Automated Sentiment Analysis using Machine Learning and Deep Learning Techniques

A Thesis

submitted in partial fulfillment of the requirements for the award of the degree of

Master of Engineering

in

Computer Science and Engineering

by

Yachika Gupta
(Roll No: 801532059)

Under the supervision of

Dr. Parteek Kumar
(Associate Professor)

Computer science and Engineering Department
THAPAR UNIVERSITY
PATIALA-147004, PUNJAB, India
June 2017

CERTIFICATE

I hereby certify that the work, which is being presented in the thesis, entitled **Automated Sentiment Analysis using Machine Learning and Deep Learning Techniques**, in partial fulfillment of the requirements for the award of the degree of **Master of Engineering** in **Computer Science and Engineering** submitted in Computer Science and Engineering Department of Thapar University, Patiala, is an authentic record of my own work carried out under the supervision of **Dr. Parteek Kumar** and refers other researchers work which are duly listed in the reference section.

The matter presented in the thesis has not been submitted for award of any other degree of this or any other University.

Yachika Gupta

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

Dr. Parteek Kumar
Associate Professor,
Computer Science and Engineering Department

Countersigned By

Dr. Maninder Singh
Head
CSED
Thapar University,
Patiala

Dr. S.S Bhatia
Dean(Academic Affairs)
Thapar University,
Patiala

Acknowledgement

The successful completion of any task would be incomplete without acknowledging the people who made it possible and whose constant guidance and encouragement secured the success.

First of all I wish to acknowledge the benevolence of omnipotent God who gave me strength and courage to overcome all obstacles and showed me the silver lining in the dark clouds.

With the profound sense of gratitude and heartiest regard, I express my sincere feelings of indebtedness to my guide **Dr. Parteek Kumar**, Associate Professor, Computer Science and Engineering Department, Thapar University for his positive attitude, excellent guidance, invaluable co-operation, generous attitude and above all his blessings. He has been a source of inspiration for me.

I am grateful to **Dr. Maninder Singh**, Head of Department, and **Dr. Ashutosh Mishra**, P.G. Coordinator, Computer Science and Engineering Department, Thapar University for the motivation and inspiration for the completion of this thesis.

Last but not the least I would like to express my heartfelt thanks to my parents and my friends who with their thought provoking views, veracity and whole hearted co-operation helped me in doing this thesis.

(Yachika Gupta)

Roll No. 801532059

Abstract

Due to exponential growth in Internet usage, social media has become common means of communication. People use to express opinions through social networks, review sites, blogs and forums. Daily a lot of data is generated on Internet. This data holds immense value as it can help in decision making which is possible through sentiment analysis. Sentiment analysis is the process of extracting useful information from user's opinions. With the advancement of social media usage, sentiment analysis has become an important area of research in today's life. But manual analysis of such a huge data is very difficult and time consuming process. So, automated sentiment analysis system is needed for analysis of this data.

A number of automated systems are available online that provide a number of features for text analysis, but no one is satisfying the requirements completely. Some features are available in one tool and some in others. In this research work a customized sentiment analysis system has been developed by merging two existing systems to facilitate the features of those tools on a common platform. Secondly, a real time Twitter sentiment analysis system has been developed using machine learning to analyze Twitter data in real time. In this research work, some deep learning approaches have also been proposed.

Table of Contents

List of Figures

List of Tables

Chapter 1

Introduction

1.1 Introduction to Sentiment Analysis

Sentiment analysis is an important research area these days. The analysis is done on the opinions expressed by people towards a particular entity. The entity may be a person, product, celebrity, policy or a firm. It is basically getting an idea of how people feel what they are talking about. The major source of these opinions is Internet. In a matter of few years Internet has changed the way people use to communicate with each other. With exponential growth in Internet usage, Facebook and Twitter has become common means of communication, with Facebook having nearly 2000 million users and Twitter 328 million users. Scraping opinions from web and doing sentiment analysis is not an easy task. So, a number of tools and techniques are available for sentiment analysis like machine learning, deep learning, lexicon based approaches and some hybrid techniques.

There are a number of ways in which the sentiments can be determined. The sentence may be classified into either of three classes as positive, negative or neutral, or it may be classified as highly positive, mildly positive, neutral, mildly negative and highly negative depending upon the tone of sentence. It can also be classified into 5 different emotions like happiness, fear, joy, disgust, anger. The sentence may be assigned a sentiment score depending upon the polarity of sentiments expressed in the sentence. Sometimes sentiments are expressed in the form of number of stars, like in movie reviews. One can transform that representation into sentence's actual sentiment class. For example, 1 and 2 stars can be considered negative, 3 stars as neutral and 4 and 5 stars a positive opinion. Sentences can also be classified as objective (containing facts) or subjective (containing emotions). Due to its many aspects sentiment analysis is often referred to with different names such as opinion mining, sentiment classification, sentiment extraction etc.

1.2 Levels of Sentiment Analysis

Depending upon the granularity of text under consideration, sentiment analysis can be performed at three different levels, *i.e.*, document level, sentence level and feature level. Document level sentiment analysis is the simplest form of sentiment analysis. The whole document is classified as either positive, negative or neutral. It is assumed that the document consists of opinions of a single object only. So, this is not the choice for document having opinions of multiple objects.

In sentence level sentiment analysis, sentence is the basic unit of analysis. The first task in sentence level sentiment analysis is subjectivity/objectivity classification. If the sentence is objective, it is filtered out. Only subjective sentences are considered for sentiment analysis.

In feature level sentiment analysis, analysis is done on the features of a product or an entity mentioned in the opinion and then judging the polarity of opinion towards that feature. Example sentence for feature level sentiment analysis is given in 1.1.

Battery life of mobile is long … (1.1)

Here, *battery life* of mobile is the feature phrase towards which sentiment is to be determined.

1.3 Applications of Sentiment Analysis

Internet and social media has made sentiment analysis a very popular area applicable to a number of domains like politics, business, marketing, box office *etc*. People use to express opinions towards different scenarios on Twitter and Facebook from where these reviews are gathered and analyzed for making predictions and amendments in policies. Sentiment analysis is very useful in improving customer satisfaction and company revenues. The main application of sentiment analysis is in decision making. Following are some of the application areas of sentiment analysis.

i. Election Results Prediction

Sentiment analysis helps political parties to get an idea of how much the public is satisfied with their policies and what are the chances of their winning in coming elections. People use to express opinions about politicians on Twitter and the

number grows during election days with 500 million tweets/day in normal days to 1 billion tweets/day during elections. These tweets are then gathered and analyzed for getting opinions of general public. Sentiment analysis has been proved successful a number of times in predicting election results. It correctly predicted the outcome of 2016 US elections and 2017 Punjab elections.

ii. Success of Social Movements

Sentiment analysis can be used for predicting the success of a newly introduced social movement or policy, for example, 'Odd-even car rule' in Delhi, 'Swachh Bharat Abhiyaan' etc. Public opinions are gathered to have an idea of how much they are happy with the new change and based on that it can be determined whether the new policy is going to be successful in future or not. This way money, time and efforts can be saved or some necessary steps can be taken to make that policy more successful.

iii. Improved Customer Experience

Almost all e-commerce websites like Amazon use sentiment analysis. They get feedback from customers about the products they have bought. These reviews later on help other customers in deciding which products to buy and which not to buy. These reviews are also used by companies to get an idea of how happy the customers are with their products. It helps companies focus on the overlooked issues in business, thus helping them to improve their business strategy. Sentiment analysis has been proved a lot useful to product companies in increasing their revenues and customer retention.

iv. Competitive Advantage

Analyzing the sentiment data of its competitors gives a company to perk up its business by hitting those targets in which the competing company is leading. Sentiment analysis is very helpful in predicting customer trends. Once the company gets acquainted with customer trends, business strategies can easily be

developed. With sentiment analysis it is always possible to adapt to the current market scenarios and better customer retention can be achieved.

v. Business Intelligence and Brand Brisking

Business Intelligence is remaining dynamic throughout. When a product is going to complete, it is exposed to public reviews so that changes can be done before finally exposing to public. This is called as concept testing a product before being launched.

Name and fame of a brand not only depends upon the products or services it provides. Campaigning strategies, social marketing and content marketing also count to its success. Sentiment analysis helps in developing more appealing and powerful marketing strategies.

vi. Stock Market Prediction

Sentiment analysis is widely used in stock market predictions. Rise and fall in stock prices of a company are highly correlated with the sentiments expressed about that company on social media. Thus, based on those opinions, one can decide whether in future the company is going to be in gain or loss and whether it is fruitful to invest money in stocks of that company.

vii. Predicting Box Office Collection

Sentiment analysis is useful in predicting the box office collection of a movie. Based on the reviews posted by people, one can predict the movie revenues in advance. Along with this application, people can also view the reviews to get an idea how good the movie is and is it worth to spend time and money on watching that movie.

List of applications is endless. A lot has been done using sentiment analysis and still a lot more can be done using it.

1.4 Challenges of Sentiment Analysis

Though sentiment analysis is widely accepted, still it has some problems resisting sentiment analysis to work to its 100%. As language processing is a complex task, some concepts remain hidden in text. Those concepts if understood properly, one will be able to get much more than ever from sentiment analysis. A lot can be done using sentiment analysis if the persisting issues are resolved. Following are some of the challenges faced by sentiment analysis.

1.4.1 Sarcasm Detection

Sarcasm is expressing emotions in a way in which words mean opposite to what one really wants to say especially to wit or insult someone or to be funny. Actually sarcasm is a form of speech. People usually express sarcasm through gestures or heavy tonal stress. Through speech and face expressions, one can easily detect sarcasm but from written text it is very difficult to judge the intentions of the writer. So, one has to rely on other factors to handle sarcasm. Sarcastic sentences have implicit sentiments hidden in them. So, it is hard even for humans to detect. For example,

You call it a work of art? … (1.2)

Great, how messy is the room … (1.3)

Sentence 1.2 is sarcastic and there is no sentiment word present in it. This is the problem of implicit sentiment. Sarcasm is a form of irony, the one used in text and speech for same purpose as sarcasm, but still the two are different.

Sentence 1.3, in spite of presence of strong positive word *Great* is negative. The sentence is ironical. So, it is very difficult for a machine to judge whether the sentence is sarcastic or not.

1.4.2 Anaphora Resolution

Anaphora problem is often overlooked in sentiment analysis. While doing sentiment analysis, only the opinion words are considered while what those opinions are about is often ignored. So, along with opinion words extraction from text, targets of opinion

words should also be considered. Anaphora is basically back reference to an object that is often mentioned in previous sentences, for example, pronouns.

I don't think this move will cross even 100 crore. It is totally awful. … (1.4)

In example 1.4, second sentence doesn't mention the entity which is *awful*. To resolve the anaphora *It*, one has to consider the first sentence also. It is therefore necessary to analyze multiple sentences to extract the targets.

1.4.3 Entity Identification

There may be multiple entities present in a sentence with different opinions. It is very important to correctly identify the entities and their corresponding opinions to avoid any type of misinterpretation. For example,

Apple is better than Samsung. … (1.5)

Staff was good but the food was terrible. … (1.6)

In sentence 1.5, *Apple* is opinionated as positive and *Samsung* as negative. Similarly in sentence 1.6, *Staff* is mentioned in positive sense and *food* in negative sense. It is necessary to make such distinctions to get correct results.

1.4.4 Domain Dependency

The sentiment of a word varies from domain to domain. In one domain it may be positive and in other it may be negative. So, while doing sentiment analysis domain knowledge is one of the essentials. For example,

Story of movie was unpredictable. … (1.7)

Her mood is unpredictable. … (1.8)

In example sentence 1.7, the word *unpredictable* represents a positive opinion about the movie, but a negative opinion in sentence 1.8 regarding the mood of a person. So, this issue must be focused to avoid any misclassification.

1.4.5 Thwarted Expectations

Thwarted expectations basically refer to the phenomena where writer firstly builds up certain expectations for the topic and then disproves them using opposite opinion. The main sentiment polarity of the review lies in the last few lines that sums-up the review. For example,

I love slim design. The screen is impressive. Camera takes good pictures. But still this phone disappoints me. I don't recommend it. … (1.9)

The review in example 1.9 is about a phone. In spite of having a good count of positive words, the overall opinion is still negative. There are chances of it classified as positive if frequency of opinion words is considered. So, an appropriate strategy must be designed to handle such cases.

1.4.6 Negation

Negation handling is also a challenging task in sentiment analysis. Negation words like not, never *etc.* change the actual sentiment of the word. If unigram word features are used, this issue remains unhandled, but it can be solved using *n*-grams. If only unigrams are considered, then the sentiment words present in the sentence will determine the polarity of sentence, may it be opposite of the actual sentiment. For example,

The results are not accurate. … (1.10)

Sentence 1.10 is actually negative, but it may be wrongly classified as positive. Negation may have multiple forms.

Her acting is not good, but she sings well. … (1.11)

In sentence 1.11, the scope of negation is limited until *but* is encountered. There are some other problems also.

Not only her acting is good, but she also sings well. … (1.12)

In sentence 1.12, presence of word *only* doesn't let *not* to reverse the polarity of sentence. So, while handling negations one should consider other impact words also.

1.4.7 Subjectivity Detection

Subjectivity detection is the problem of distinguishing subjective and objective sentences. Objective sentences consist of factual description carrying no sentiments at all, while subjective sentences actually consist of sentiment data that needs to be considered. So, the system should be able to distinguish between the two types, so that objective sentences are filtered out and only subjective sentences are retained for sentiment analysis. It will make the system more accurate and efficient. For example,

The concert is going on. … (1.13)

I love attending concerts. … (1.14)

In above example, the sentence 1.13 is objective and sentence 1.14 is subjective with positive opinion.

1.4.8 World Knowledge

Sometimes, without having world knowledge it is not possible to determine the sentiments of the text. So, world knowledge is the crucial part to be considered. For example,

ISRO launched 104 Satellites in one go. … (1.15)

She has completed her Post Doctorate degree. … (1.16)

In above sentences, it is must to have world knowledge. In sentence 1.15, system should have the knowledge of *ISRO* and *Satellites*. In sentence 1.16, the system should have the knowledge of *Post Doctorate*. If system has this knowledge, only then it can analyze the impact of those words.

1.4.9 Slang and Short Forms

Slang words are popular emotion words mainly used on social media. People use '*LOL*', '*PJ*' type of slang words to express feelings. Until now this issue remained unhandled, but now various lexicons have been designed especially for slang words addressing this issue to some extent.

There are some other problems also like people use short words *i.e.* ignoring some characters while typing a word. For example, '*gud*' for '*good*', '*cum*' for '*come*', '*ppl*' for '*people*' *etc*. Using short words change the meaning of actual words and these short words are not found in any dictionary. Though, special dictionaries have been created for these types of words, but short words are created so dynamically that it is impossible to cover all the variations, because different people have different ways of expressing a word.

1.4.10 Code-Switching

This problem is also related to social media. Mostly people use multiple languages while expressing opinions. Code-switching is the concept of expressing words of one language using the alphabets of other language. For example,

Main aur wait nahin kar sakta. … (1.17)

In example sentence 1.17, some Hindi words are expressed using alphabets of English which is hard for a machine to identify. These are called as foreign words. Words like *Main, aur, nahin, kar, sakta* come under this category. There are several issues in the above sentence. As the text consists of both English and Hindi vocabulary, the system first needs to identify whether the sentence consists of only English words or some foreign words are also present, then to detect the actual language of foreign words (in this case Hindi) and then to transform the foreign words to their actual language which is a very complex task.

1.5 Sentiment Analysis Process

Sentiment analysis is a complex process that involves different phases to analyze sentiment data. Figure 1.1 displays displays different phases involved in sentiment process.

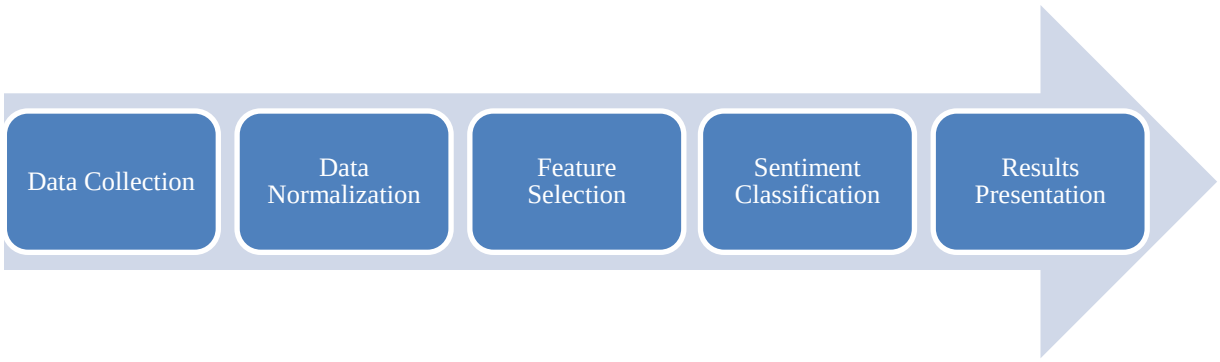

Figure. 1.1 Sentiment Analysis Process

1.5.1 Data Collection

First step in sentiment analysis process is data collection. Internet is a huge repository of data. With exponential increase in Internet usage, social media sites like Facebook, Twitter, Blogs, User forums *etc.* have become common means of communication. People generally use these social media sites for expressing opinions regarding a particular product, firm, movie or any other topic of interest. There exist a number of techniques for collecting this data. Data from twitter can be collected using Twitter API. For real time data collection Twitter Streaming API is used. Web scraping can be done for extracting reviews from blogs or review sites, for example, web scraping of movie reviews.

