

3.4.3 Automatically Creating Sentiment Lexicon

In our experiment we have found that accuracy result of sentiment classification using SentiWordNet varies from domain to domain. In some domains it gives better result where as in some other domain results is poor this because sentiment score of domain related words are not there in the predefined lexicons. So it is better to generate our own lexicon from the corpus itself. As we are finding the sentiment of tweets, our corpus contains large number of *emoticons* and *hash-tag* words whose sentiment scores are not in SentiWordNet lexicon. We have used the pointwise mutual information (PMI) formula given by *Turney et al.* [13] for generating domain specific lexicon.

For every term/word t in the set of millions of tweets an association score is generated based on the following formula:

$$score\,(t) = PMI\,(t, positive) - PMI\,(t, negative) \tag{3.5}$$

where PMI is given by the formula:

$$PMI\,(t_1 \wedge t_2) = \log\left(\frac{p\,(t_1 \wedge t_2)}{p\,(t_1) * p\,(t_2)}\right) \tag{3.6}$$

where,

$p(t_1 \wedge t_2)$ is probability of how often t_1 t_2 co-occur.

$p(t_1)$ is probability of occurance of t_1.

$p(t_2)$ is probability of occurance of t_2.

If $score(t) > 0$ then word w is classified as positive. If $score(t) < 0$ then word w is classified negative. A snapshot of our lexicon generated from the tweets collected using twitter API is shown in Figure 3.1.

```
>>> ================================ RESTART ================================
>>>
{'forget': 7.7005, 'tweettext': 7.7005, 'androids': 7.7005, 'soon': 7.7005, 'tha
nks': 7.7005, 'bunch': 6.7005, 'adding': 7.7005, 'referring': 7.7005, 'la': 6.11
55, 'lo': 7.7005, 'fix': 7.7005, 'imo': 7.7005, 'failed': 6.7005, 'add': 6.7005,
 'offers': 7.7005, 'removable': 7.7005, 'crack': 7.7005, 'integrate': 7.7005, 'p
resentation': 7.7005, 'match': 7.7005, 'lack': 7.7005, 'crash': 7.7005, 'di': 7.
7005, 'beat': 7.7005, 'de': 6.7005, 'sums': 7.7005, 'hilarious': 7.7005, 'gonna'
: 7.7005, 'ouch': 7.7005, 'facial': 6.1155, 'dando': 7.7005, 'hemp': 7.7005, 'da
y': 6.7005, 'instructions': 7.7005, 'unidos': 7.7005, 'found': 7.7005, 'estados'
: 7.7005, 'quarter': 6.7005, 'looks': 7.7005, 'google': 6.1155, 'art': 7.7005, '
phones': 7.7005, 'twitter': 6.7005, 'hard': 7.7005, 'whens': 7.7005, 'unlock': 7
.7005, 'video': 7.7005, 'liking': 7.7005, 'specs': 7.7005, 'recognition': 5.7005
, 'rt': 5.7005, 'living': 7.7005, 'jewelry': 7.7005, 'giving': 7.7005, 'literall
y': 7.7005, 'ulteriore': 7.7005, 'sms': 7.7005, 'ice': 6.1155, 'innovation': 7.7
005, 'del': 6.7005, 'staff': 7.7005, 'missing': 7.7005, 'burn': 7.7005, 'reason'
: 7.7005, 'que': 7.7005, 'jetzt': 7.7005, 'swype': 7.7005, 'guy': 7.7005, 'evolu
zione': 7.7005, 'agree': 7.7005, 'itunes': 7.7005, 'solco': 7.7005, 'eta': 7.700
5, 'products': 7.7005, 'disappointment': 7.7005, 'social': 7.7005, 'con': 6.7005
, 'named': 7.7005, 'love': 7.7005, 'battery': 7.7005, 'vittoria': 7.7005, 'siri'
: 7.7005, 'groupon': 7.7005, 'son': 7.7005, 'droid': 7.7005, 'inbox': 7.7005, 'c
enan': 7.7005, 'viu': 7.7005, 'ios5': 6.7005, 'creato': 7.7005, 'exciting': 7.70
05, 'che': 7.7005, 'removing': 7.7005, 'support': 7.7005, 'hey': 7.7005, 'los':
7.7005, 'iphone': 5.1155, 'estava': 7.7005, 'sucks': 7.7005, 'abbia': 7.7005, 'w
ar': 7.7005, 'ontem': 7.7005, 'buy': 7.7005, 'easy': 7.7005, 'pittsburgh': 7.700
5, 'brand': 7.7005, 'releasing': 7.7005, 'else': 7.7005, 'notifications': 7.7005
, 'ho': 7.7005, 'dios': 7.7005, 'stellt': 7.7005, 'mb': 7.7005, 'cap': 7.7005, '
iteration': 7.7005, 'unveils': 7.7005, 'record': 6.7005, 'un': 7.7005, 'ui': 7.7
005, 'offiziell': 7.7005, 'fun': 7.7005, 'duet': 7.7005, 'guessers': 6.7005, 'vc
': 7.7005, 'security': 7.7005, 'nexus': 7.7005, 'site': 7.7005, 'revamps': 7.700
5, 'vor': 7.7005, 'ship': 7.7005, 'apparently': 7.7005, 'tells': 7.7005, 'shit':
 7.7005, 'cream': 6.1155, 'parte': 7.7005, 'recente': 7.7005, 'latter': 7.7005,
'storage': 7.7005, 'feature': 7.7005, 'note': 7.7005, 'read': 7.7005, 'amazing':
 7.7005, 'carrier': 7.7005, 'sabias': 7.7005, 'android': 6.7005, 'incl': 7.7005,
 'tool': 7.7005, 'nice': 7.7005, 'week': 7.7005, 'sandwich': 6.1155, 'disappoint
men': 7.7005, 'finale': 7.7005, 'update': 6.7005, 'paid': 7.7005, 'dos': 7.7005,
 'usuarios': 7.7005, 'realized': 7.7005, 'desayunan': 7.7005, 'phone': 6.7005, '
hello': 7.7005, 'switch': 7.7005, 'hella': 7.7005, 'compete': 7.7005, 'time': 7.
7005, 'professional': 6.7005, 'pretty': 7.7005, 'scroll': 7.7005, 'jajajajajajaj
                                                                          Ln: 11 Col: 4
```