1.5.2 Data Normalization

It is one of the necessary phases of sentiment analysis process. Data that is collected cannot be used for direct sentiment analysis. It consists of a lot of junk content like @, #, URLs, RTs and Unicode characters, that don't play any role in revealing sentiments. So, this data needs to be cleaned before analysis. This data cleaning also known as data normalization or data preprocessing is done in a number of ways like stripping, lowercasing, removal of unwanted characters, stop words removal *etc.*

1.5.3 Feature Selection

Feature selection is an important process of extracting useful information from data. Feature selection can be done in a number of ways. It is up to the programmer to select

the most relevant features from the available data to get the best information. Feature selection can make the sentiment classification process more accurate and efficient. For example, along with considering individual word entries, considering their part of speech also can give better results. Sometimes system generates a lot of features, but all are not that important. They unnecessarily increase the complexity of model. So, it is better to weigh them by relevance and selecting only the top most features for analysis.

1.5.4 Sentiment Classification

Sentiment classification is the process of classifying a text into either of the pre-specified classes. Classes may be positive, negative or neutral or more specific like fear, anger, joy, disgust, happiness depending upon the requirement. There are a number of techniques available for sentiment classification. For example, Machine learning trains the model on known examples, lexicon approach is a comparison based approach which doesn't train the model instead classification is done by comparing individual words or phrases with a pre-annotated lexicon. Sometimes no annotated data is available and the model itself has to find similarities and dissimilarities between the data, this is unsupervised technique. A combination of the discussed techniques can sometimes give better results and is known by the name hybrid.

1.5.5 Results Presentation

After sentiment analysis is done, the next step is presentation of results. There can be a number of ways of presenting sentiment results to users. For example, a simple textual representation marking polarity class against each sentence or graphical representations like bar graphs, line graphs, pie charts *etc.* can be used. A time analysis can also be done if the goal is trend analysis of data, *i.e.*, displaying the variation of sentiment data with time. This approach can be used for real time sentiment analysis also to see how users' opinions change with time. There are a number of libraries available both in python and R for presentation of output.

1.6 Thesis Outline

The thesis is divided into 6 chapters. Chapter 1 includes introduction to sentiment analysis. It also covers various levels of sentiment analysis, its application areas and the challenges faced by sentiment analysis. This chapter also discusses the process of doing sentiment analysis. Chapter 2 discusses various tools and techniques for sentiment analysis. Techniques include lexicon based approaches, machine learning, deep learning and hybrid approaches. A brief overview of related work is also provided in that chapter. Chapter 3 presents the problem statement, objectives and methodology for developing the sentiment analysis system. In chapter 4 Customized Sentiment Analysis System has been proposed that is an ensemble of two existing sentiment analysis systems. In chapter 5 different machine learning and deep learning models have been trained and tested on varied sets and best model is selected to work for real time sentiment analysis system. In chapter 6, real time sentiment analysis system is discussed which is tested for results prediction of 2017 Punjab elections. Next the work done in this research work is concluded.

Chapter Summary

In this chapter, an introduction to sentiment analysis and its importance is done. It can be realized that how important sentiment analysis is and how much applicable it is. Along with its applications, some challenges of sentiment analysis have also been discussed. Then all the phases that are required for sentiment analysis are discussed. In the next chapter, literature survey has been done depicting the work done in sentiment analysis.

Chapter 2

Literature Review

In this chapter, various tools and techniques for sentiment analysis have been discussed. A number of techniques are available like machine learning, deep learning and lexicon-based approaches. Some online tools are also available for sentiment analysis. Feature selection is also discussed that plays an eminent role in making an algorithm more efficient and accurate. Along with it, a brief overview of the work done so far using these techniques is also given.

2.1 Techniques for Sentiment Analysis

There are a number of techniques that have been used and are evolving for sentiment analysis. These include machine learning, deep learning, lexicon based approaches, hybrid and rule based approaches. These techniques are discussed below.

2.1.1 Lexicon Based Approaches

Lexicon based approach is a comparison based approach. A lexicon is a pre annotated set of features. To determine the sentiment value of the text under consideration, features of text are matched with the features that are present in the lexicon. Corpus based and dictionary based are the two lexicon based approaches. These are discussed below.

- **Dictionary based approach**

  In dictionary based approach, a set of opinion words with known orientations is collected manually. This dictionary is then grown by searching for synonyms and antonyms of already collected words in other popular corpora like WordNet or thesaurus. This process is repeated until no new words are found. This approach has one major disadvantage i.e. inability to find words with domain specific orientations.

- **Corpus based approach**

Corpus based approach is relatively more accurate and helps solve the problem of finding opinion words with domain specific polarity. Its methodology depends upon the syntactic patterns that exist within the list of already collected opinion words to find other opinion words in a large corpus.

Lexicon based approach has not been very effective. Due to accuracy concern and domain dependence, it is not so popular. Looking at its limitations, machine learning approach is followed.

2.1.2 Machine Learning

Machine learning is a subfield of Artificial Intelligence that makes computers learn without being explicitly programmed. Machine learning is basically constructing the algorithms that can learn from data and make predictions on the related data [2]. Machine learning models try to learn from trends in data and apply that learning while making predictions. Machine learning is categorized into two classes namely supervised and unsupervised machine learning. Supervised models learn from labeled data whereas no labels are provided for unsupervised learning, models themselves try to find the patterns to reach the goal. The approach used in this paper is supervised learning.

Machine learning is a complex process involving a number of phases. These are depicted in figure 2.1.

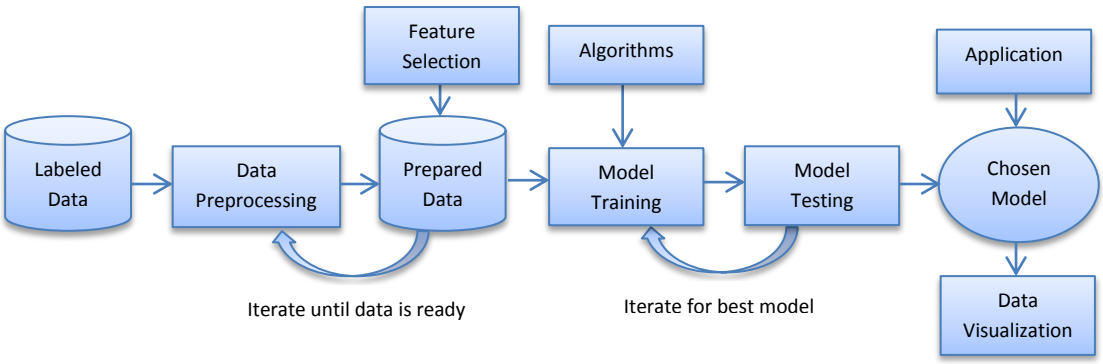

Figure. 2.1 Machine Learning Process

Firstly, data is gathered from various sources and it may be in labeled form or may be labeled manually after collection. This data is then preprocessed to have homogeneity

and the process is repeated until data is preprocessed completely. Then feature selection phase comes during which pre-selected features are used for training the models and best features are selected for finally building the models. Different models are trained and tested on the data and this process is repeated until a model with best performance is obtained. This model is then selected to be applied on the problem in hand and the results are presented in any form suitable to the problem, for example, line graphs, bar charts or pie charts.

Feature Selection

Feature selection is an important phase in sentiment analysis. It is the process of getting information from the available data, for example, getting part of speech tags along with unigrams. A lot of information can be gathered from text that can be used to impart learning to the classifiers. Sometimes the feature space is very large and is not very informative in whole. In that case a subset of most informative features is selected for training the model. Following are some of the feature choices used for sentiment analysis [1].

i. **Term Presence or Term Frequency**

   Term presence and term frequency are two different approaches for making features. Features in term presence consider the presence or absence of a particular term, *i.e.*, sentiment orientation of a text varies depending upon whether a particular term is present or not. Term frequency is based on the count of a particular term in text. Both are applicable in different text mining problems. While term presence is more appropriate for sentiment analysis, term frequency is mostly used for topic categorization. Terms can be unigrams, bigrams or trigrams. It has been claimed that unigrams perform better for sentiment analysis of movie reviews.

ii. ***N*-grams**

   N-grams refer to unigrams, bigrams or trigrams. The performance of a classifier varies with whether the feature set consists of unigram terms, bigrams or trigrams.

Generally unigrams are observed to be most accurate, but bigrams are sometimes used to handle negation problem in text and trigrams are used to deal with intensifiers like very, extremely *etc*. For movie reviews dataset, better results are seen with unigrams and for product reviews dataset bigrams performed better.

iii. Part Of Speech (POS) Tagging

POS tagging also called as grammatical tagging or word category disambiguation, is the process of tagging each word in text with its associated part of speech. POS tag of a word varies with the context it is used in or the type of words surrounding it, correspondingly changing the sentiment orientation of text.

For example,

I love this movie … (1.18)

This is a love story … (1.19)

In sentence 1.18, POS tag of the word *love* is verb and the sentence is in positive sense. In sentence 1.19, POS tag of the same word is adjective and the sentence is neutral. Including Part Of Speech in feature space imparts more learning to the classifier as the classifier learns to differentiate between different contexts of a word.

iv. Lemmatization

Lemmatization is the process of grouping different inflected forms of a word together so that they can be interpreted as a single term. Lemmatization is the process of determining the base form or lemma of a given word and converting it to its base form. For example, the words *good*, *better* and *best* all have *good* as their lemma. Lemmatization takes part of speech into consideration as the lemma of a word varies with the intended part of speech of word.

v. Opinion words and phrases

Sometimes sentiment classification of text is relied on the opinion words and phrases present in text. Opinion words are the words that carry sentiment, for example, *hate*, *nice*, *lovely etc.* These are adjectives and adverbs actually. Similarly opinion phrases include phrases like *Nobody bothers*. But this approach may lead to misinterpretations when the opinion words are negated, resulting in reverse polarity of text.

vi. Negation Handling

Negation is important information to be considered while doing sentiment analysis, as it reverses the polarity of the opinion word. Handling negations has reported an accuracy increment of 3% on electronic product reviews. Consider, for example, the sentence 2.20,

I am not happy. … (2.20)

If only the opinion word *happy* is considered, the sentence will be classified as *positive*, in spite of being negative. Thus, negation handling can give better results.

Supervised Machine Learning

In supervised machine learning, the model is trained on the labeled data. The model learns from the examples that can be manually annotated or can be gathered from some other source like Internet. Three mostly used classifiers for supervised machine learning are Naïve Bayes Classifier (NBC), Support Vector Machines (SVM) and Maximum Entropy (ME). Among these classifiers, Support Vector Machines are most accurate for sentiment analysis. But as SVMs require a large dataset to build a high quality model, Naïve Bayes is more suitable when large amount of training data is not available. The description of three classifiers is as follows.

i. Naïve Bayes Classifier

Naïve Bayes classifier belongs to the family of probabilistic classifiers and is based on Bayes theorem [3]. It takes the probability distribution of words in the

training dataset and assumes them to be mutually independent. Given a feature vector $(x_1,...,x_n)$ and a class variable *y*, Naïve Bayes assigns the class to the feature vector according to Bayes formula which is given in 2.1.

$$P(y \mid x_1,...,x_n) = \frac{P(y)P(x_1,...x_n \mid y)}{P(x_1,...,x_n)} \quad \text{… (eq. 2.1)}$$

In this formula, $P(y \mid x_1,...,x_n)$ is posterior probability.

P(*y*) is prior class probability and $P(x_1,...,x_n)$ is the prior probability of feature set. These prior probabilities are obtained from training dataset. $P(x_1,...x_n \mid y)$ is the conditional probability of feature vector $(x_1,...,x_n)$ given the class *y*. The formula can be generalized as given the feature vector, Naïve Bayes finds the probability of each class to be assigned to this feature vector and assigns the class with maximum probability [4]. Naïve Bayes assumes mutual independence among the features. This can be seen in equation 2.2

$$P(x_1,...,x_n \mid y) = \prod_i P(x_i \mid y) \quad \text{… (eq. 2.2)}$$

ii. Multinomial Naïve Bayes

Multinomial Naïve Bayes is a specific version of Naive Bayes. Whereas a simple Naïve Bayes classifier models the document as the presence or absence of words, Multinomial Naïve Bayes takes into account the words counts. Given a class *c*, Multinomial Naïve Bayes estimates the conditional probability of a particular word as the relative frequency of the word in that class. This formula is shown in equation 2.3.

$$P(t \mid c) = \frac{T_{ct}}{\sum_{t' \in V} T_{ct'}} \quad \text{… (eq. 2.3)}$$

Here, *t* is the term/word and *c* is the class under consideration. This formula calculates the probability of a word to be classified into a class as the count of that word in the class w.r.t count of all the words in that class.

iii. Bernoulli Naïve Bayes

Bernoulli Naïve Bayes is designed for Boolean features, *i.e.*, it does not take into account the word counts but it takes the presence or absence of words. It takes 1 if the word is present in the document under examination and 0 if not present. So, those words are also taken care of that were present in training data but are absent in the predictions dataset. This is not the case with the previous classifiers. Absent terms remain completely ignored in those classifiers.

iv. Maximum Entropy Classifier

Maximum Entropy classifier is similar to Naïve Bayes classifier except that it doesn't make any assumption about the independence of features [5]. The principle idea behind Maximum Entropy is that it tries to maximize the Entropy and at the same time satisfying the constraints specified. The idea behind Maximum Entropy is to have a model that is as unbiased as possible and thus the probability distribution to be as uniform as possible. Maximum Entropy is when all the events are equally likely to occur and have maximum uncertainty.

The formula for Entropy is given in equation 2.4. The goal is to maximize $H(p)$.

$$H(p) = -\sum p(a,b) \log p(a,b) \quad \ldots \text{(eq. 2.4)}$$

v. Support Vector Machines

Support Vector Machines work on the concept of a decision plane or a hyper plane [6]. It tries to find a hyper plane which separates the data belonging to two classes as far apart as possible.

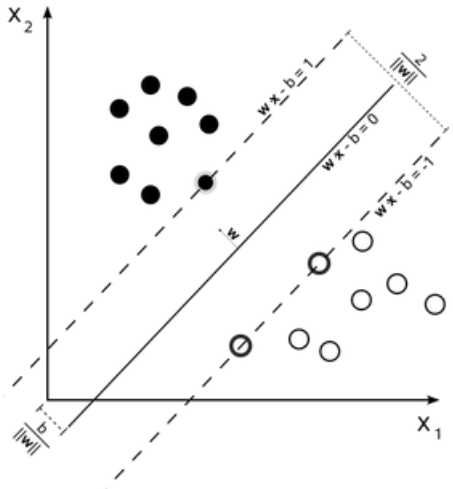

Figure. 2.2 Support Vector Machine

The equation for hyper plane is given in 2.5.

$$\left(\vec{w}\cdot\vec{x}\right)+b=\sum_i y_i\alpha_i(\vec{x_i}\cdot\vec{x})+b \qquad \ldots \text{(eq. 2.5)}$$

Here, $\vec{x_i}=(x_{i1},...,x_{in})$ is input feature vector, y_i is output class, $\vec{w_i}=(w_{i1},...,w_{in})$ is the weight vector defining the hyper plane and α_i is Lagrangian multiplier. Once the hyper plane is constructed, the class of any feature vector can be determined.

Existing work using Machine Learning

A lot of work has been done in sentiment analysis using machine learning and machine learning is still an area of interest. Pang *et al.* (2002) did sentiment analysis on movie reviews dataset using three classifiers Naïve Bayes, Support Vector Machines and Maximum Entropy and concluded that machine learning outperforms human-produced baselines [7].

Hiroshi *et al.* (2004) proposed a method to extract sentiment units from sentences. They used transfer-based machine translation engine for this purpose. Data was collected from bulletin boards on the WWW discussing digital cameras. Their experimental results

showed that the precision of sentiment polarity was significantly higher than for the conventional methods [8].

Boiy *et al.* (2009) performed experiments with three machine learning classifiers Support Vector Machine, Multinomial Naïve Bayes and Maximum Entropy using different features like unigrams, stems, negation *etc.* The data collected from blogs, reviews and forums was found in three languages English, Dutch and French. Their system attained an accuracy of 83% for English texts, 70% and 68% for Dutch and French respectively using unigrams as features. They have used Cascade Architecture for their research by splitting the classification task to different models and then cascading those models [9].