Figure 3.1: Snapshot of the lexicon generated from tweet corpus

We have also created a model which doesn't require any training data to train the classifier or any corpus to generate domain specific lexicon. Here we have used NLTK Part–of–Speech tagger to find the adjective, verbs and Nouns for a given tweet. For each word in the adjective, verb, noun list we pulled the Google search engine to find number of hits with an extremely positive word "excellent" and number of hits with an extremely negative word "poor". A score is generated for each word by using the following rule:

$$score\,(t) = \log\left(\frac{hits\,(t \wedge excellent) * hits\,(poor)}{hits\,(t \wedge poor) * hits\,(excellent)}\right) \tag{3.7}$$

Based on the score of each word a weight is given to each sentence/tweets. If the weight of the sentence is positive then sentence is classified as positive, otherwise classified as negative.

3.5 Results and Discussion

We have used SentiWordNet lexicon for classification of tweets and compare the results with movie review data set. We have used NegEX algorithm discussed in previous section and both Weighted average and Average of Max of Pos and Neg score approach to handle WSD. In order to conduct our experiment each tweet or movie review is scanned and every term would get a score in view of SentiWordNet information and Part–of–Speech tagging. We have used NLTK Part–of–Speech tagger to find POS of movie reviews and tweets and Penn Treebank tags to access the score from SentiWordNet as discussed in Section 2.3. We have given a positive weight and negative weight to each tweet or review based on the scores received from SentiWordNet data base. If positive weight is greater than negative weight we classify the tweet into positive class otherwise it is classified to negative class. We applied the above approach to our Polarity dataset and tweets retrieved from twitter public domain using keyword Microsoft, Google, Twitter, Apple. The results of comparison are shown in Figure 3.3. A snapshot of our program which extract scores for a single movie review using SentiWordNet lexicon is shown in Figure 3.2.

When we used the model where Google search engine is pulled to find the score for each term present in adjective, noun, verb list as discussed in Section 2.4 we got an accuracy of 80.68% for our tweets corpus.

```
>>> ================================ RESTART ================================
>>>
[('steers', 'NNS'), ('snappy', 'VBP'), ('screenplay', 'NN'), ('curls', 'NNS'), (
'edges', 'NNS'), ('clever', 'VBP'), ('hate', 'JJ'), ('somehow', 'NN'), ('pulls',
 'NNS'), ('care', 'VBP'), ('cat', 'JJ'), ('offers', 'NNS'), ('refreshingly', 'RB
'), ('slice', 'NN'), ('asian', 'JJ'), ('cinema', 'NN')]
List of adjective:
['hate', 'cat', 'asian']
list of noun:
['steers', 'screenplay', 'curls', 'edges', 'somehow', 'pulls', 'offers', 'slice'
, 'cinema']
list of verb:
['snappy', 'clever', 'care']
[]
[]
[SentiSynset('asian.a.01')]
[SentiSynset('tip.n.03'), SentiSynset('bullock.n.02')]
[SentiSynset('screenplay.n.01')]
[SentiSynset('coil.n.02'), SentiSynset('curl.n.02'), SentiSynset('lock.n.02')]
[SentiSynset('edge.n.01'), SentiSynset('boundary.n.02'), SentiSynset('edge.n.03'
), SentiSynset('edge.n.04'), SentiSynset('edge.n.05'), SentiSynset('edge.n.06')]
[]
[SentiSynset('pull.n.01'), SentiSynset('pull.n.02'), SentiSynset('pull.n.03'), S
entiSynset('pull.n.04'), SentiSynset('wrench.n.01'), SentiSynset('puff.n.07'), S
entiSynset('pull.n.07')]
[SentiSynset('offer.n.01'), SentiSynset('offer.n.02'), SentiSynset('crack.n.09')
]
[SentiSynset('slice.n.01'), SentiSynset('piece.n.08'), SentiSynset('cut.n.05'),
SentiSynset('slice.n.04'), SentiSynset('slice.n.05'), SentiSynset('slice.n.06')]
[SentiSynset('film.n.02'), SentiSynset('cinema.n.02')]
[]
[]
[SentiSynset('care.v.01'), SentiSynset('care.v.02'), SentiSynset('wish.v.02'), S
entiSynset('manage.v.02'), SentiSynset('worry.v.02')]
POSITIVE: steers turns in a snappy screenplay that curls at the edges it s so cl
ever you want to hate it but he somehow pulls it off take care of my cat offers
a refreshingly different slice of asian cinema  positive_score =0.625
```

Figure 3.2: Part of speech collected from a movie review

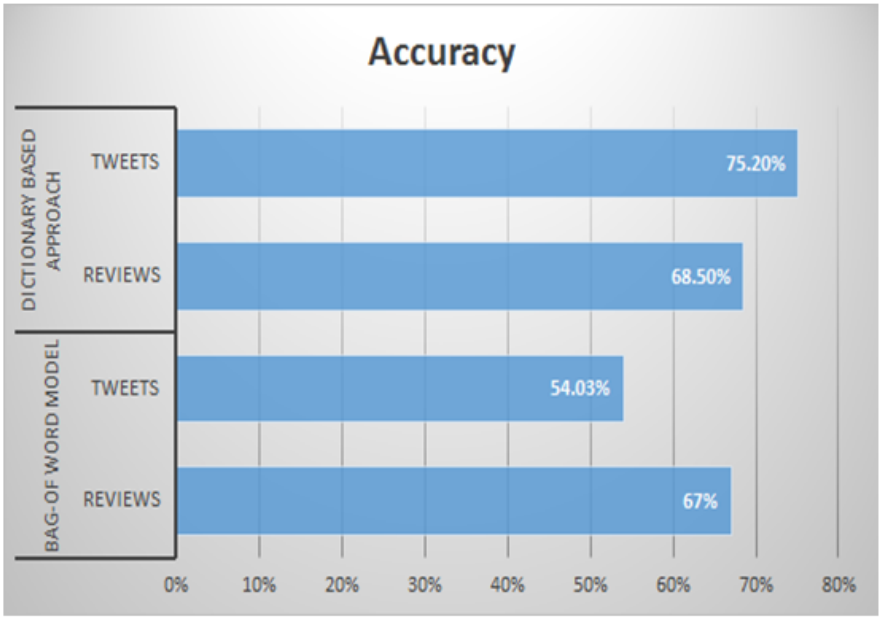

Figure 3.3: Sentiment score comparison between polarity dataset and tweets

It has been found that this approach not only handle Word sense disambiguation but also able to handle another challenge in sentiment analysis which is sudden deviation from positive sentiment to negative sentiment. A snapshot of the result of such a tweet where there is a sudden change in polarity is shown in Figure 3.4.