Schrauwen, Sarah (2010) annotated a Dutch Netlog corpus collected from social networking site Netlog. The corpus was annotated on three levels, *i.e.*, valence (positive, negative, both, neutral and n/a), performance (standard and dialect) and chat (chat and non-chat). Three machine learning classifiers Naïve Bayes, Maximum Entropy and Decision Trees were used for the purpose with features as most informative words of the corpus. They attained an accuracy of 65.1% for valence classification, 77.6% for performance classification and 84.2% for chat classification [10].

Omaima *et al.* (2014) [11] used Naïve Bayes classifier for doing location based sentiment analysis of Twitter data to identify the trends towards Indian General Elections of 2014. They analyzed the data for comparison between two political parties Aam Aadmi Party (AAP) and Bhartiya Janta Party (BJP) and classified the tweets as either positive or negative. For building feature set they removed stop words and duplicate words from tweets and attained an accuracy of 70%.

Amolik *et al.* (2015) [12] used machine learning techniques to analyze the twitter content related to Bollywood and Hollywood movies. They used two algorithms Naïve Bayes and Support Vector Machines (SVM) for classifying the tweets. They considered removing stop words and limiting character repetitions more than twice to twice only for building feature set and achieved an accuracy of 65% for Naïve Bayes and 75% for SVM.

Suchita and Sachin (2015) [13] compared two machine learning algorithms Naïve Bayes and Support Vector Machines (SVM) for classification on Internet Movies Database

(IMDB), a movie reviews dataset and concluded that Naïve Bayes give better results than SVM. They converted the whole text into lower case and omitted control characters, numbers and punctuations to get better matches. They got an accuracy of 65.57% for Naïve Bayes and 45.71% for SVM.

Sahayak *et al.* (2015) [14] did sentiment analysis of Twitter data using three machine learning classifiers Naïve Bayes, Maximum Entropy and Support Vector Machines. They considered n-grams, Part Of Speech (POS) tags and removal of stop words as their features and used a tree kernel to avoid monotonous feature engineering.

Zhang *et al.* (2016) [15] compared different machine learning approaches for sentiment analysis of Chinese text. They considered two models Support Vector Machines (SVM) and Extreme Learning Machine (ELM) with kernals for analysis. Their feature set includes verbs, adverbs and adjectives and TF-IDF has been used to calculate the weight of words. They achieved an accuracy of 88.54% for SVM and 88.74% for ELM with kernals and concluded that the later model is better in terms of accuracy as well as execution time.

Kiran *et al.* (2017) [16] did sentiment analysis of latest Twitter reviews of mobile and PC games using machine learning algorithms. They used Support Vector Machines (SVM), Naive Bayes (NB) and Maximum Entropy (ME) and achieved the respective accuracies of 83.2%, 64.2% and 90%.

2.1.3 Deep Learning

Deep learning also known as deep structured learning is a branch of machine learning inspired from human brain. Like human brain, it consists of numerous layers of artificial neurons termed as nodes in artificial intelligence. A neural network is a cascade of neuron layers with output of one layer fed as input to the next successive layer. Each layer passes on the modified version of data to the next layer to promote more informative features further. Deep learning has emerged as a powerful tool for pattern recognition and language processing in recent years. Because of its ability to automatic feature engineering and appreciable accuracy it is getting widespread popularity these days. Neural networks cannot process direct words, but they work on *word embeddings*

or more specifically feature vectors representing those words. One of the promises of deep learning is replacing handcrafted features with efficient algorithms for feature learning. These are capable of capturing very high level features from input data. One advantage drawn from this automatic feature engineering is domain independence. As neural networks learn features from the task in hand they can adapt to any domain. Given sufficient amount of training data and training time, deep nets can perform better than traditional machine learning approaches. Deep learning has found applications in a number of areas like sentiment analysis, computer vision, automatic speech recognition *etc.* Figure 2.3 shows the basic architecture of a fully connected neural network model.

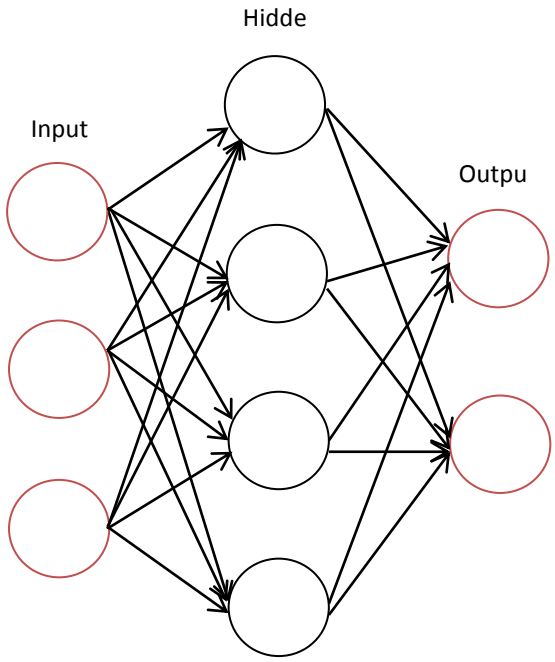

Figure. 2.3 Architecture of basic Neural Network

The basic neural network consists of three layers, *i.e.*, input layer, hidden layer and output layer. Input layer accepts input from user in the form of vectors and passes on the modified version to hidden layer. Hidden layer then further processes the income data and passes it on to the output layer. Output layer then finally generates the results to user.

i. Multi-Layer Perceptron Model

A Multi-Layer Perceptron model is a feed forward supervised Artificial Network (ANN) model that learns a function $f(\cdot): R^m \rightarrow R^o$ by training on a dataset where m is the number of input dimensions and o is the number of output dimensions. For a classification problem, number of nodes in input layer depends upon the length of input vector and number of nodes in output layer depends upon the number of pre-defined classes. There can be any number of hidden layers in between input and output layer. Input layer takes $x_1, x_2, ..., x_m$ as input vector. Each neuron in the hidden layer transforms the values from the previous layer with a weighted linear summation $w_1x_1 + w_2x_2 + ... + w_mx_m$ followed by a non-linear activation function $g(\cdot): R \rightarrow R$. Output layer then transforms values from the last hidden layer into output. Figure 2.4 displays the principle architecture of Multi-Layer Perceptron model.

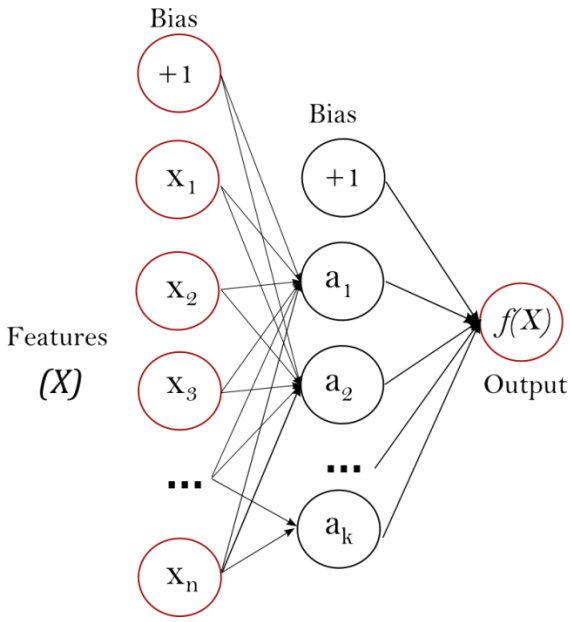

Figure. 2.4 Architecture of Multilayer Perceptron Model

The architecture shown above is having one hidden layer but it may have any number of hidden layers.

ii. Convolutional Neural Network

Convolutional Neural Networks (CNNs) are very much similar to the ordinary neural networks. Like ordinary neural networks, neurons in CNNs take some input, process it and propagate it further. The difference is that convolutional neural networks explicitly assume input as images. This is the reason they are explicitly used for analyzing image data. Regular neural networks don't scale well to full images. For small dimensions these are manageable, but as the dimensions grow, more neurons and parameters are required leading to the problem of over fitting.

As CNN is specifically designed for image data it constrains the architecture in a more sensible manner. Unlike regular neural networks neurons in each layer of CNN are arranged along three dimensions, *i.e.*, height, width and depth. A CNN has three types of layers namely convolutional layer, pooling layer and fully-connected layer. Convolutional layer is the main building block of a CNN as most of the computations are done at this layer. Figure 2.5 displays the architecture of Convolutional Neural Network.

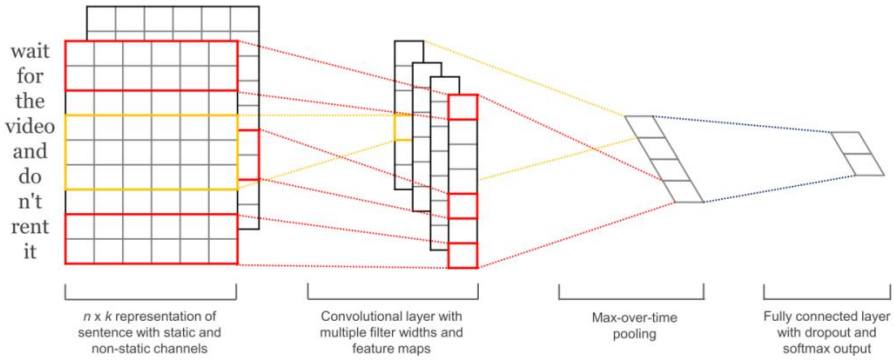

Figure. 2.5 Architecture of Convolutional Neural Network

The CNN architecture shown above consists of 4 layers. First is input layer that represents the sentences over n x k dimension, second is convolutional layer, then max pooling layer and finally fully connected layer producing output results.

iii. Recurrent Neural Network

In a regular neural network all inputs and outputs in a layer are considered independent. This is the reason that from the present state future events cannot be predicted and this is the major shortcoming of an ordinary neural network. This problem is resolved by Recurrent Neural Networks (RNN). RNNs make use of loops making information to persist. Thinking another way, RNNs have memory which stores the information calculated so far and using that information for future predictions. RNNs have applications in a number of areas which an ordinary neural network cannot solve, for example, based on the current events in a movie RNN can determine the next event. Similarly, given a sequence of words, the next word in sequence can be determined using RNN. Other applications include handwriting recognition and speech recognition. The most common Recurrent Neural Network is Long Short Term Memory or LSTM in short. The principle architecture of RNN is shown in figure 2.6.

In this figure, h_t is input and x_t is corresponding output of neural network. As RNN is unfolded it becomes similar to regular neural network.

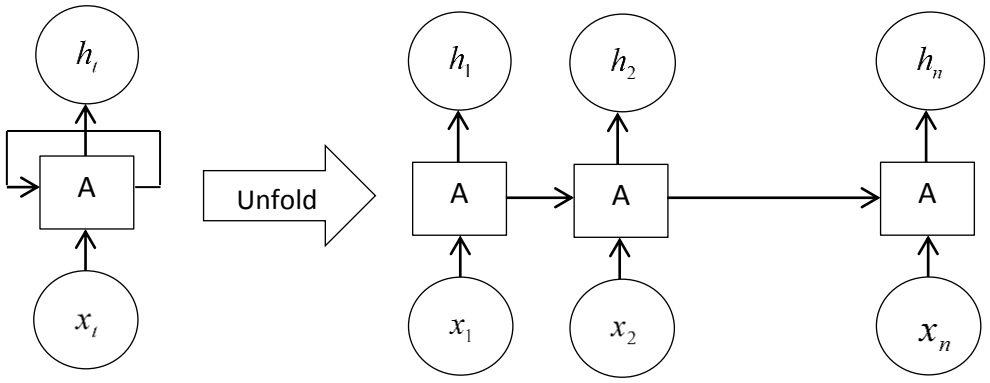

Figure. 2.6 Recurrent Neural Network

Existing Work using Deep Learning

In recent years, deep learning has got more popularity because of its accuracy and automatic feature engineering. Mehr *et al.* (2014) [17] compared different deep learning algorithms like Recurrent Neural Networks, Recursive Neural Networks and Convolutional Neural Networks with Naïve Bayes on a movie reviews dataset. They attained an accuracy of 40.3% for Recurrent Neural Networks, 42.2% for Recursive

Neural Networks, 40.5% for Convolutional Neural Networks, 46.4% for Convolutional Neural Networks+word2vec and 40.3% for Naïve Bayes on testing dataset.

Stojanovski *et al.* (2015) [18] analyzed the performance of Convolutional Neural Networks (CNN) for sentiment analysis of twitter data and classify the text into three sentiment classes, *i.e.*, positive, negative and neutral. They modified two existing approaches of deep learning by adding two fully connected layers to the existing neural networks and achieved an F1-score of 64.85% on Twitter2015 dataset which was comparable to the existing traditional approaches to sentiment analysis.

Hassan *et al.* (2017) [19] proposed a ConvLstm neural network architecture that employs two architectures (Convolutional Neural Networks) CNN and (Long Short term Memory) LSTM for sentiment analysis of short texts. They subsituted LSTM in the pooling layer of CNN for reducing loss of detailed information and to capture long term dependencies in sentences. The models were built on pre-trained word vectors and IMDB movie reviews dataset and Stanford Sentiment Treebank (SSTb) dataset were used for evaluating these models. They achieved an accuracy of 88.3% for binary classification and 47.5% for five-class classification problem.

2.2 Online Tools for Sentiment Analysis

There are a number of sentiment analysis tools available online. A detailed description of these tools is given as follows.

2.2.1 AlchemyAPI

Alchemy API was founded by Elliot Turner in 2005. It is a collection of various text and image analytics APIs like AlchemyLanguage, AlchemyDataNews and AlchemyVision [20]. AlchemyLanguage is a text analytics API used for sentiment analysis of text and helps understand various other concepts like keywords, entities, emotions *etc.* [21]. AlchemyAPI can take text or URL as input. It is a very powerful tool in that before processing it removes unwanted content like ads and headers from a webpage leaving only meaningful content for analysis. Maximum size that can be fed to the tool is 600KB and it supports a number of languages like English, German, French, Italian, Spanish,

Russian, Swedish and Portuguese. AlchemyLanguage uses REST API for making API calls for data analysis and responses are returned in JSON format. AlchemyAPI also provides client libraries for node.js, python, java, swift, unity and android. The tool is available online at www.alchemyapi.com. Figure 2.7 and figure 2.8 show the working of AlchemyLanguage.

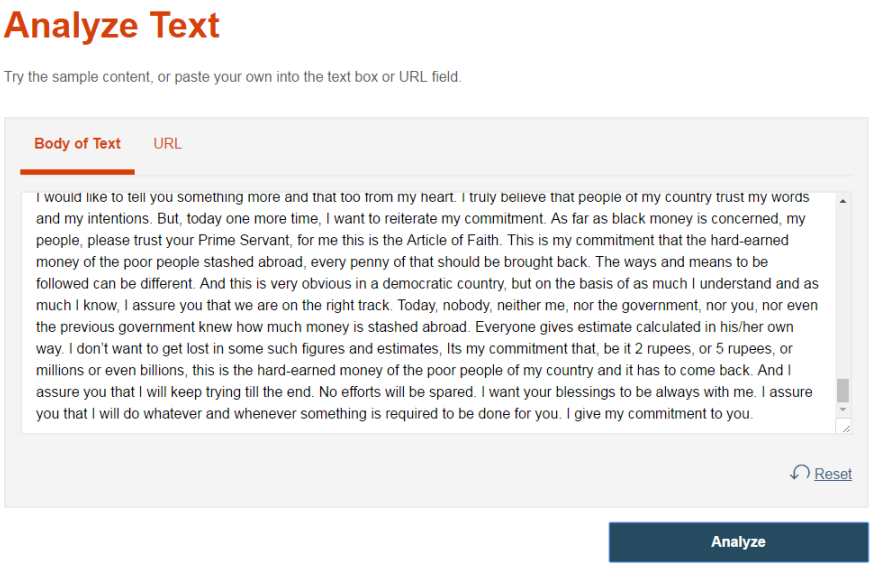

Figure. 2.7 Text Input to AlchemyLanguage

Figure 2.5 shows the text input to AlchemyLanguage. Along with it, URL option is also given if user wants to submit a URL. Figure 2.8 depicts results for the corresponding input text. The feature *Entity Extraction* is used here. The system has identified entities from text and has also found their relevance, sentiment class and the type of each entity.

Following are the features provided by AlchemyLanguage.

- **Sentiment Analysis:** It classifies the text as either positive, negative or neutral within the range (-1,1). The tool can do sentiment analysis at various levels like document level, user specified targets, entity level, and keyword level.
- **Emotion Analysis:** Sentiments can be classified into more fine grained classes like joy, sadness, fear, disgust, anger.
- **Keywords Detection:** Important keywords in the text are identified, associated with sentiment and are ranked by relevance.

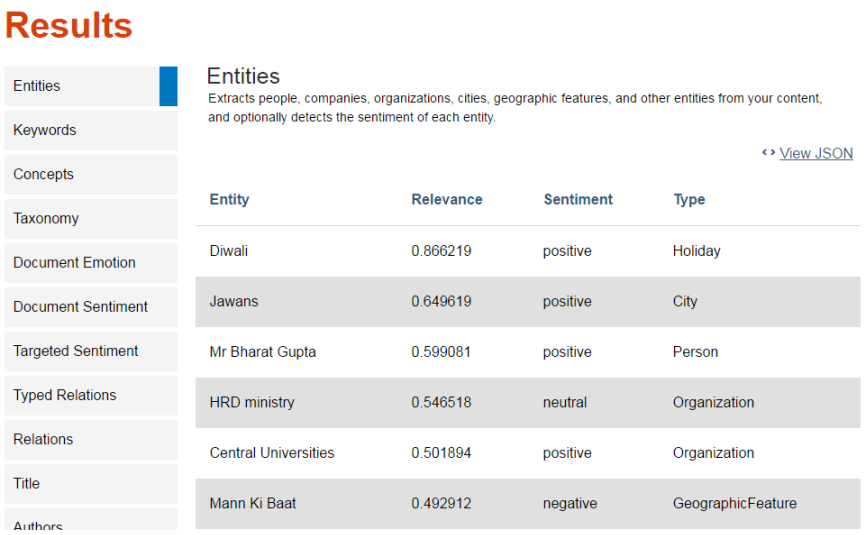

Figure. 2.8 Entity extraction done by AlchemyLanguage

- **Concepts Identification:** This feature identifies general concepts that may not be directly referenced in the text. For example, if a post is about Audi, BMW or Porsche, the concept identified will be 'Automotive Industry'.
- **Entities Extraction:** It extracts entities like person, company, city, organizations *etc.* from the content and optionally assigns sentiment score to each entity.
- **Taxonomy:** This feature classifies the text into a hierarchy which can be up to five levels deep. For example, /business and industrial/advertising and marketing/advertising.
- **Targeted Sentiment Analysis:** Here, the sentiments are detected for user-specified targets present in input text. For example, 'I like cakes but hate wraps'. If targets specified are 'cakes' and 'wraps', then 'cakes' will be assigned a positive sentiment and 'wraps' a negative sentiment.
- **Language Detection:** It detects the language of the text in which it is written.
- **Text Extraction:** This feature works with URL only. On input of URL, tool extracts the text present at that URL.
- **Author Extraction:** This feature also works with URL input. It extracts the author from blog posts or news articles.

- **Title Extraction:** It extracts the page title from a webpage. This feature also works with URL input.
- **Publication Date:** This feature works with URL and tells the date on which the article was published.

2.2.2 Repustate

Repustate was founded on August 1, 2008. It was originally designed for social media analysis [22]. It uses complex language models and grammatical analysis for each language. It uses POS tagger for sentiment analysis. Maximum size that the tool can process is 2048 characters. It only takes text as input and supports 13 languages including English, Spanish, French, German, Italian, Arabic and Chinese. Repustate uses REST API and all responses are in JSON format. Repustate also provides client libraries for languages like Python, Ruby, Java, C# and PHP for development purpose. The system is available online at https://www.repustate.com. Figure 2.9 shows the working of Repustate.

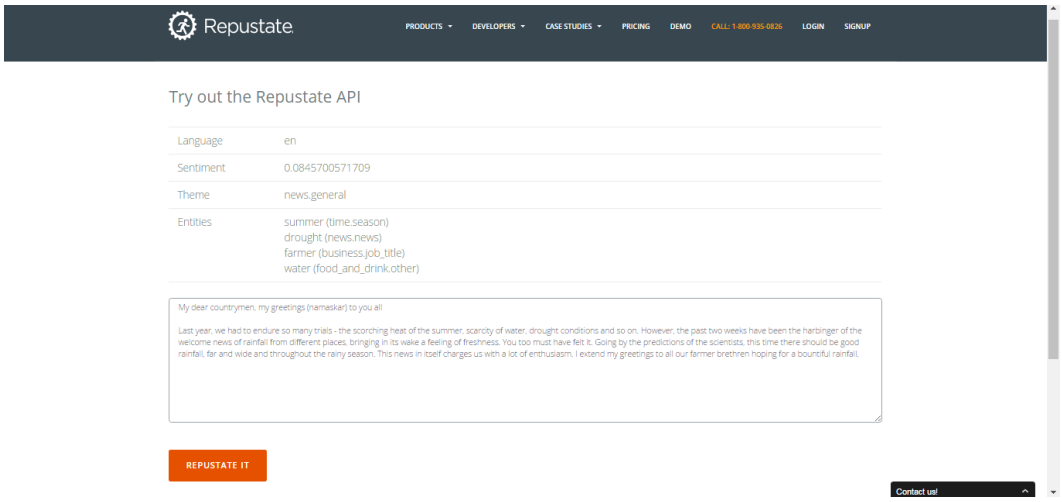

Figure. 2.9 Working of Repustate

Given the text input, the tool has identified the language of text, corresponding sentiment score, theme and the entities present in text.

Following are the features provided by Repustate.

- **Sentiment Analysis:** It determines the sentiment score of the text and scores it in range (-1,1). Emoticons and slang words are given more weight.
- **Sentiment by topic:** It can be used if opinion towards some specific terms in the text is to be determined. For example, *Battery life is good but camera takes time to focus*. Here, two terms may be *Battery life* and *camera*.
- **Named Entity Recognition:** Repustate identifies named entities from the text and classifies them into one of the following themes: arts_and_entertainment, careers, food_and_drink, pets, science, society, automotives *etc.* counting upto 500.
- **Categorizations:** Repustate provides sentiments by categories also. For example, if one of our categories is hotel, one may be interested in knowing about its food, location, staff, price etc. Likewise, there are following categories:

  Hotel: food, price, location, accommodations, amenities, staff.

  Airline: price, staff, in-flight, loyalty.

  Restaurant: price, food, staff, location, atmosphere, events.

  Telecom: price, service, products, staff.

  If someone says *food was good but the staff was a bit rude* and specifies the category *hotel*, as hotel has two aspects in this sentence, *food* and *staff*, *food* will be classified as positive and *staff* as negative.

- **Clean HTML:** In this API call the most important part of the web page will be taken out, removing all tags and any common header or footer content. It takes URL as input.
- **Language Detection:** This API call detects the language of input text. A two letter language code will be returned as output, for English it is en.

2.2.3 Semantria

Semantria is owned by a text analytics company named Lexalytics that was founded by Oleg Rogensky. Semantria offers text analysis through API and Excel Plug-in. Semantria API is sentiment analysis in cloud whereas its Excel Plug-in allows a user to submit data in excel format and get results back in excel. Semantria has machine learning at its base

but it uses other techniques also depending upon the problem in hand. It uses a hybrid approach, *i.e.*, instead of using machine learning alone, it uses a combination of dictionaries, machine learning, pattern files and natural language algorithms. Semantria takes text and URL as input [23]. It supports over 20 languages including English, French, Russian, German and Dutch. With Semantria one can analyze a maximum of 16384 characters at a time. Semantria is available online at https://semantria.com/. Figures 2.10 shows the working of Semantria for URL input.

Figure. 2.10 Identification of Positive, Negative and Neutral words

The system has fetched the text present at the specified URL. Along with that, it has identified positive, negative and neutral words in text. The system also displays the overall sentiment score of the text. The system also extracts the entities and themes present in text.

Following is the list of features provided by Semantria [24].

- **Sentiment Analysis:** Semantria has a very strong sentiment analysis part. Along with providing the overall sentiment and score of the text, it highlights all the positive, negative and neutral words present in the text.

- **Named Entity Extraction:** Named entities extraction means identifying named text figures like people, places, products, brands and organizations.
- **Themes Extraction:** It is determining the context of entities through themes and facets and identifying the topics of discussion. Highly complex text mining techniques are used for this purpose.
- **Categorizations:** This feature is useful for sorting the content into buckets that are relevant to a business. For example, a retail store might be interested in categories like staff, location, parking, stock availability, lighting, pricing etc.
- **Intentions:** This is useful in predicting future behavior. For example, sentence 2.1 reveals a *buy* intent.

  I lost my camera, I need to buy a new one … (2.1)

  Semantria deals with four types of intents: Buy**,** Sell**,** Recommend and Quit. Using intentions will let a business find new customers as well as prevent customer churn.

- **Text Summarization:** It outputs the summary of the text input to it to get a quick grasp over a long document.

2.2.4 iFeel

iFeel was developed by Araujo *et al.* in 2014. The tool compares and combines the existing sentiment analysis methods for output sentiment class [25]. iFeel compares 21 sentiment analysis methods [26] including Emoticons, PANAS–t [27] [28], SentiWordNet [29], Happiness index [30], SentiStrength [31], Senticnet [32], SASA and 13 more. It puts together 19 public datasets that include text from twitter, forums [40], YouTube comments, blogs, reviews *etc.* and makes them available to the research community so that new methods can be compared across different datasets. It supports over 60 different languages [33]. Input to iFeel can be either a simple text or a text file that can be uploaded to the tool and the resultant file can be downloaded in either excel or XML format. The output file includes the sentiment output of all the methods listed above [34]. The file that user uploads can be up to 1000 sentences long which may be up to 500 KBs in size [35] [36]. iFeel is available online at http://www.ifeel.dcc.ufmg.br.

Figure 2.11 shows the working of iFeel. Given an input text, the system has output the sentiment class determined by each of the 21 methods mentioned above. Majority has determined positive, so it will be considered as actual sentiment for the given text.

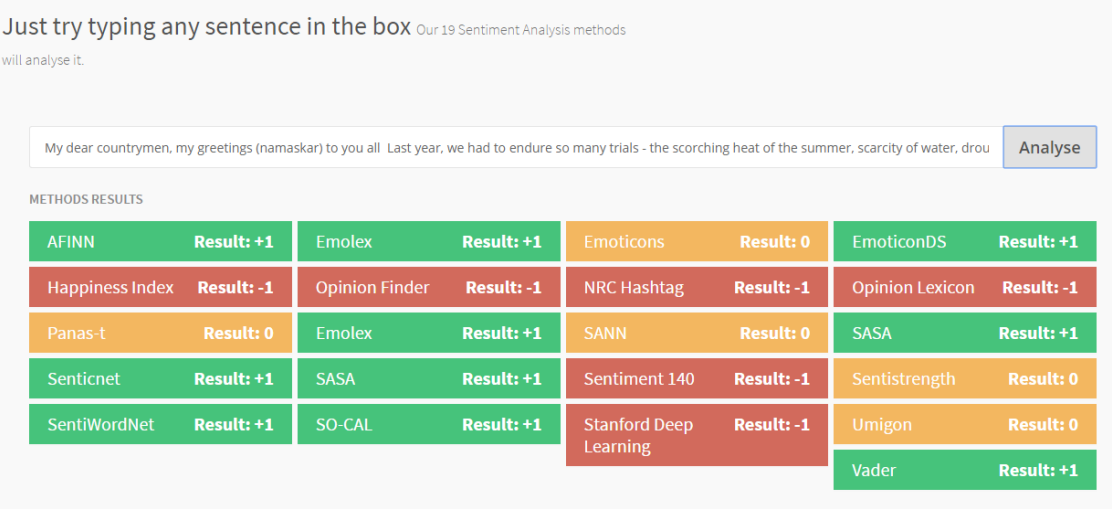

Figure. 2.11 Sentiment output of 21 methods implemented by iFeel

2.2.5 Socroutes

SocRoutes was developed by Kim et al. in 2014 to find a safer, friendlier, and more enjoyable route based on the sentiments inferred from real-time, geo-tagged reviews from Twitter. There are some navigation systems that suggest routes based on the shortest distance. SocRoutes finds a safer and more enjoyable route with marginal increase in total distance, by searching paths with least total count of negative sentiments. SocRoutes recommends routes based on regional context inferred from Twitter. It utilizes the complete set of real-time geo-tagged data from Twitter and suggests routes by avoiding places with extremely negative sentiments, helping to prevent any negative future experience [37].

SocRoutes takes source, destination, mode of travel, the baseline sentiment threshold & the maximum number of waypoints (1 to 8) as input & outputs a safer path but at the cost of path length. Socroutes is available online at http://pororo.kaist.ac.kr/socroutes/. Figure 2.12 depicts the working of Socroutes. In this figure, the user has mentioned Chicago University as source, Humboldt Park Library as destination and the mode of travel is bicycle. The system has output a safe path of safety score 100 with a total distance of

10.095 km and 0 crime spots. This path may be slightly longer the actual path but with more crime spots.

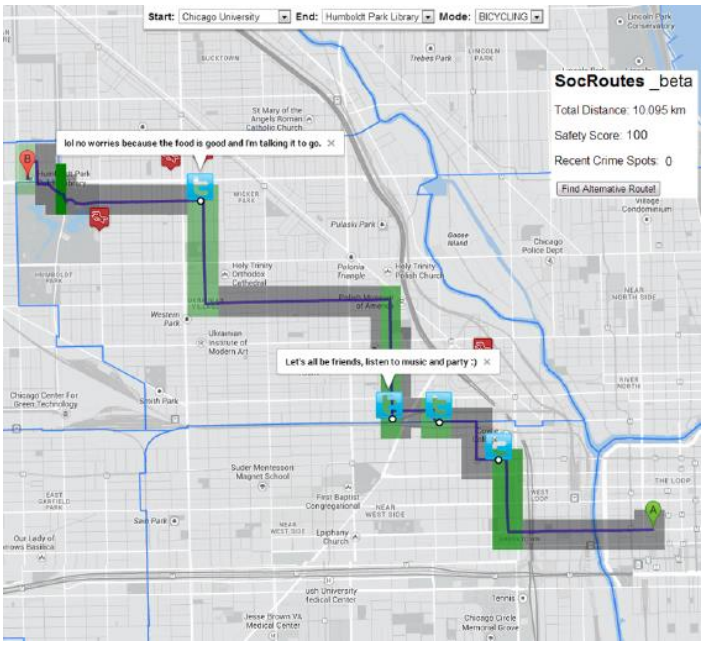

Figure. 2.12 Working of Socroutes

Table 2.1 shows a comparative analysis of all the above discussed tools based on their features.

Table 2.1 Comparative Analysis of Sentiment Analysis Tools

	Features	**AlchemyAPI**	**Repustate**	**Semantria**	**iFeel**	**Socroutes**
1	Sentiment Analysis	✓	✓	✓	✓	✓
2	Emotion Detection	✓	×	×	×	×
3	Slang & Emoji	×	✓	×	×	×
4	Keywords Extraction	✓	×	×	×	×
5	Entity Extraction	✓	✓	✓	×	×
6	Concepts Identification	✓	×	×	×	×
7	Language Detection	✓	✓	×	×	×
8	Title Extraction	✓	×	×	×	×
9	Text Extraction	✓	×	×	×	×
10	Author Extraction	✓	×	×	×	×

11	Targeted Sentiment Analysis	✓	✓	×	×	×
12	Targeted Emotion Analysis	✓	×	×	×	×
13	Taxonomy	✓	×	×	×	×
14	Categorized Sentiment Analysis	✓	✓	×	×	×
15	HTML Cleaning	✓	✓	×	×	×
16	Publication Date Extraction	✓	×	×	×	×
17	Text Summarization	×	×	✓	×	×
18	Themes Extraction	×	×	✓	×	×
19	Feeds Extraction	✓	×	×	×	×
20	Part Of Speech Tagging	×	✓	×	×	×
21	Intention Detection	×	×	✓	×	×

The sentiment analysis tools have been compared for a total of 21 features. Analyzing the above table, it can be observed that AlchemyAPI provides maximum number of features and Socroutes the least, but the contribution done by this tool is really applaud able as the system is application based that is using tweets to find a safe route. iFeel itself has provided the exposure of 21 sentiment analysis tools. Semantria is again a tool with its unique feature set and Repustate can process emoticons and slang. But it is concluded that AlchemyAPI is best of all the available tools.

Chapter Summary

In this chapter various tools and techniques for sentiment analysis were discussed. The tools are available online and can be easily used by anyone without manually putting any efforts. In next chapter, the existing problems, the objectives and methodology for solving those problems are discussed.