Table 3.5: Result comparison between SentiWordNet Lexicon based approach and Proposed approach

Approaches	Dataset	Accuracy	Domain
SentiWordNet Lexicon	reviews	68.50 %	movie reviews
	tweets	**75.20** %	Google,Apple,microsoft,Twitter
Proposed Approach	reviews	NA	movie reviews
	tweets	**80.68** %	Google,Apple,microsoft,Twitter

```
>>> ================================ RESTART ================================
>>>
great battery life perfect size but a tid bit quieter than i would like
['great', 'battery', 'life', 'perfect', 'size', 'tid', 'bit', 'quieter', 'would'
, 'like', '']
[('great', 'JJ'), ('battery', 'NN'), ('life', 'NN'), ('perfect', 'NN'), ('size',
 'NN'), ('tid', 'VBD'), ('bit', 'NN'), ('quieter', 'NN'), ('would', 'MD'), ('lik
e', 'VB'), ('', '-NONE-')]
List of adjective:
['great']
list of noun:
['battery', 'life', 'perfect', 'size', 'bit', 'quieter']
list of verb:
['tid', 'like']
6269140000000000
6643500000000000
3707200000000
6942200000000
4958380000000000
6715600000000000
-2.32583732244
>>> |
```

Figure 3.4: Snapshot of the tweet weight of our model

3.6 Conclusion

In this chapter, we have analyzed the structure of SentiWordNet database in more detail, with the goal of deciding how to best utilize SentiWordNet to construct a model that represents sentiment information from text documents. This chapter highlighted the requirement to avail of Natural Language Processing techniques like Part-of–Speech tagging to complement the model. We have also discussed about the limitation of SentiWordNet lexicon and how to overcome this by creating domain specific lexicon from the test corpus. From our experiment, we conclude that creating domain specific lexicon and using it for sentiment identification gives better result than using SentiWordNet lexicon.

Chapter 4

Sentiment Analysis using Machine Learning Techniques

According to Samuel (1959) machine learning is the *field of study that gives computers the ability to learn without being explicitly programmed*. Using this definition, machine learning can be appropriately applied to the problem of text classification, and by way of inheritance, can duly be related to sentiment analysis. What can be drawn from the literature review is that machine learning techniques have the potential to contribute an efficient solution to the problem of sentiment analysis. Both supervised and unsupervised machine learning approaches have been applied to the challenge of sentiment analysis, and for some limited domains that exhibit little topical variation, performance has been good. In this chapter, we have applied three different supervised algorithm (Naive Bayes (NB), Maximum Entropy (ME) and Support Vector Machines (SVM)) for sentiment identification of tweets, to study the effectiveness of various feature combination.

4.1 Supervised Methods

A supervised learning algorithm requires training data to train the model, which examines the training data and generates an inferred function, that can be used for mapping new samples. An ideal situation will permit for the algorithm to accurately determine the class labels for unseen examples. This needs the learning algorithm to generalize from

the training data to unseen situations in a “reasonable” way.

This is where most of the time and effort was spent. Under this section, different supervised machine learning approaches are used. A three step process was used to conduct our experiment with different machine learning algorithm and to identify the factors that affect the results. Following are the steps to be followed before applying any of the machine learning algorithm.

- **Step1** Preprocessing the training data
- **Step2** Feature extraction and data representation
- **Step3** Training the classifier and testing with different machine learning algorithms

4.1.1 Preprocessing Training Data

- **Cleaning training data** For begin, the subsequent improvement operations are done on the training data. We have followed the preprocessing steps defined in section 3.4 of chapter 3. All URLs have been removed from tweets. All special character and stop words are removed from tweets. The hashtag(#), a symbol used to indicate nouns, has been removed. The word RT, used for retweets, has been removed. Tweets that contain both negative and positive emoticons are removed from the dataset to avoid confusing the machine learning algorithm.
- **Removing duplicate tweets** It has been observed that, tweets that are retrieved for training and testing the classifier contain duplicate tweets. This is because the same tweet is twittered and retweeted. So we need to remove the duplicate tweets. Only exact duplicate tweets are removed from the training corpus.

4.1.2 Feature Extraction

Before applying any machine learning algorithm one question is how to represent the data. At the point when input data to a learning algorithm is too vast to be processed and it is expected to be redundant, then the input data can be transformed into a reduced set of features. This process is called *feature extraction*. The features extracted from

input data are expected to contain the relevant information, so that the desired assignment can be performed by utilizing this reduced representation rather than the complete initial information. In order to build a model using machine learning algorithm both the training and test data must be represented in some way. Most of the machine learning algorithms used Bag-of-Words (BoW) representation as their feature. In machine learning feature means some attributes that are thought to capture the pattern of the data are initially chosen and the whole dataset must be depicted in terms of them before it is supplied to any machine learning algorithm. Variety of features like unigram, bigram, part-of-sppech(POS), syntactic and semantic feature are used for sentiment analysis.

In BoW representation a tweet is represented as the bag of its words and phrases, disregarding grammar and even word order but keeping multiplicity. But in case of set-of-tweets representation of a Twitter post if a word occurs twice, it will only be present once regardless of how many times it is found in the Twitter post. In this chapter bag-of-word and feature representation are used for different purposes at different stages.