Chapter 3

Problem Statement

Sentiment analysis is the process of mining people's opinions expressed on social media like Twitter, Facebook, blogs, forums *etc*. Theses opinions help a lot in decision making process, for example, people before buying a new product or before watching a movie check online reviews to have an idea whether it is worth to spend time and money. These sentiments can be used by companies to get user opinions about their products. Internet is a huge repository of such data. A lot of content is generated per minute, per day, per month and multiplying over the years. On Twitter more than 500 million tweets are posted each day and this volume grows during important events like Elections, World Cup, social movements or a public aggravation reaching up to 1 billion tweets a day. As of January 2017, there are 317 million active Twitter users 80% of which use on mobile phones. In US there are 67 million Twitter users covering major portion of overall Twitter users and India having 41 million users. 37% of all users are in age range 18 to 29 and 25% in range 30 to 49. With huge Twitter usage a lot of data is generated on Twitter every day. Fortune is that this data holds an immense value. Tweets data besides containing the subjective information also consists of some other useful information like timestamps which can be used for trend analysis. It also consists of name and user id of the person mentioned in the tweet which can be very useful in getting opinions about that person. Location information if available helps to gauge trends over different geographical locations. But manual sentiment analysis of such huge and varied data is very difficult and time consuming process. So, there is a need for an automated sentiment analysis system.

In this research two problems have been addressed. These are as follows.

- There exist a number of online automated sentiment analysis systems providing a number of features, but still they lack in functionality somewhere. So, there is a need to build a sentiment analysis system with enhanced feature set. The system

developed in this research work is an improvement over the existing systems entertaining a number of quality features.

- A real time sentiment analysis system was needed for processing Twitter data in real time. The customized system mentioned above had some problems like it had a limit on call usage per minute. As the problem is real time, it was not possible to use that system. Bearing this in mind a real time Twitter sentiment analysis system has been developed using machine learning. Five supervised learning models have been tested on labeled data available online.
- Machine learning has been very popular for sentiment analysis since last few decades. But because of some limitations like manual feature engineering and accuracy concern, deep learning is more a choice for sentiment analysis. A deep learning model has also been prepared for doing sentiment analysis.
- The proposed real time sentiment analysis system has been tested on the existing application of 2017 Punjab Elections.

3.1 Objectives

The main aim of this of this research work is sentiment analysis. In order to carry out this task following objectives have been accomplished.

- To compare existing sentiment analysis systems based on their features.
- To build a customized sentiment analysis system using the APIs of existing systems to enhance their feature set and to accommodate most qualitative features.
- To build machine learning and deep learning models for the proposed sentiment analysis system.
- To build a real time sentiment analysis system for analysis of Twitter data.
- To validate and test the proposed sentiment analysis system for Punjab Elections 2017.

3.2 Methodology

To achieve the objectives mentioned above, following methods have been used.

3.2.1 To compare existing sentiment analysis systems

5 online sentiment analysis systems have been surveyed on the basis of their feature set, techniques they use for sentiment analysis, the way of input \textit{etc.} These systems are AlchemyAPI, Repustate, Semantria, iFeel and Socroutes. Some qualities and limitations of these tools have been outlined.

3.2.2 To build a customized sentiment analysis systems

A customized sentiment analysis system has been developed using APIs of two existing systems AlchemyAPI and Repustate. Python SDKs of these tools are used for development purpose.

3.2.3 To build machine learning and deep learning models for sentiment analysis

Five machine learning and three deep learning models have been trained and tested on annotated dataset. Dataset was collected from online gitub repository. Data was preprocessed and machine learning models were trained on different features. To select best features, information gain has been used. For deep learning no feature selection was done. Then best performing machine learning model was finally selected for sentiment analysis. Machine learning models have been used from Python's \textit{scikit-learn} library and deep learning models from \textit{tensorflow} library.

3.2.4 To build a real time system for sentiment analysis of Twitter data

A real time Twitter sentiment analysis system has been developed for analyzing Twitter data. Twitter API is used for tweets collection. Backend is developed in Python and user interface in HTML using JavaScript and AJAX technologies.

3.2.5 To test the real time system for 2017 Punjab Elections

To test the real time system, tweets about 2017 Punjab elections were fetched using Twitter API in real time. Then tweets are separated and analyzed party wise and the sentiment results are presented to user in real time. These tweets are side by side stored in

a CSV file for final predictions. Then the system has been used for making prediction on the data collected over 24 days and the system stood successful in predicting the election results.

Chapter 4

Customized Sentiment Analysis System

The system discussed here is customized sentiment analysis system. It is developed by merging two existing sentiment analysis systems *AlchemyAPI* and *Repustate*. Though both these tools provide a good count of features but still they lack in functionality, for example, AlchemyAPI cannot analyze emoticons and slang words which is done by Repustate. Repustate also does domain specific sentiment analysis which is again missing in AlchemyAPI. AlchemyAPI itself is a very good tool providing a number of other important features which Repustate is missing. So to entertain the features of both tools on a common platform, the tools are merged together to form a customized sentiment analysis system.

4.1.1 Tools Used

- **API Key**

  Features of the tools cannot be directly accessible. To authenticate access to these tools, user has to register for an API key on the official website of these tools. Then the key issued is used every time an API call is made to the tool for performing any type of function.

- **Software Development Kit (SDK)**

  Both tools provide SDKs in a number of programming languages like Python, Ruby, Java, node.js and C# for developing a personalized sentiment analysis system. For current system Python SDK have been used.

The backend is implemented using Python and user interface is designed in HTML using JavaScript and AJAX.

4.1.2 Architecture of System

The detailed architecture of customized sentiment analysis system is shown in figure 4.1.

The system can take either text or URL as input from user. User while feeding input to the system, has to specify the query function. Query function is the feature for which the input is to be analyzed. The system provides access to 13 features but all features are not compatible with both text and URL. Some of them work with both text and URL and some work with either of the two. Both the input value and the query function are sent to Flask server. Flask server is a Python Web Framework for serving user requests. It is written in Python and is based on Werkzeug WSGI toolkit and Jinja2 template engine. Flask server sorts the requests according to whether it is a text and a URL. Text and URL requests are handled by separate modules. If it is text it is forwarded to text processing unit otherwise to URL processing unit. While text processing module directly processes the text for the intended function, URL processing module processes the text at the specified URL. To perform any function, API calls are made and the results are returned in JSON format. From here these results are directed back to Flask server and then back to user for display.

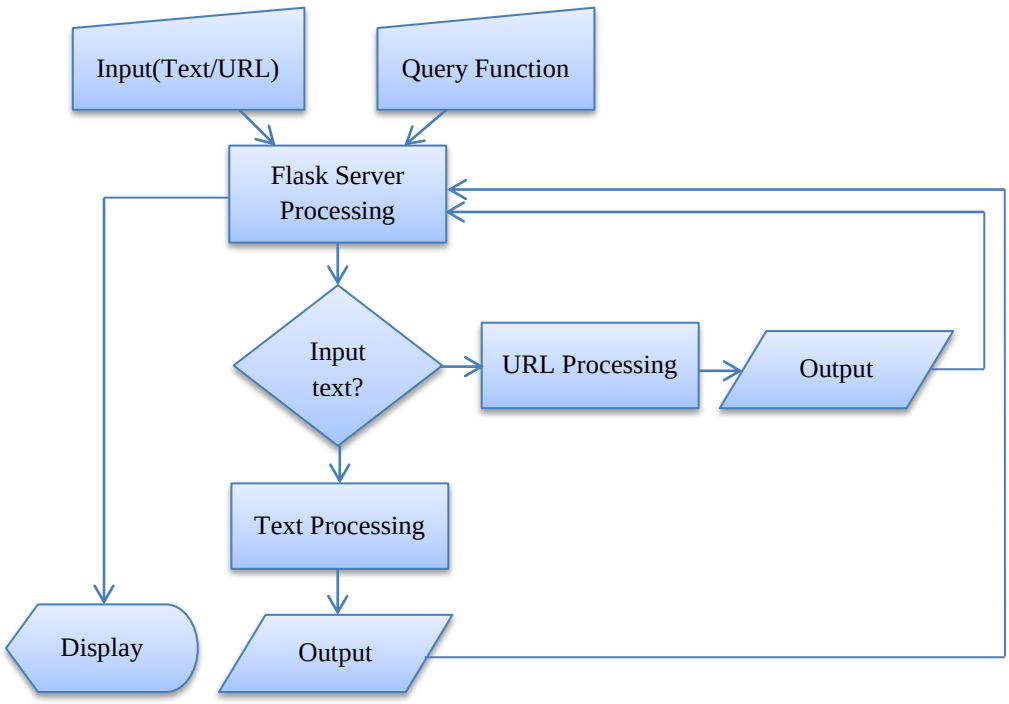

Figure. 4.1 Architecture of Customized Sentiment Analysis System

4.1.3 Features provided by the system

The system provides 13 features in all, 11 features belong to *AlchemyAPI* and 2 are from *Repustate*. Following is a detailed description of these features along with examples.

i. Sentiment Analysis

It calculates the sentiment score of the input submitted to it. Score varies over the range (-1,1) with -1 for most negative sentiment and 1 for most positive. While doing sentiment analysis emoticons and slang are given more weightage. Figure 4.2 depicts the working of system for sentiment analysis.

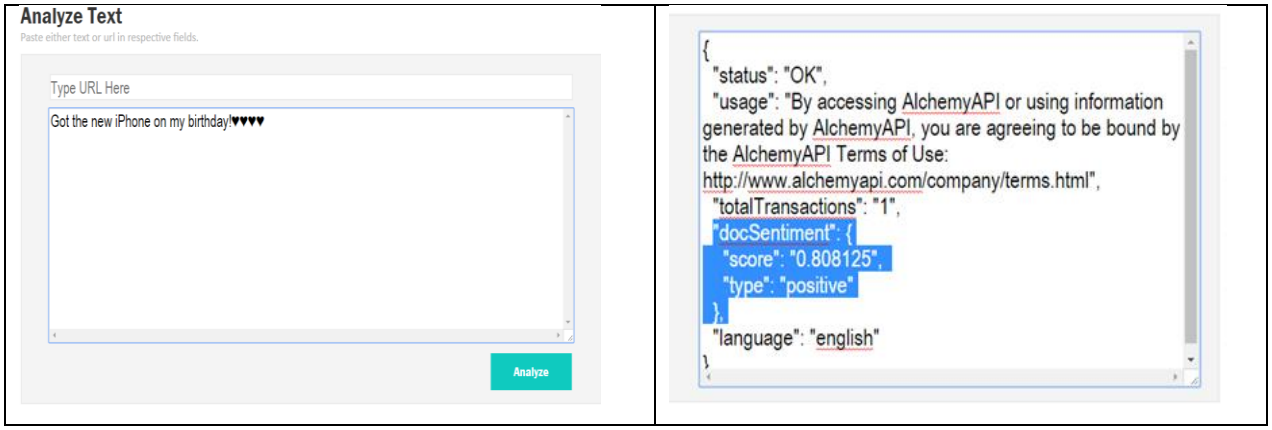

Figure. 4.2 Sentiment Analysis

ii. Emotion Detection

Sentiment analysis can be done at a more fine level, *i.e.*, on the basis of emotions expressed in the text. The system considers these five emotion classes, *i.e.*, Fear, Anger, Joy, Disgust and Sadness. The text is classified into of these classes. Figure 4.3 depicts this feature.

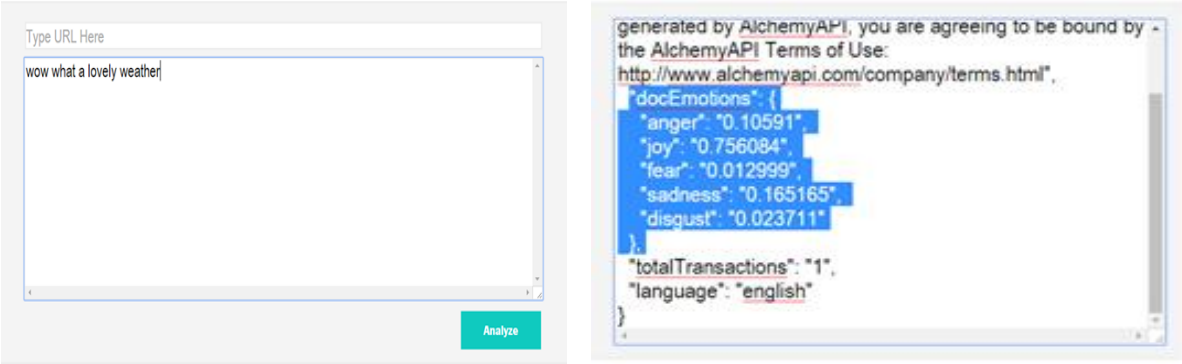

Figure. 4.3 Emotion Detection

As the input text was in positive sense, *joy* is having highest score than any other emotion.

iii. Entity Extraction

This feature identifies the entities present in the text. Entities may be a person, city, company or an organization. The system optionally assigns sentiment to entities. This feature is depicted in figure 4.4.

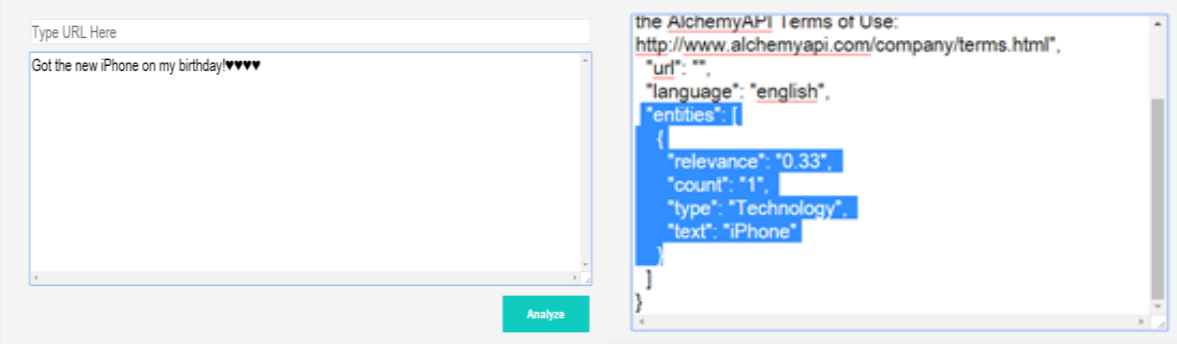

Figure. 4.4 Entity Extraction

iv. Concepts Identification

It identifies the general concepts that may not be directly referenced in the text. For example, if the input text or URL is about BMW, Audi or Porsche the concept will be 'Automotive Industry'. The tool also calculates the relevance and evidence of concepts. Evidence is the world database in which that word is found. Figure 4.5 displays the concepts identification feature. Here, URL is given as input.

Figure. 4.5 Concepts Identification

v. Targeted Sentiment Analysis

For this feature to work, along with input text target words need to be specified towards which the sentiments are to be determined. Figure 4.6 depicts sentiment analysis for two targets *songs* and *movie*. At most three targets may be specified.

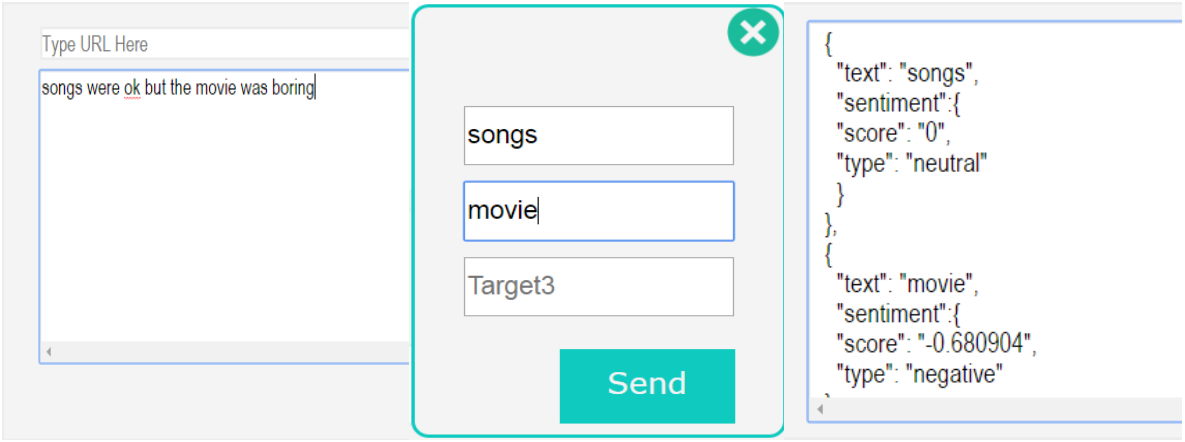

Figure. 4.6 Targeted Sentiment Analysis

vi. Taxonomy

This feature classifies the text into a five level deep hierarchy of categories. Figure 4.7 depicts this feature.

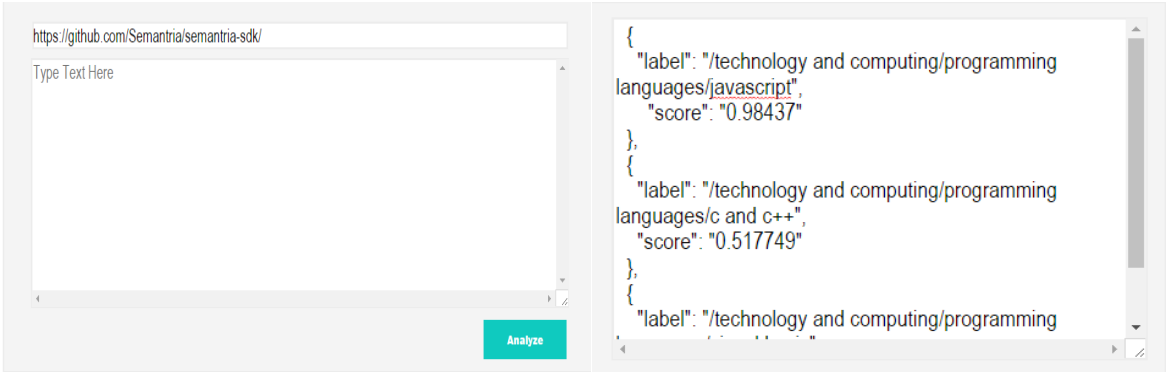

Figure. 4.7 Targeted Sentiment Analysis

vii. Text Extraction

This feature works only for URL. It takes URL as input and fetches the text present at that URL. Figure 4.8 displays this feature.

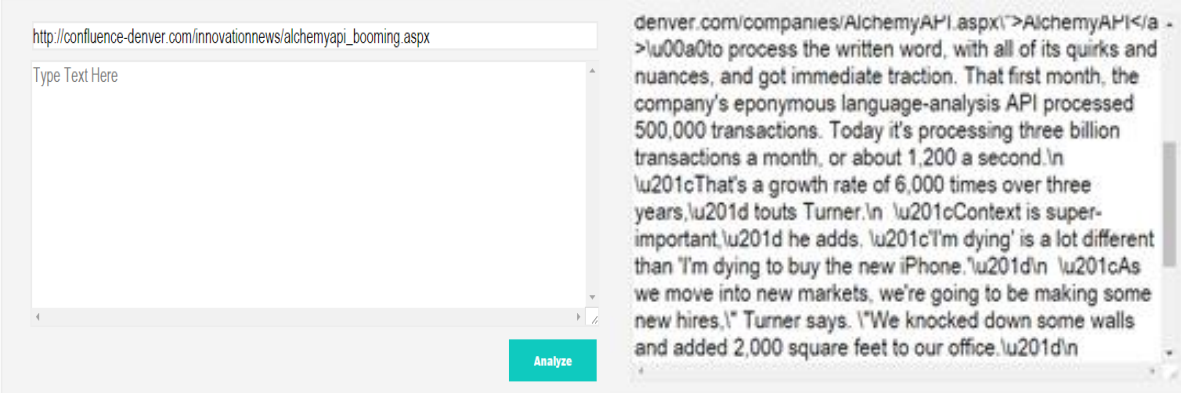

Figure. 4.8 Text Extraction

viii. Author Extraction

This feature also works with URL. Given the URL of a blog post or a news article, it extracts the author of that post. Figure 4.9 depicts this feature.

Figure. 4.9 Author Extraction

ix. Categorized Sentiment Analysis

This feature does category-wise sentiment analysis of data, *i.e.*, user has to specify the domain for which the text is to be analyzed. Following are the categories for which the system works.

Hotel: location, food, price, staff, accommodations, amenities.

Airline: staff, loyalty, price, in-flight.

Restaurant: food, price, staff, atmosphere, location, events.

Telecom: service, price, staff, products.

Figure 4.10 shows the working of categorized sentiment analysis. For the present case category *Hotel* is selected.

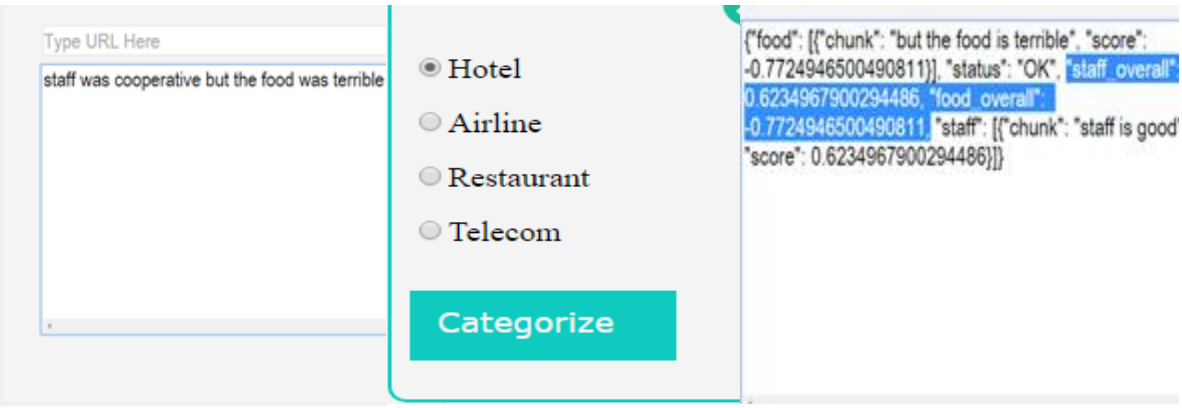

Figure. 4.10 Categorized Sentiment Analysis

x. Keywords Extraction

It identifies the important keywords present in text. These keywords are associated with sentiment and are ranked by their relevance in the text. For example,

Input text: '*Larry Page is the founder of Google*'
Output:

Keyword → 'Google', Relevance → 0.986899, Sentiment → neutral.
Keyword → 'founder', Relevance → 0.942144, Sentiment → neutral.

xi. Language Detection

It detects and outputs the language the text is written in. Along with, it displays the native speakers of the language. For example,

Input text: '*Où est-ce que tu habites?*'
Output: language → French,
Native speakers → 80 million

xii. Title Extraction

This feature works with URL only. It determines the title of an HTML document or a webpage. For example,

Input url: '*http://dbpedia.org/page/Python_(programming_language)*'
Output: Python (programming language)

xiii. Feeds Extraction

It extracts RSS and atom feeds from web pages and returns their links. For example,

Input url: '*https://semantria.readme.io/*'

Output:

http://dbpedia.org/data/Python_(programming_language).atom,

http://dbpedia.org/data/Python_(programming_language).rdf,

http://www.w3.org/1999/02/22-rdf-syntax-ns#type

Chapter Summary

The system discussed in this chapter was developed by merging two existing sentiment analysis systems. The system is an improvement over existing systems with an alleviated feature set. In next chapter, a real time sentiment analysis system has been developed for analyzing Twitter content.

Chapter 5

Building a Machine Learning and Deep Learning based Sentiment Analysis System

5.1 Development Platform Introduction

Python has been used as the base language for building proposed sentiment analysis system.

5.1.1 Python

Python is a powerful, fast and easy to learn object-oriented programming language. It is an open source language developed under OSI-approved open source license. Python makes use of high-level efficient data structures and it is an effective approach to object-oriented programming. Because of its elegant syntax and interpreted nature, Python can be used on a number of platforms and for developing a number of applications. Python can also be used as an extension language for customized applications as it can be easily extended to include new data types and functions that are implemented C or C++.

i. Python for Sentiment Analysis

Python plays a very important role in sentiment analysis. It has a very extensive library providing access to a wide range of functions. It provides a number of modules that can be used for sentiment analysis. Python provides Natural Language Tool Kit (NLTK) which is a massive Python library aimed at easing Natural Language Processing task. NLTK aids with a number of functions that are required for text processing. For example,

- *word_tokenize()* is used for splitting a sentence into words.
- *WordNetLemmatizer* is used for grouping together all the inflected forms of a word.
- *BigramCollocationFinder* is used for finding most frequently occurring bigram collocations within a corpus.

- *BigramAssocMeasures* is used to score n-grams based on their frequency in the corpus.
- *NaiveBayesClassifier* is a supervised machine learning classifier.
- *MaxentClassifier* is also a supervised machine learning classifier.
- *SklearnClassifier* is a machine learning classifier that requires scikit-learn package.

Python provides a robust machine learning library named *scikit-learn*, *sklearn* for short. Built on the SciPy module of Python, Scikit-learn provides a number of supervised and unsupervised machine learning algorithms. Some of the supervised scikit-learn algorithms used in this research are *MultinomialNB, BernoulliNB* and *LinearSVC*. For building neural networks, Python provides a deep learning library named *tensorflow*.

5.2 Sentiment Analysis System

The following section discusses the model building and testing for sentiment analysis. Different machine learning and deep learning models have been trained and tested on annotated data. For machine learning, features have been manually crafted but deep learning is a self-learning process with no requirement for manual feature designing. The underlying process for system building is discussed as follows.

5.2.1 Model Building

Figure 5.1 displays the process for building sentiment analysis model and algorithm used for building the model is discussed in Algorithm 1.

The algorithm explains the pseudo code for building the sentiment analysis model. It completes this task by calling some procedures which are explained in subsequent sections.

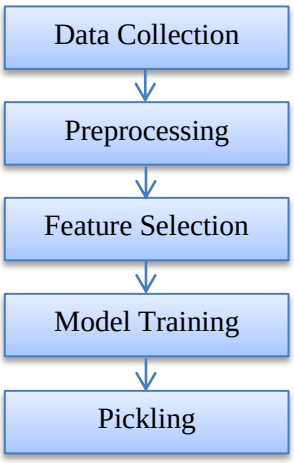

Figure. 5.1 Process for building a sentiment analysis model

ALGORITHM 1: Sentiment Analysis Model Building (T)

Input: Training Data (T)
Output:

1. Training data collection
2. Pre-processing of training data ▷ Call to procedure pre-process(T)
3. Feature selection ▷ Call to procedure feature-selection (T')
4. Selection of best feature words ▷ Call to procedure info-gain (T'')
5. Building of feature set ▷ Call to procedure feature-set (T'', B)
6. Training the model on feature set F

---

i. Data Collection

Data is required for training the models. So, labeled data is needed for that purpose. Data can be manually labeled or labeled data may be available online. Training data for current task has been collected from online github directory and it is in labeled form. It consists of two columns, tweet text and its associated sentiment class. There are three sentiment classes in the dataset *viz.* positive, negative and neutral. Tweet content consists of general tweets without focusing any particular domain.

ii. Data Preprocessing

Data that was collected could not be directly used for sentiment analysis as it contained a lot of junk content. This junk may be @ (mentions), # (hashtags), RT (retweets), URLs or Unicode characters which do not play any role in efficient learning, instead unnecessarily increase the learning time. So, this content needs to be removed. Following are the issues addressed in this phase.

- **Removal of Irrelevant Content:** Characters like @, # and RT and other non-sensible Unicode characters are stripped off from text. For example, @George was converted to George.
- **Stripping of Repeating Characters:** There are many other variations found in social media content like it is trend to use repeated characters in a word. For example, '*sooo sweeeet*' for '*so sweet*'. This problem was solved by replacing repeated characters to maximum two repetitions, *i.e.*, converting '*sooo sweeeet*' to *'soo sweet'*.
- **Stripping Punctuations and spaces:** The text fetched is either prepended or appended with punctuation marks or some extra spaces, for example, '*Awesome weather today!!!*'. This form changes the original meaning of word, *i.e.*, the word *today* is now considered as *today!!!*. Stripping of such unwanted characters brings the words back to their original form. Stripping is optionally tested for the system performance.
- **Stop Words Removal:** Stop words are commonly used words that are often ignored while text processing. When doing sentiment analysis, these can be removed considering that they don't hold any sentiment value. Stop words list include words like *is*, *am*, *are*, *was*, *were*, *being*, *having*, do *etc.* These are also optionally tested for performance but actually they have been retained.
- **Case Loweing:** The whole text was converted to lower case so that the words *'Amazing'* and *'amazing'* are considered same, otherwise these would be considered as different.

Sometimes emoticons are also considered junk but actually they convey important information. Commonly used emoticons are :), :-) for smile, :D for happiness, :|

for neutral and :(, :-(, :'(for representing sad emotions. These have been retained for analysis.

After performing above tasks, data becomes normalized and suitable for further processing. Procedure 1 shows the pseudo code for accomplishing preprocessing task.

Procedure 1 pre-process(T)

```
Input: Training data (T)
Output: Preprocessed training data (T')
// Preprocessing of each row in training data
for each r ∈ T
    t = r(text)
    s = r(sentiment)
    remove characters out of range (0,127)
    lowercase all characters of t
    remove URLs, mentions and hash tags
    strip double quotes from t
    add (t', s) to T'
return (T')
```

Preprocessing is done for tweet by tweet and a modified version of training data T' is returned to the main algorithm.

iii. Feature Selection

Feature selection is an important part of sentiment analysis process. Performance of a model varies with varying feature set. A number of features were tested for performance analysis of the models and only the best features where models were most accurate were selected for finally building the models. Except lemmatization, no other features discussed in this phase have been used for deep learning models as they learn by building their own feature representations. Features that are tested for this system are discussed below.

Unigrams

Unigrams are the base of all feature selection techniques. With this approach single words are considered for feature building. For example, if a sentence is '*Awesome weather today*' and its corresponding sentiment is *positive*, it will be tokenized into individual words called unigrams as ['*Awesome*', '*weather*', '*today*']. So, the feature set will conduct information with the presence of words, *i.e.,* [{'contains(Awesome)': True, 'contains(weather)': True, contains(today): True}, positive]. This approach will work for some words like *Awesome* used to be positive in almost all cases but what about other words. They may be equally present in all the three classes. Machine learning classifiers like Naïve Bayes learn very intelligently by considering probability distribution of words over all the three classes, *i.e.*, if the probability of a word to be positive is highest than any other class, tweets containing this word will be classified as positive.

Feature choices discussed below add more information to unigram features.

Unigrams and Bigrams

In this approach, along with using unigrams, bigrams are also included in making of feature set. In unigrams single words were considered. But here, proceeding with previous approach, two words combinations are also included. Feature set is now of this form, [{'contains(Awesome)': True, 'contains(weather)': True, contains(today): True, 'contains(Awesome weather)': True, 'contains(weather today)': True }, positive].

Lemmatization

Lemmatization is the process of reducing a word to its root word. For example, the word *good* is having *better* and *best* as its two inflected forms. During lemmatization the words *better* and *best* are converted to *good* as all these words belong to same sentiment class, thus increasing the probability of a word to be classified into its intended class.

Part Of Speech (POS) Tagging

Using this approach, unigrams and their associated part of speech tags are also considered as features. Part Of Speech tags or POS tags are lexical classes assigned to words. These include *Nouns*, *Verbs*, *Adjectives*, *Adverbs etc.* and add more information to a classifier because a word may belong to different sentiment classes based on its part of speech. For example,

This is a love story … (5.1)

I love this song … (5.2)

In sentence 5.1, POS tag of the word *love* is *Verb* and the sentence is neutral. In sentence 5.2, the same word has POS tag *Adjective* and the sentence is in positive sense.

Treating Punctuations as Separate Unigrams

Sometimes punctuations can be considered as separate unigrams as punctuations may hold sentiments. For example, '*???*' are mostly used to express negative opinion and '*!!!*' for positive opinion. There is no hard and fast rule for using punctuation marks but this pattern has been observed.

Procedure 2 depicts the pseudo code for feature selection. As the performance of model was varying with varied feature set, so only those feature combinations for which the model was most accurate are accommodated in the procedure below. These are the features that are finally retained for model building.

Procedure 2 feature-selection (T')

```
    Input: Preprocessed training data (T')
    Output: Feature data (T'')
    // Make word features from each row of training data
   for each r ∈ T'
       t' = r(text)  // Extraction of text from row
       s = r(sentiment)
       WL = tokenize(t')  // Tokenization of text; WL is word list
       // For each word in WL replace more than twice repeating
characters to twice
       for each w in WL
```

```
        w' = replaceMoreThanTwo(w)
        p = posTag(w')
        add (w', p) to WL'
    add (WL', s) to T''
return(T'')
```

Processing is done for each word. For that, firstly the tweet is tokenized and then feature selection is done. Processing includes character repetition to twice and associating POS tag with each word. This feature data is then returned to the main algorithm.

Information Gain

Using Information Gain for feature selection a tremendous increase in accuracy has been observed. Information gain weighs words by their relevance in text. A total of 8500 unigram features were generated in procedure 2 but research has shown that all are not equally important. Some feature words unnecessarily choke the processing speed and hinders the accuracy of algorithm. The procedure below measures each feature word by information gain score and selects 3000 most informative words for further analysis. Metric Chi square is used for calculating information gain [38].

The pseudo code for selecting top k features using information gain is shown in Procedure 3.

Procedure 3 info-gain (T'')

```
Input: Feature data (T'')
Output: k best word features (B)
define set WF  // WF is a set of features
for each r in T''
   WL' = r(text)
   for each w' in WL'
      add w' to WF
      count(w') = count(w') + 1
      posCount(w') = posCount(w') + 1
      negCount(w') = negCount(w') + 1
```

```
        neuCount(w') = neuCount(w') + 1
    for each w' in WF
      compute pos_score(w'), neg_score(w'), neu_score(w') using chi-sq
      scr(w') = pos_score(w') + neg_score(w') + neu_score(w')
    sort(WF) by score
    B = select top k words from WF
    return B
```

This procedure takes feature data T'' prepared in procedure 2. Then for each word, it calculates the positive, negative, neutral and overall count. Based on these counts it calculates the respective scores of each word using *chi-square* and then calculates the overall score of the word. Then it sorts the feature words by their score and selects the top k features from the feature set which are returned to the final algorithm.