- **Attribute selection** It is the method of extracting features by that the data are going to be delineated. In machine learning algorithm for representing data instances attribute selection is considered as the first task to be performed. Once attributes are selected, the training or testing data will be described using these attributes. So attributes are the features. For attribute selection we have converted the entire dataset into uni-grams or bigrams or trigrams or a combination of any of them. Bag-of-uni-grams is equivalent to BoW. During attribute selection we need to think about which words to exclude from being selected as attributes. As there are certain words which do not contribute to polarity detection, we have to remove such attributes. This is done by using stop words.

- **Instance representation** Once attributes are chosen, the data must be represented in terms of these attributes. Now these attributes are called as features. A decision of whether to utilize uni-gram presence or uni-gram frequency, bi-gram presence or bi-gram frequency, tri-gram presence or tri-gram frequency, or a mixture of uni-gram + bi-gram presence or frequency, and so on must be taken. In spite of the reality we have utilized the whole information set as a part of selection of attributes, the

representation of the information must be done on every each Twitter post.

4.1.3 Training and Testing the Classifier

For training and testing the classifier we need to find the feature vector. This is because feature vector is the most important concept in implementing a classifier. A good feature vector directly determines how successful the classifier will be. A feature vector is just a vector that contains information describing an object's important characteristics. In our case feature vectors contains features such as unigram presence, bigram presence or combination of both etc. The entire feature vector will be a combination of feature words of each data instances(Twitter post). We have trained the classifier using these feature vector. For training the classifier we have used three different machine learning algorithm that are described in next section.

For testing a tweet, we need to find out the feature words and we get one more pattern of feature vector, based on the model learned the classifiers predict the tweet sentiment.

4.2 Machine Learning Algorithm

Our aim in this thesis is to find out the effective feature for sentiment classification of tweets being positive sentiment or negative sentiment. We have experimented with three standard classifier: Multinomial Naive Bayes classifier with Laplace smoothing, Maximum Entropy classifier and Support Vector Machine classifier. The philosophies behind these three algorithms are quite different, but each has been shown to be effective in previous classification studies.

We used the standard bag-of-feature frame work for implementing these machine learning algorithms. Let $\{w_1, ..., w_m\}$ be the m words that can appear in a tweet/sentence; examples include the word silent or the bigram low quality. Let $n_i(d)$ be the number of times w_i occurs in tweet t. Then, each tweet t is represented by the tweet vector $t := (n_1(t), n_2(t), ..., n_m(t))$

4.2.1 Naive Bayes

The Naive Bayes classifier is a simple probabilistic classifier which is based on Bayes theorem. Naive Bayes performs well in many complex real-world problems and it is one of the basic text classification techniques with various applications in email Spam detection, personal email sorting, document categorization etc. Naive Bayes classifier is very efficient as it is less computationally intensive (in both CPU and memory) and it requires a small amount of training data. One approach to classify tweets is to assign to a given tweet t the class $c^* = argmax_c P(c \mid t)$. We derive the Naive Bayes (NB) classifier by first observing that by Bayes rule,

$$p(c \mid t) = \frac{p(c)p(t \mid c)}{p(t)} \tag{4.1}$$

where $p(t)$ plays no role in selecting c^*. For estimating the term $p(t \mid c)$, Naive Bayes breaks down tweet by assuming the w_i's are conditionally independent given t's class. The most likely class according to the Naive Bayes classifier is the class among all classes which maximizes the product of two probabilities prior and likelihood, the word in a tweet given the class i.e. how often that word is expressed in a positive tweets or in a negative tweets

$$C_{NB} = \arg\max_{c_j \in C} p(c_j) \prod_{i \in positions} p(w_i \mid c_j) \tag{4.2}$$

Research on Sentiment Analysis tells that word occurrence may matter more than word frequency. As tweets are 140 character length occurrence of a word tell us a lot, but the fact that if it occurs more than once may not tell us much more. So we need to clip all the word count in each tweet at one and remove duplicate words in each tweet to retain a single instance of the word. So for our work we have used another variant of Naive Bayes classifier i.e. binarized (Boolean feature) Multinomial Naive Bayes classifier which assumes the features to be occurrence of count. The reasoning behind this is often that the occurrence of the word matters over the word frequency and so weight it multiple times doesn't improve the accuracy of the model.

Laplace Smoothing.

Here we used Laplace smoothing assuming that even if we have not seen a given word in the whole corpus, there is still a chance that our sample of tweets happened to not include that word.

$$\hat{p}(w \mid c) = \frac{count(w, c) + 1}{count(c) + \mid V \mid} \tag{4.3}$$

4.2.2 Maximum Entropy

The maximum entropy classifier is a probabilistic classifier which belongs to the class of exponential models which has proven effective in a number of natural language processing applications. Nigam *et al.* [20] show that it sometimes, but not always, outperform Naive Bayes at standard text classification. It does not assume that the features are conditionally independent of each other. Here our target is to use the contextual information of the tweets (unigram, bigram, and other characteristics) within the text in order to categorize it to a given class (positive or negative). Maximum entropy estimates of takes the following exponential form [12]:

$$P_{ME} = \frac{1}{Z(t)} \exp\left(\sum_i \lambda_{i,c} F_{i,c}(t, c)\right) \tag{4.4}$$

where $Z(t)$ is the size of the training dataset used as a normalization function . $F_{i,c}$ is a indicator function for feature f_i and class c, defined as follows,

$$F_{i,c} = \begin{cases} 1, & n_i > 0 \text{ and } c' = c \\ 0, & \text{otherwise} \end{cases}$$

This binary valued indicator function returns 1 only when the category of a specific tweet is c_i and also the tweet contains the word w_k.

Maximum entropy classifier takes more time to train comparing to Naive Bayes classifier primarily due to the optimization problem that needs to be solved in order to estimate the parameter of the model. For estimating the λ parameters we use ten iteration of IIS(improved iterative scaling) algorithm (this was a sufficient number of iterations for convergence of training-data accuracy), together with a Gaussian prior to counteract over

fitting.