Now, as the best words have been identified, only those words will be retained in the dataset for further analysis. Procedure 4 explains the strategy for retaining best words in tweets and stripping out others.

Procedure 4 feature-set (T'', B)

```
Input: Feature data (T''), List of k best words (B)
Output: Feature set (F)
for each r in T''
   WL' = r(text)
   s = r(sentiment)
   define set f
   for each w' in WL'
      if w' in B
         add w' in f
   add (f, s) to F
return F
```

The procedure takes feature data and list of best words as input. It then compares each word in feature data with the list of best words and keeps that word only if it is present in best words otherwise discards it. The procedure is called as feature selection as it selects the features from those available.

iv. Model Training

Dataset used for models building consists of 1900 tweets in all 70% of which have been used for training and 30% for testing the models. There are 686 positive tweets, 674 negative and 542 neutral tweets in the dataset. So, overall this is a balanced dataset. Tweet distribution of the collected dataset is shown in table II.

Table 5.1 Total Number of Training and Testing Tweets

Tweet Count	Training Data	Testing Data
Positive Tweets	482	204
Negative Tweets	477	197
Neutral Tweets	372	170
Total Tweets	1331	571

Five machine learning models were trained and tested on the above mentioned features. These models are Naïve Bayes, Multinomial Naïve Bayes, Bernoulli Naïve Bayes, Maximum Entropy and Support Vector Classifier. Three deep learning models including 3 layer Perceptron, Recurrent Neural Network (RNN) and Convolutional Neural Network (CNN) have also been trained and tested on labeled data.

v. Pickling

Pickling is the process of saving and loading models or data to and from disk. An image of models resulting from previous phase is saved to disk and can be reloaded whenever required. The process is also known as model persistence. Pickling saves a lot of time as it avoids re-training of models which is a very time consuming process. A library named *pickle* is available in Python to entertain this feature.

5.2.2 Model Testing

The pickled models have been tested on 30% of the collected data.

i. Testing Machine Learning Models

Table 5.2 shows the accuracy of five machine learning models on test set for different features.

Table 5.2 Accuracy of Machine Learning Models on Different Features

	Features	Naïve Bayes	Multinomial Naïve Bayes	Bernoulli Naïve Bayes	Maximum Entropy	Support Vector Machines
1	Unigrams	67.8	68.5	63.4	60.1	70.1
2	Unigrams and Information gain	75.1	77.8	78.1	73.7	68.1
3	Unigrams, Stripping and Information gain	74.8	77.1	77.6	73.4	69.9
4	Unigrams, Information gain and Lemmatization	75.3	76.4	78.1	73.9	66.6
5	Unigrams, Information gain, Stripping and Lemmatization	74.6	76.0	77.6	72.7	68.3
6	Unigrams, Bigrams and Information gain	75.0	73.7	53.1	70.4	69.2
7	Unigrams, POS Tag and Information gain	75.1	77.8	78.1	73.7	68.1
8	Unigrams, Stop words removal and Information gain	73.2	69.9	63.1	68.8	62.9
9	Unigrams, treating punctuations as separate unigrams and Information gain.	74.4	76.0	76.4	72.0	68.7

10	Unigrams, limiting more than 2 occurrences of a character to twice and Information gain	75.8	78.1	78.1	73.6	67.4

From above table it is observed that classifiers Naïve Bayes (NB), Multinomial Naïve Bayes (MNB) and Bernoulli Naïve Bayes (BNB) are most accurate when unigram features are measured by information gain and limiting their character repetitions to twice. MNB is equally accurate when considering POS tags and Lemmatization as features. Maximum Entropy (ME) classifier is most accurate with Lemmatization and Support Vector Machines (SVM) when stripping is used. Finally it is concluded that Bernoulli Naïve Bayes (BNB) performs best out of all the listed models and the feature set which is most suitable consists of unigram features measured by Information gain and limiting the characters repeating more than twice to twice only. This feature set is finally selected for the proposed system.

Table 5.3 shows the evaluation of all the five classifiers using precision, recall and f1-score.

Table 5.3 Evaluation Metrics Precision, Recall and F1-score

Metrics	**Naïve Bayes**	**Multinomial Naïve Bayes**	**Bernoulli Naïve Bayes**	**Maximum Entropy**	**Support Vector Machines**
Precision	0.794	0.802	0.784	0.770	0.700
Recall	0.764	0.786	0.790	0.742	0.674
F1-score	0.765	0.790	0.786	0.744	0.682

Again BNB is having maximum recall value as compared to others. So, BNB is selected as the sentiment model for this sentiment analysis system. Table 5.4 displays the confusion matrix for all the five classifiers.

Table 5.4 Confusion Matrix

		Naïve Bayes			**Multinomial Naïve Bayes**			**Bernoulli Naïve Bayes**			**Maximum Entropy**			**Support Vector Machines**		
		Predicted														
Actual		-1	0	1	-1	0	1	-1	0	1	-1	0	1	-1	0	1
	-1	176	1	20	165	1	31	148	9	40	169	1	27	141	3	53
	0	16	141	13	9	147	14	3	160	7	11	139	20	20	113	37
	1	84	4	116	67	3	134	55	11	138	90	2	112	58	15	131

From above table, it can be observed that for positive class prediction, Naïve Bayes classifier is most accurate and for other two classes, accuracy of Multinomial Naïve Bayes is highest.

ii. Testing Deep Learning Models

Table 5.5 shows the performance of three deep learning models 3-Layer Perceptron, Convolutional Neural Network and Recurrent Neural Network on test set for different iterations.

Table 5.4 Accuracy of Deep Learning Models on Different Iterations

Number of Iterartions	**3-Layer Perceptron Model**	**Convolutional Neural Network**	**Recurrent Neural Network**
20	57.44%	41.51%	35.2%
20	60.52%	42.38%	35.61%
20	65.68%	42.7%	35.9%
Average Accuracy	**61.21%**	**42.2%**	**35.57%**
25	80.7%	40.81%	35.7%
25	81.6%	41.51%	36.2%
25	88.67%	41.68%	36.6%
Average Accuracy	**83.66%**	**41.33%**	**31.17%**
30	89.32%	42.38%	34.33%
30	92.64%	43.26%	35.9%

30	96.32%	43.78%	36.2%
Average Accuracy	**91.43%**	**43.14%**	**35.48%**

From above table, it is observed that the performance of 3-layer perceptron model gets improved as the number of iterations are increased, whereas for others the effect is negligible. It can be concluded that 3-layer perceptron model is most accurate of all other deep learning models and machine learning models as well.

Chapter Summary

In this chapter, the process of building a sentiment analysis model is discussed. A number of machine learning and deep learning models have been trained and evaluated for performance. Various metrics have been used for this purpose and it is observed that deep learning is a better approach to sentiment analysis than machine learning. In next chapter, a case study on 2017 Punjab elections is done. A real time sentiment analysis system is developed that employs BNB for sentiment classification task.

Chapter 6

A Case Study of 2017 Punjab Elections

India is a large country of 29 states and 7 Union Territories. State Assembly elections in India are held every 5 years but these are not conducted in the same year in all the states. In year 2017, elections were held in Punjab on February 4 and the results were to be announced on March 11. But with sentiment analysis results can be predicted prior to actual results. In this section a real time system has been discussed that was developed during Punjab elections. This system analyzed election data from Twitter in real time and finally predicted the results very closely prior to the declaration of actual results. This system is discussed as follows.

6.1 Real Time Sentiment Analysis System

The machine learning system proposed in the previous chapter has been used for Twitter sentiment analysis of 2017 Punjab Elections. The system extracts related data from Twitter in real time, determines their sentiments and presents the analysis results back to

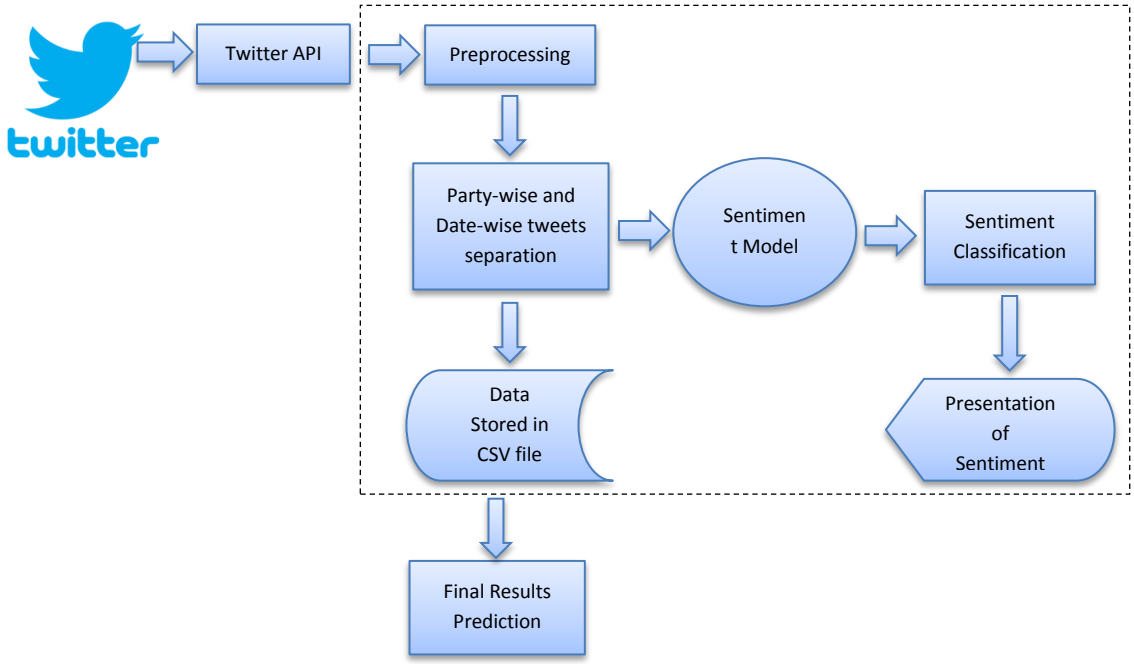

Figure. 6.1 Working of Real Time Sentiment Analysis System

user simultaneously [39] [40]. As the incoming data was to be used for final results prediction also, so side by side it is getting stored in CSV file. A user interface is designed to facilitate user interaction. Backend of the system is developed in Python and user interface in HTML using JavaScript and AJAX technologies. Though the system was developed for Punjab elections, but it can be easily adapted to other domains as well. Figure 6.1 shows the working of this system.

The very first task performed by the system is data collection. Data in this research work is collected from Twitter and Python module named *twitter* has been used for making secure and authenticated requests to Twitter API. Twitter API is Twitter's REST API that authenticates an application with *OAuth* protocol to read and write Twitter data and all responses are returned in JSON format. Before *OAuth*, username and password were required for authorizing this access which was a major security concern but with *OAuth* this access is authenticated by access tokens issued while app generation. These tokens include Consumer Key, Consumer Secret, Access Key and Access Secret.

Figure 6.2 shows the Twitter app named *alantic* being generated issued with access tokens.

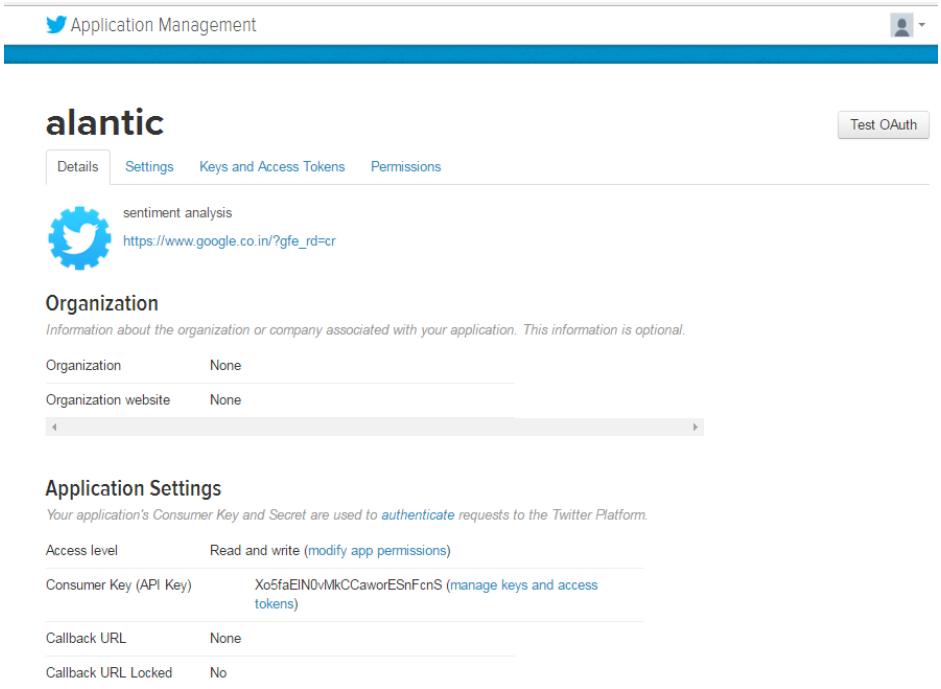

Figure. 6.2 Twitter App Generation

A channel is specified from where the data is to be collected. By default it is ‘HTPunjab’, but user can mention any channel of his choice. Data from Twitter is returned in JSON format and besides containing tweets it consists of a lot of other information also. The data coming over a particular time span needs to be preprocessed. It includes all the preprocessing techniques discussed in section 4.2.1.2.

As channels broadcast a lot of other information also, so tweets need to be filtered specific to the topic of interest. As the system is developed for Punjab elections, therefore some keywords were mentioned for collecting tweets related to Punjab elections only stripping out others. There are three political parties in Punjab namely Aam Aadmi Party (AAP), Congress and SAD-BJP ally. Tweets are collected having these parties and their leaders as mentions. Table 6.1 shows the keywords used for filtering tweets specific to Punjab elections.

Table 6.1 Table Used for Filtering Tweets

Party Name	Keywords
Aam Aadmi Party	'AamAadmiParty', 'ArvindKejriwal', 'BhagwantMann', 'AAPPunjab2017', 'AAPPunjab'.
Congress	'WithCongPunjab', 'sherryontopp', 'punjabpcc', 'capt_amarinder', 'Capt_amarinder', 'capt_Amarinder', 'INCIndia', 'incindia', 'Capt Amarinder', 'Capt.Amarinder'.
SAD-BJP	'Akali_Dal_', 'officeofssbadal', 'BJP4Punjab', 'BJP4Punjab', 'Akali Dal'.

As the problem is real time it is necessary to filter tweets by date and time also. In JSON data returned from Twitter, each tweet has an associated timestamp with field name ‘created_at’. This field is used to filter tweets by timestamps.

After preprocessing and sorting, data is completely suitable for sentiment analysis. Hereafter, it is directed towards the sentiment model to have real time insights. The model classifies each tweet as either positive, negative or neutral and simultaneously the results are displayed to user. At the same time, data is stored in CSV file separated by

three columns, *i.e.*, *Party name*, *Date* and *Text*. This data can later on be used for final election results prediction.

Figure 6.3 shows the dashboard for displaying sentiment analysis results to user. In top left corner of figure, tweets with the timestamps they were posted on and the associated sentiment class is displayed. Top right corner of figure displays the comparative sentiment analysis of three parties over a particular time span. Along with it there is a choice of channel from where tweets are to be fetched. User can mention any channel of his choice, by default it is 'HTPunjab'. The system also separates tweets' words by sentiment class corresponding to each political party and displays separate positive and negative word clouds for them. Bottom left of figure displays the positive sentiment word cloud of Aam Aadmi Party (AAP) and bottom right displays the negative word cloud of same party. Similarly, these are displayed for other two parties also.