4.2.3 Support Vector Machine

This algorithm works in a completely different way from the above algorithm. Support Vector Machines (SVMs) are widely used for various text categorization in past, usually outperforming Naive Bayes classifier. In all form of Baye's algorithm missing values are ignored but support vector machine replaces them globally. In case of two-class problem with d dimension, the basic idea is to search out a hyperplane, represented by vector $\overrightarrow{w}$, that not just differentiates the tweet vectors in one category from those in alternative, yet for which the separation, or margin, is as large as attainable[12]. Let positive and negative be the correct class of tweet t_j and $c_j \in \{1, -1\}$ refers to the class labels positive and negative, then searching a hyperplane corresponds to a constrained optimization problem; where the solution is described as,

$$\overrightarrow{w} = \sum \alpha_j c_j \overrightarrow{t_j} \qquad , \alpha_j \geq 0 \tag{4.5}$$

where the α_j's are derived by solving a dual optimization problem. The tweet vectors t_j are called support vectors for which α_j is greater than zero, as these tweet vectors contribute to the hyperplane. Classification of tweets includes primarily deciding that facet of $\overrightarrow{w}$'s hyperplane they fall on.

4.3 Experimental Set-up

The core component of our system is shown in Figure 4.1. We proceed to discuss this in detail.

For implementation we have used our data set which contains 6000 hand classified tweets. The corpus contains tweets about apple, goggle, Microsoft and twitter. Tweets are classified into four classes positive, negative, neutral and irrelevant. Irrelevant tweets are those tweets that are not in English language or not related to the topic. In our experiment we have considered three classes positive, negative and neutral. So we converted all irrelevant class to neutral class. The polarity data set is a set of film review documents available for research in Sentiment Analysis and opinion mining. It was first introduced as

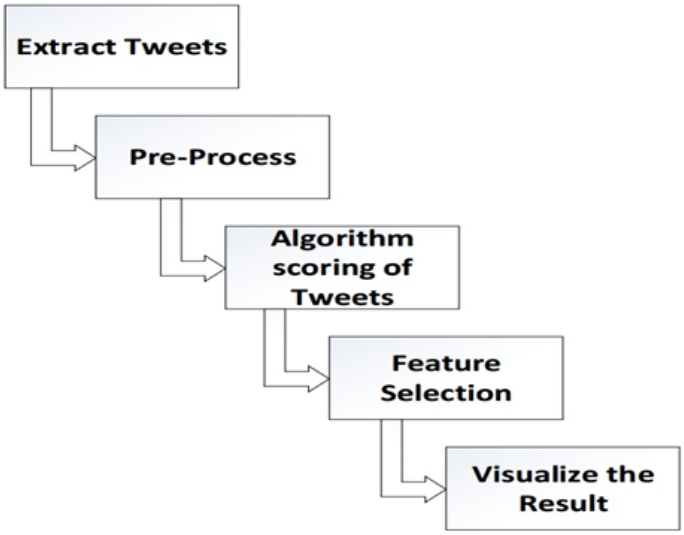

Figure 4.1: Block diagram of proposed experiment

a research data set along with Bo Pang and Lillian Lees initial results on machine learning methods for sentiment classification. The most recent available data set is version 2.0. It comprises 1000 positive labeled and 1000 negative labeled film reviews extracted from the Internet Movie Database Archive. In our experiment we have done a comparative study between the polarity data set used by Pang and Lee and our dataset. We have also generated a model that can efficiently identify the emotion of a tweets.

We have selected Multinomial Naive Bayes (MNB), Maximum entropy (MaxEnt or ME) and Support Vector Machine (SVM) to use, since they are very efficient for handling millions of tweets. We have used python regular expression for data pre-processing. We have employed python Natural Language Toolkit (NLTK 3.0) to get unigram, bigram, adjective and Part of speech (POS) features of tweets. We have used linear SVM for our experiment.

4.3.1 Evaluation Metrics

The overall performance of individual classifier is measured by:

$$accuracy = \frac{\#of\ correctly\ labeled\ tweets}{\#of\ all\ the\ tweets\ in\ the\ test\ dataset} \tag{4.6}$$

Precision

It is a measure of accuracy provided that a specific class has been predicted. It is defined by:

$$Precision = \frac{TP}{TP + FP} \tag{4.7}$$

Recall

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

$$Recall = \frac{TP}{TP + FN} \tag{4.8}$$

where, TP =Number of true positive predictions for the considered class.
FP = Number of false positive predictions for the considered class.
FP = Number of false negative predictions for the considered class.

F1 score

$$F1\ score = \frac{2 \times Precision \times Recall}{Precision + Recall} \tag{4.9}$$

4.4 Results and Discussion

4.4.1 For Twitter Dataset

Table 4.1: Accuracy of tweets using different features

	Features	# of Features	Frequency or Presence	Naive Bayes		Maximum Entropy		Support Vector Machine	
				reviews	tweets	reviews	tweets	reviews	tweets
(1)	unigram	5989	presence	81.0 %	81.5%	80.4%	78.36%	82.9%	82.5%
(2)	bigram	19148	presence	77.3%	78.60%	77.4%	78.0%	77.1%	77.8%
(3)	unigram + bigram	25,748	presence	80.6%	80.92%	80.8%	79.78%	82.7%	81.6%
(4)	Unigram+POS	19061	presence	81.5 %	82.0%	81.2%	80.3%	81.9%	81.99%
(5)	Adjectives	1197	presence	77.0%	69.48%	77.7%	76.4%	75.1%	76.4%

Table 4.2: F1 score of MNB classifier

Class label	**Precision(%)**	**Recall(%)**	**F1 score(%)**
Positive	65.25%	20.51%	31.21%
Negative	77.41%	16.05%	27.20%
Neutral	80.48%	61.82%	69.93%

We explore a variety of features that are potent for Sentiment Analysis. We have used N-gram features like unigrams (n=1), bigrams (n=2) that are widely used in different of text classification, including Sentiment Analysis. In our study we experimented with unigrams and bigrams with boolean features. Each n-gram feature is associated with a boolean value, which is set true if and only if the n-gram is present in the tweet [12]. Table 4.1 represents the different features we have used and the accuracy results of individual classifier. Here we have performed a comparison between the movie review data set used by Pang Lee *et al.* and our dataset. From Table-1, it has been observed that when we used NB classifier with Laplace smoothing, the classification accuracies resulting from using unigram as features gives better result in case of tweets than movie reviews, but when we used MaxEnt classifier the accuracy result of Movie reviews are more than the tweets.

We additionally considered usage of bigrams to capture negation words for handling negation and phrases for dealing with Word Sense Disambiguation (WSD). *Line*(2) of results table demonstrates that using bigram as feature does not improve performance of the classifier as that of unigram presence. In our experiment we observed that, although bigram presence does not improve the classification accuracy it is as equally useful a feature as unigram; in reality bigrams are found to be effective features for handling word sense disambiguation. We also experimented considering bigram as single feature but the results were not as good, but combination of unigram and bigram features (*Line*(3) of results table) produces results competitive with those obtained by using unigram.

POS features are verified effective in Sentiment Analysis . Since adjectives are good indicators of sentiment, they are usually considered as effective feature for Sentiment Analysis. Our experiment shows (*Line*(5) of results table) that considering only adjectives produces results competitive with those obtained by using unigram and bigram. *Line*(4) of

results table shows that all the three classifier produces better result considering unigram and POS as feature. *Line*(1) of results table shows that SVM with unigram as feature produces best result out of all the features we have considered. Table 4.2 represents the detailed results of MNB classifier with F1 score. Figure 4.2 represents the Receiver Operating Characteristic (ROC) curve of MNB classifier for tweets that are manually annotated.

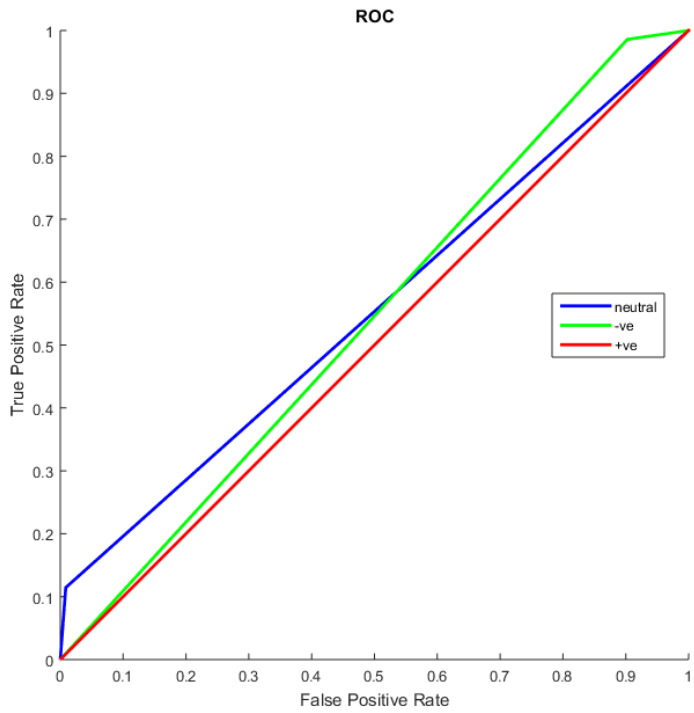

Figure 4.2: ROC curve of MNB classifier for tweets

From our experiment we found that F1 score of positive class and negative class are not good as compared to neutral class. This is because in our data set most of the tweets are manually annotated to neutral and irrelevant class. So in order to apply any of the machine learning techniques we need a training data set that is not biased and also do not need manually annotation. To do this we have collected tweets with emotion hashtags as described in Chapter 2 and apply the MNB classifier on it to find out whether hashtags are useful for emotion identification or not.

4.4.2 Emotion Dataset

People do tend to use *hashtags* to express their sentiment or emotions. So these hashtagged words are good level of sentiments and emotions. We have used these hashtag to add more data to our machine learning algorithm. We have used MNB classifier to our emotion dataset and the result is shown in Table 4.5. A snapshot of the confusion matrix of our emotion dataset for unigram features is shown in Figure 4.3 and F1 score of each class for unigram feature is shown in Table 4.6. ROC curve of our classifier is shown in Figure 4.4.

```
Confusion Matrix
             |                          t |
             |                          h |
             |                          a |
             |                          n |
             |                      s   k |
             |                      u   f |
             |                      r   u |
             |   a                  p   l |
             |   n   f       l      r   n |
             |   g   e   j   o   s   i   e |
             |   e   a   o   v   a   s   s |
             |   r   r   y   e   d   e   s |
-------------+-----------------------------+
       anger |<195>  2   .   2   .   .   . |
        fear |   .<207>  3   .   3   1   . |
         joy |   .   1<189>  .   6   .   . |
        love |   1   .   4<211>  4   .   . |
         sad |   .   1   .   2<179>  .   . |
    surprise |   .   1   .   .   4<203>  . |
thankfulness |   .   6   9   5  10   .  <.>|
-------------+-----------------------------+
(row = reference; col = test)

>>>
```

Figure 4.3: Snapshot of emotion dataset

Table 4.3: Accuracy of emotion dataset using different features

Features	# of features	MNB classifier
Unigram	4635	95.0%
Bigram	17628	71.23%
Unigram+Bigram	35356	95.3%
POS	12443	92.9%
Adjective	1503	84.5%

From our experiment it has been found that preparing a dataset by automatically collecting tweets using hashtags shows its advantage as compared to the data set which

Table 4.4: F1 score of MNB classifier for unigram feature

Class label	Precision(%)	Recall(%)	F1 score(%)
anger	99.48%	97.98%	98.72%
fear	94.95%	96.72%	95.82%
joy	92.19%	96.42%	94.25%
love	95.90%	95.90%	95.9%
sad	86.89%	98.35%	92.26%
surprise	99.5%	97.59%	98.53%

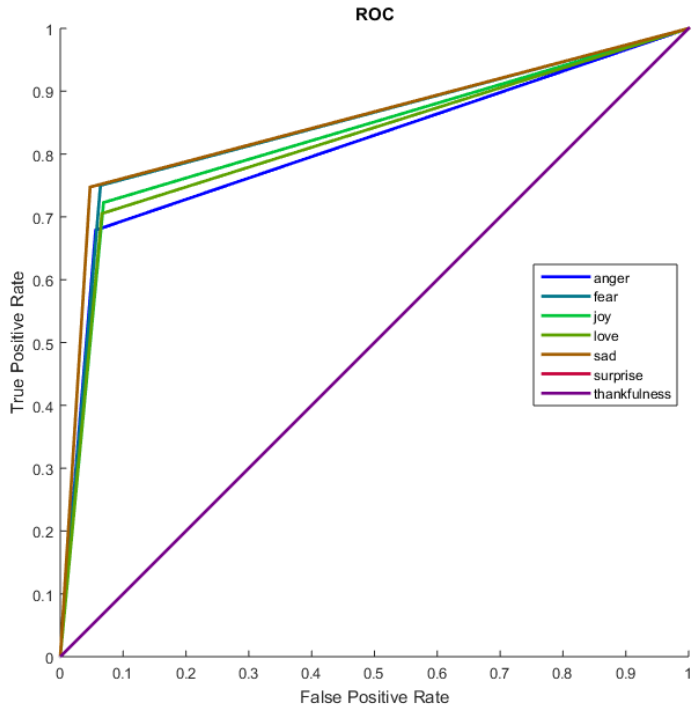

Figure 4.4: ROC curve of MNB classifier for emotion data set

is formed by manually annotation. This is because writers are correct regarding their own emotions, while the traditional means of annotating information needs annotators to infer the writers' emotions from text, which can not be accurate.

4.4.3 SMS Dataset

We have collected some SMS (short message services) and tasted our model with these SMS. The reason behind testing our model with SMS are they are more unstructured than tweets. We have applied Multinomial Naive Bayes classifier and Support Vector machine to our SMS data and the results are shown in Table 4.3. We have used different features and we found that unigram features are more potent for Sentiment Analysis of unstructured text. The detailed result of unigram feature for MNB classifier and ROC curve is shown in Table 4.4 and Figure 4.5.

Table 4.5: Accuracy of SMS datset using different features

Features	**# of features**	**MNB classifier**	**SVM classifier**
Unigram	2696	67%	48.7%
Bigram	21041	63%	45.8%
Unigram+Bigram	42008	60.2%	47.3%
POS	21014	65%	46.7%
Adjective	1503	44%	34.5%

Table 4.6: F1 score of MNB classifier for unigram feature

Class label	**Precision(%)**	**Recall(%)**	**F1 score(%)**
Positive	66.66%	70.22%	68.39%
Negative	72.53%	78.21%	75.26%
Neutral	62.5%	20.83%	31.24%

Proposed Approach

We proposed an approach to find out sentiment of more unstructured data like SMS. We followed the following rules:

- A small sentiment of positive and negative orienataion is first collected manually which forms the seed.

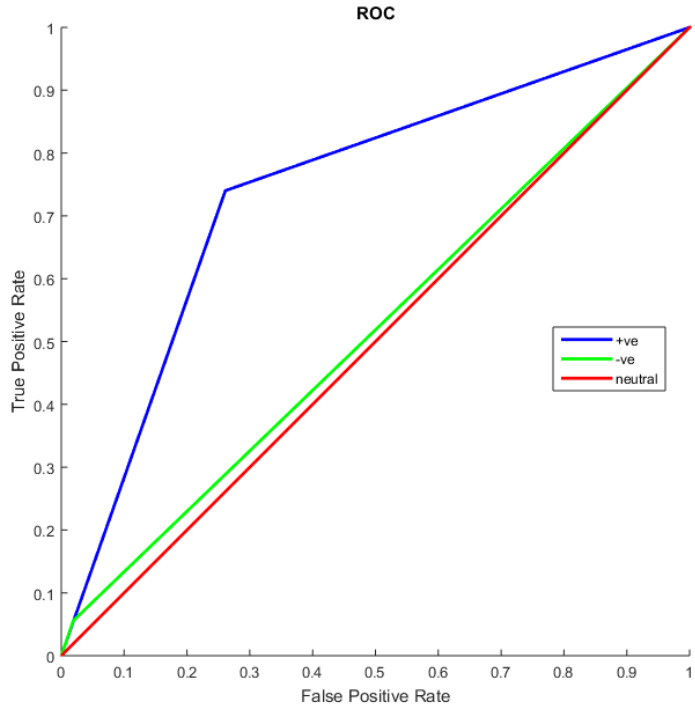

Figure 4.5: ROC curve of MNB classifier for SMS data set

- Repeat

  1. for each word in positive and negative list search in the WordNet for their synonyms and antonyms.
  2. add the newly found words to the seed list

  Until no more new words can be found

- Use these newly founded positive and negative list as features to train the classifier.
- Find the term level sentiment for all the word in a given SMS, then basically the message level sentiment is simply the majority word of all the word.

In our approach we have used MNB classifier for sentiment identification of SMS. We have found that by identifying term level sentiment of all the word in a message gives better result than MNB with unigram as feature. A snapshot of our program is shown in Figure 4.6.

```
>>> ================================ RESTART ================================
>>>
test sms = I do not like this car, sentiment = negative

test sms = This view is horrible, sentiment = negative

test sms = I love this car, sentiment = positive

test sms = It hurts me, sentiment = negative

test sms = amoure wins oscar, sentiment = positive
```

Figure 4.6: Result of short messages

4.5 Conclusion

We have applied machine learning algorithm to different dataset. We found that for sentence level classification accuracies using unigram presence and POS as feature turned out to be most effective as compared to other alternative features we employed. But these features are not giving better results for SMS. As SMS are more unstructured than tweets in our experiment we found that it is better to develop a system that can find the term level sentiment for all the word in a given SMS, then basically the message level sentiment is simply the majority word of all the word. Finally we have developed a model which can able to find the emotion of a tweets and we found that hashtagged words are good in finding emotions of a tweets.

Chapter 5

Conclusions

This thesis is a research work in the area of sentiment analysis that evaluates the application of lexical resources and machine learning algorithm for sentiment classification of unstructured data such as tweets and SMS. As subjective information are publicly available on Internet in digital format, the field of Sentiment Analysis requires an automated method to identify subjective content in a text and has applications in a variety of fields like on-line advertising and market research. Many times these sentiment information serves as a key attribute for taking productive decision and therefore used as an important criteria in the field of knowledge management.

During this research, we explored the challenges in Sentiment Analysis and different techniques used in this field to develop a reliable Sentiment Analysis system. Because of the complexity and nuances of social media data it is very difficult to identify the sentiment of such data. As part of this research we performed our experiment on tweets retrieved from twitter public domain to find the effective features for Sentiment Analysis. We have applied both lexicon based and Machine learning algorithm for SA. In our experiment we have used SentiWordNet lexicon with the objective to best use of this lexical resource to build a Sentiment Analysis system for tweets. We got an accuracy of 75.20% for our dataset using SentiWordNet lexicon and found that the result varies from domain to domain. Therefore it is better to generate a lexicon from the test corpus and use it for classification, this is because though existing lexicon contains large number of words with their sentiment score but they lack certain words that are found in a particular domain. In comparison to SentiWordNet

lexicon, our model where Google search engine is pulled to find the score for each term using point wise mutual information gives better result for our dataset and also able to handle one of the challenge in Sentiment Analysis i.e. sudden deviation from positive polarity to negative polarity.

We have applied machine learning algorithm to different dataset. We found that for sentence level classification accuracies using unigram presence and POS as feature turned out to be most effective as compared to other alternative features we employed. But these features are not giving better results for SMS. As SMS are more unstructured than tweets in our experiment we found that it is better to develop a system that can find the term level sentiment for all the word in a given SMS, then basically the message level sentiment is simply the majority word of all the word. Finally we have developed a model which can able to find the emotion of a tweets and we found that hashtagged words are good in finding emotions of a tweets.

Bibliography

[1] John A Horrigan. Online shopping. *Pew Internet & American Life Project Report*, 36, 2008.

[2] Bing Liu. Sentiment analysis and opinion mining. *Synthesis Lectures on Human Language Technologies*, 5(1):1–167, 2012.

[3] Bo Pang and Lillian Lee. Opinion mining and sentiment analysis. *Foundations and trends in information retrieval*, 2(1-2):1–135, 2008.

[4] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271. Association for Computational Linguistics, 2004.

[5] Bing Liu. Opinion mining and sentiment analysis. In *Web Data Mining*, pages 459–526. Springer, 2011.

[6] Bing Liu. Sentiment analysis and subjectivity. *Handbook of natural language processing*, 2:627–666, 2010.

[7] Andrés Montoyo, 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):675–679, 2012.

[8] Erik Cambria, Bjorn Schuller, Yunqing Xia, and Catherine Havasi. New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems*, 28(2):15–21, 2013.

[9] Khairullah Khan, Baharum Baharudin, Aurnagzeb Khan, and Ashraf Ullah. Mining opinion components from unstructured reviews: A review. *Journal of King Saud University-Computer and Information Sciences*, 26(3):258–275, 2014.

[10] Ronen Feldman, Moshe Fresko, Jacob Goldenberg, Oded Netzer, and Lyle Ungar. Extracting product comparisons from discussion boards. In *Data Mining, 2007. ICDM 2007. Seventh IEEE International Conference on*, pages 469–474. IEEE, 2007.

[11] Mohammad Sadegh, Roliana Ibrahim, and Zulaiha Ali Othman. Opinion mining and sentiment analysis: A survey. *International Journal of Computers & Technology*, 2(3):171–178, 2012.

[12] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up? sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002.

[13] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on association for computational linguistics*, pages 417–424. Association for Computational Linguistics, 2002.

[14] Andrea Esuli and Fabrizio Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In *Proceedings of LREC*, volume 6, pages 417–422. Citeseer, 2006.

[15] Alaa Hamouda and Mohamed Rohaim. Reviews classification using sentiwordnet lexicon. In *World Congress on Computer Science and Information Technology*, 2011.

[16] Bruno Ohana. Opinion mining with the sentwordnet lexical resource. 2009.

[17] Mitchell P Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

[18] Doug Cutting, Julian Kupiec, Jan Pedersen, and Penelope Sibun. A practical part-of-speech tagger. In *Proceedings of the third conference on Applied natural language processing*, pages 133–140. Association for Computational Linguistics, 1992.

[19] Shitanshu Verma and Pushpak Bhattacharyya. Incorporating semantic knowledge for sentiment analysis. *Proceedings of ICON*, 2009.

[20] Kamal Nigam, John Lafferty, and Andrew McCallum. Using maximum entropy for text classification. In *IJCAI-99 workshop on machine learning for information filtering*, volume 1, pages 61–67, 1999.

[21] Huifeng Tang, Songbo Tan, and Xueqi Cheng. A survey on sentiment detection of reviews. *Expert Systems with Applications*, 36(7):10760–10773, 2009.

[22] Daniel M Bikel and Jeffrey Sorensen. If we want your opinion. In *Semantic Computing, 2007. ICSC 2007. International Conference on*, pages 493–500. IEEE, 2007.

[23] Hiroshi Kanayama and Tetsuya Nasukawa. Fully automatic lexicon expansion for domain-oriented sentiment analysis. In *Proceedings of the 2006 Conference on Empirical Methods in Natural Language Processing*, pages 355–363. Association for Computational Linguistics, 2006.

[24] Matthias Hagen, Martin Potthast, Michel Büchner, and Benno Stein. Twitter sentiment detection via ensemble classification using averaged confidence scores. In *Advances in Information Retrieval*, pages 741–754. Springer, 2015.

[25] Saif M Mohammad, Svetlana Kiritchenko, and Xiaodan Zhu. Nrc-canada: Building the state-of-the-art in sentiment analysis of tweets. 2013.

[26] George Miller, Christiane Fellbaum, Randee Tengi, P Wakefield, H Langone, and BR Haskell. *WordNet*. MIT Press Cambridge, 1998.

[27] Gebrekirstos Gebremeskel. Sentiment analysis of twitter posts about news. *Sentiment Analysis. Feb*, 2011.

[28] I Hemalatha, GP Saradhi Varma, and A Govardhan. Preprocessing the informal text for efficient sentiment analysis. *International Journal*, 2012.

Dissemination

A. K. Dash, J. K. Rout, S. K. Jena. Harnessing Twitter for Automatic Sentiment Identification Using Machine Learning Techniques 3rd *International Conference on Advanced Computing, Networking, and Informatics (ICACNI-2015)*