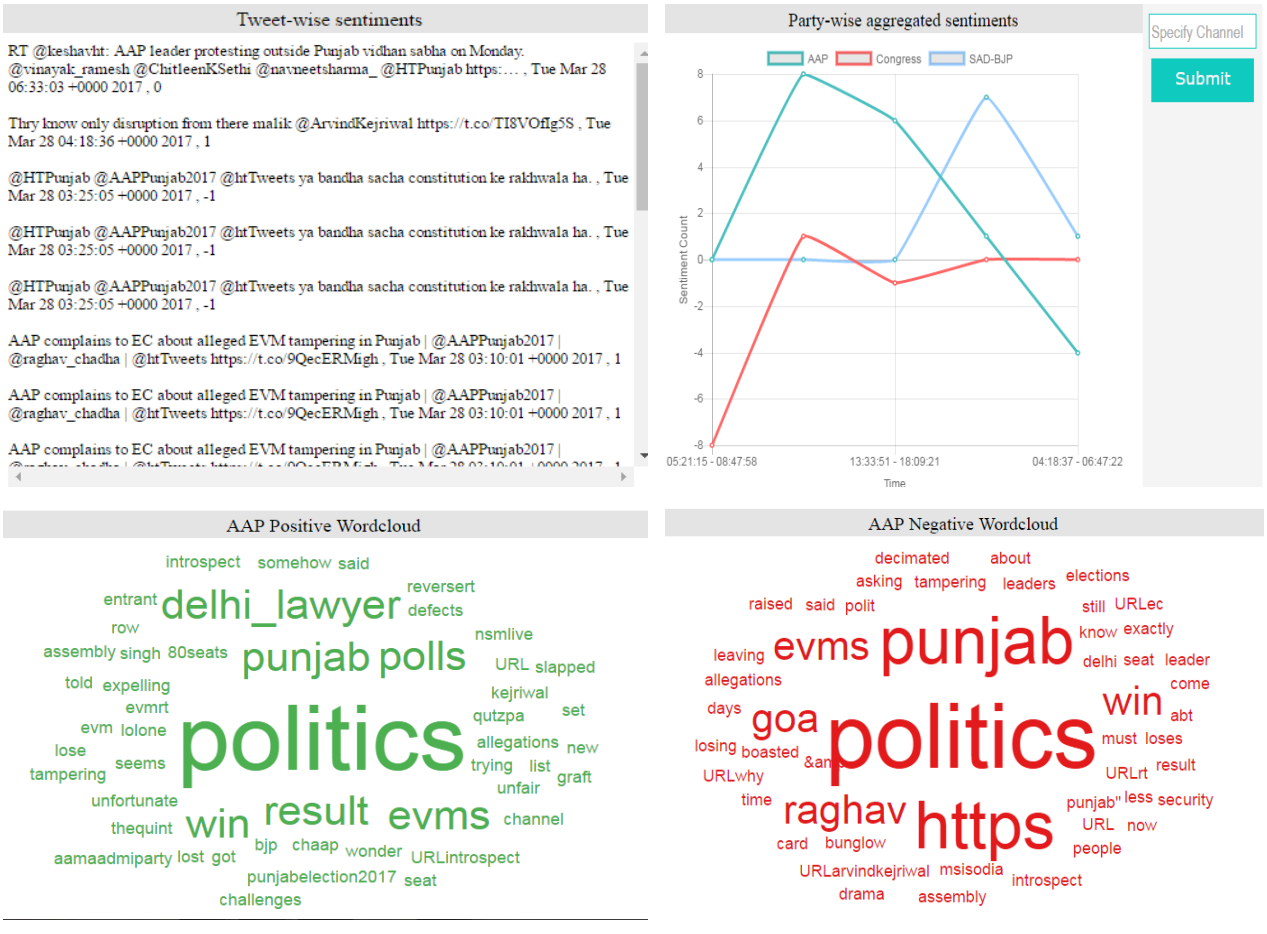

Figure. 6.3 Presentation of Results in Real Time

6.2 Final Election Results Prediction

Data that was stored in CSV format is used for final results prediction. Predictions of all the five classifiers are recorded and compared against actual election results declared on 11 March 2017. A total of 1900 tweets were collected from January 13 to February 6 2017, but some were duplicates and were removed. Finally the prediction dataset was left with 1573 tweets.

Table 6.2 shows the predictions done by all the five classifiers. Predictions are done in the form of total positive tweets for each party over 24 days period. Along with, actual results declared on 11 March 2017 are also displayed. Results display total percentage of seats won by each party.

Table 6.2 Comparison of Predictions with Actual Results

	Naïve Bayes	Multinomial Naïve Bayes	Bernoulli Naïve Bayes	Maximum Entropy	Support Vector Machines	Actual Results
Congress	73.2%	72.3%	65.8%	75%	44.7%	65.8%
AAP	20.2%	20.9%	23.7%	18.9%	37.6%	17.2%
SAD-BJP	6.6%	6.8%	10.5%	6.1%	17.7%	15.4%

Predictions of all the classifiers are almost same to the actual results except Support Vector Machines. Bernoulli Nave Bayes which is the base model for this system has predicted most accurately. Thus, the system has successfully predicted the election results with minor deviations.

Figure 6.4 displays the percentage positive and negative tweets of three parties as predicted by the classifiers.

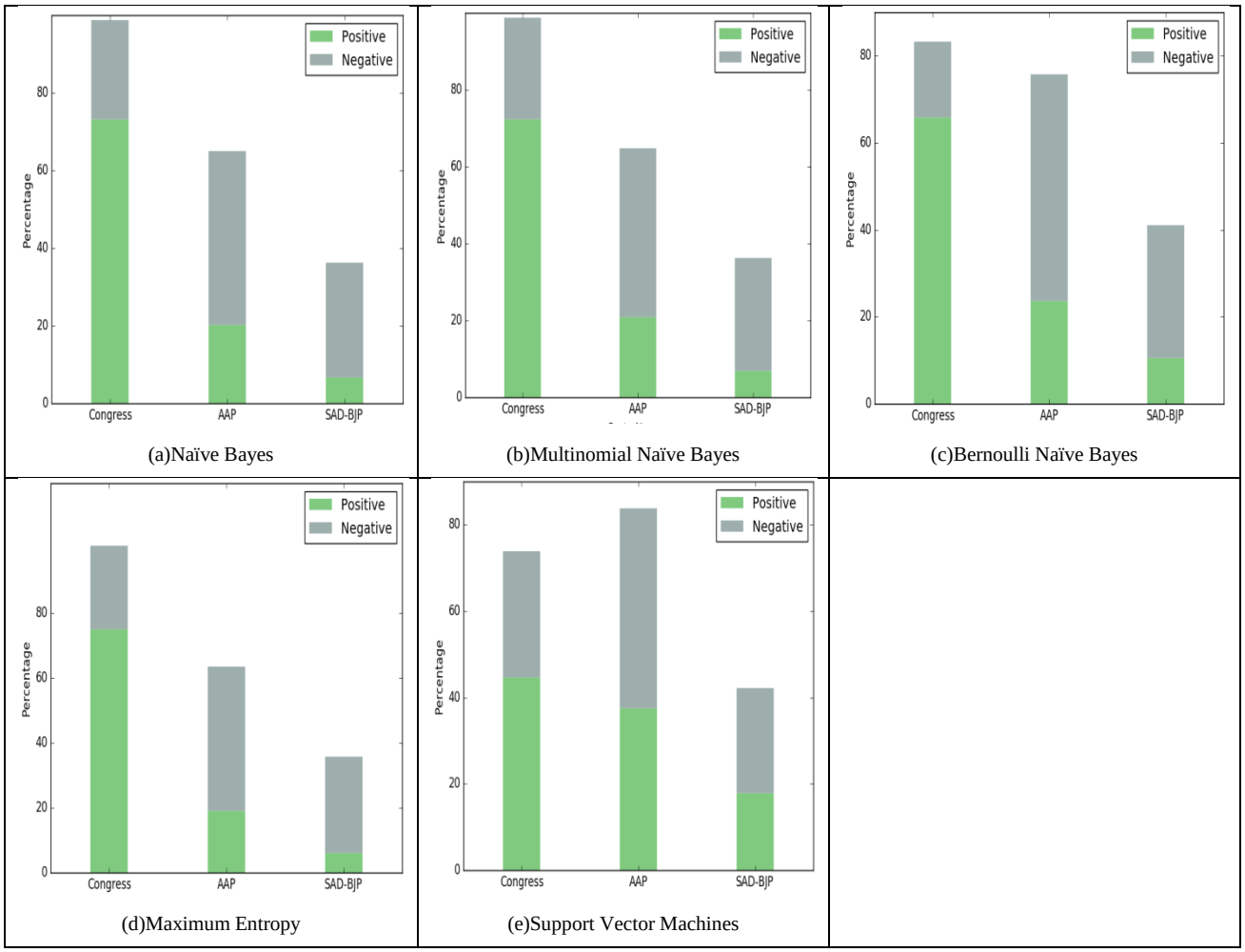

Figure. 6.4 Percentage of positive and negative tweets of each party predicted by classifiers

Chapter Summary

In this chapter, real time sentiment analysis system and it working was discussed. The complete process of how system fetches Twitter data in real time, processes it for sentiment analysis and presents results to users in real time is acknowledged. Lastly, the predictions done by the system are displayed and it is observed that system has successfully predicted election results.

Chapter 7

Conclusions and Future Scope

7.1 Conclusions

In this research work two sentiment analysis systems have been developed, one by customizing the existing tools and other by using machine learning and deep learning techniques. For building the customized system a number of online tools were surveyed and it was observed that AlchemyAPI is best of all. But due to lack of some important features a new customized system was needed. It has been developed by accommodating some features from other tool named Repustate. This customized system provides a total of 13 features for sentiment analysis. Then a number of machine learning and deep learning techniques were explored and it was found that deep learning performs better than machine learning. A number of features were tested on machine learning models and Information gain has been used for selecting best quality features. The metric-square was used for calculating information score of each feature. Using information gain, an accuracy increment of 8% has been achieved. A real time Twitter sentiment analysis system was developed using one of the best performing machine learning algorithms, *i.e.* Bernoulli Naïve Bayes. Using this real time system, election data of 2017 Punjab elections was analyzed and results were predicted with minor deviation from actual results. The system was designed for Punjab elections but it can equally work for other applications as well.

7.2 Future Scope

The system developed in this work suffers from some limitations. Some resolutions to these limitations that can be addressed in future are given below.

- The real time sentiment analysis system works only for English, but the tweets usually consist of mixed languages. So, this issue can be handled by incorporating a language identification and translation module.

- Tweets also consist of transliterated text which may hold very useful information. As transliteration is not addressed in this research, some content remains unanalyzed. So, transliteration issue can be handled by having complete knowledge of multiple languages. Using phonetics of each language this issue can be resolved.
- For current research, some issues like slang, anaphora and sarcasm remain unaddressed which if handled can impart a lot more information. So, these issues need to be addressed.
- Currently the models have been trained on a dataset of 1330 instances and tested on 550 instances. In future, dataset can be increased to get better results.
- Deep learning can be integrated with real time system to get more effective results.

References

[1] Savoy, OlenaKummer—Jacques. "Feature Selection in Sentiment Analysis."*CORIA* (2012).

[2] Khan, Aamera ZH, Mohammad Atique, and V. M. Thakare. "Combining lexicon-based and learning-based methods for Twitter sentiment analysis." *International Journal of Electronics, Communication and Soft Computing Science & Engineering (IJECSCSE)* (2015): 89.

[3] Schrauwen, Sarah. "Machine learning approaches to sentiment analysis using the Dutch Netlog Corpus." *Computational Linguistics and Psycholinguistics Research Center* (2010).

[4]Mitchell, T. "Generative and discriminative classifiers: naive Bayes and logistic regression, 2005." *Manuscript available at http://www.cs. cm. edu/~ tom/NewChapters.html* (2016).

[5] Mehra, Nipun, ShashikantKhandelwal, and Priyank Patel. *Sentiment identification using maximum entropy analysis of movie reviews*. Working paper, 2002.

[6] Text Classification and Sentiment Analysis. [Online] Available: http://ataspinar.com/2015/11/16/text-classification-and-sentiment-analysis/#ch3

[7] Pang, Bo, Lillian Lee, and ShivakumarVaithyanathan. "Thumbs up?: sentiment classification using machine learning techniques." *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*.Association for Computational Linguistics, 2002.

[8] Hiroshi, Kanayama, Nasukawa Tetsuya, and Watanabe Hideo. "Deeper sentiment analysis using machine translation technology." *Proceedings of the 20th international conference on Computational Linguistics*.Association for Computational Linguistics, 2004.

[9]Boiy, Erik, and Marie-Francine Moens. "A machine learning approach to sentiment analysis in multilingual Web texts." *Information retrieval* 12.5 (2009): 526-558.

[10] Schrauwen, Sarah. "Machine learning approaches to sentiment analysis using the Dutch Netlog Corpus." *Computational Linguistics and Psycholinguistics Research Center* (2010).

[11] Almatrafi, Omaima, SuhemParack, and BravimChavan. "Application of location-based sentiment analysis using Twitter for identifying trends towards Indian general elections 2014." *Proceedings of the 9th International Conference on Ubiquitous Information Management and Communication*.ACM, 2015.

[12] Amolik, Akshay, et al. "Twitter Sentiment Analysis of Movie Reviews using Machine Learning Techniques." *International Journal of Engineering and Technology* 7.6 (2015): 2038-2044.

[13] Wawre, Suchita V., and Sachin N. Deshmukh. "Sentiment Classification using Machine Learning Techniques."*Indian Journal of Science and Research (IJSR)*.

[14] Sahayak, Varsha, VijayaShete, and ApashabiPathan. "Sentiment Analysis on Twitter Data." *International Journal of Innovative Research in Advanced Engineering (IJIRAE)* 2.1 (2015): 178-183.

[15] Zhang, Xueying, and XianghanZheng. "Comparison of Text Sentiment Analysis Based on Machine Learning."*Parallel and Distributed Computing (ISPDC), 2016 15th International Symposium on*. IEEE, 2016.

[16] Kiran, T. D. V., K. Gowtham Reddy, and JagadeeshGopal. "Twitter sentiment analysis of game reviews using machine learning techniques." *Journal of Chemical and Pharmaceutical Sciences.*

[17] Shirani-Mehr, Houshmand. "Applications of Deep Learning to Sentiment Analysis of Movie Reviews.". [Online] Available: http://cs224d.stanford.edu/reports.html[July 2015]

[18] Stojanovski, Dario, et al. "Twitter sentiment analysis using deep convolutional neural network." *International Conference on Hybrid Artificial Intelligence Systems*.Springer International Publishing, 2015.

[19] Hassan, Abdalraouf, and AusifMahmood. "Deep Learning approach for sentiment analysis of short texts."*Control, Automation and Robotics (ICCAR), 2017 3rd - International Conference on*.IEEE, 2017.

[20] 'Alchemy API'. [Online]. Available: http://www.alchemyapi.com/.

[21]'Alchemy Language'. [Online]. Available: https://www.ibm.com/watson/developercloud/doc/alchemylanguage/.

[22] 'Repustate'. [Online]. Available: https://www.repustate.com/docs.

[23] Semantria API. [Online]. Available: https://www.lexalytics.com/semantria.

[24] Text Analytics. [Online]. Available: https://www.lexalytics.com/technology/text-analytics.

[25] Araújo, Matheus, et al. "iFeel: a system that compares and combines sentiment analysis methods." *Proceedings of the 23rd International Conference on World Wide Web*.ACM, 2014.

[26] P. Gonçalves, M. Araújo, F. Benevenuto, and M. Cha. Comparing and combining sentiment analysis methods. In COSN, 2013.

[27] P. Goncalves, F. Benevenuto, and M. Cha.Panas-t: A psychometric scale for measuring sentiments on twitter. CoRR, 2013.

[28] D. Watson and L. Clark. Development and validation of brief measures of positive and negative affect: the panas scales. Journal of Personality and Social Psychology, 54(1):1063–1070, 1985.

[29] Esuli and Sebastiani. Sentiwordnet: A publicly available lexical resource for opinion mining. In LREC, 2006.

[30] P. S. Dodds and C. M. Danforth.Measuring the happiness of large-scale written expression: songs, blogs, and presidents. Journal of Happiness Studies, 11(4):441–456, 2009.

[31] E. Cambria, R. Speer, C. Havasi, and A. Hussain.Senticnet: A publicly available semantic resource for opinion mining. In AAAI Fall Symposium Series, 2010.

[32] H. Wang, D. Can, A. Kazemzadeh, F. Bar, and S. Narayanan. A system for real-time Twitter sentiment analysis of 2012 U.S. presidential election cycle. In ACL, pages 115–120, 2012.

[33] E. Cambria, A. Hussain, C. Havasi, C. Eckl, and J. Munro. Towards crowd validation of the uk national health service. In ACM WebSci, 2010.

[34] E. Cambria, A. Livingstone, and A. Hussain.The hourglass of emotions. Lecture Notes in Computer Science. Springer, 2011.

[35] M. M. Bradley and P. J. Lang. Affective norms for English words (ANEW): Stimuli, instruction manual, and affective ratings. Technical report, University of Florida, 1999.

[36] Y. R. Tausczik and J. W. Pennebaker. The psychological meaning of words: Liwc and computerized text analysis methods. Journal of Language and Social Psychology, 29(1):24–54, 2010.

[37] Kim, Jaewoo, Meeyoung Cha, and Thomas Sandholm. "SocRoutes: safe routes based on tweet sentiments." *Proceedings of the 23rd International Conference on World Wide Web*.ACM, 2014.

[38] Forman, George. "An extensive empirical study of feature selection metrics for text classification." Journal of machine learning research 3.Mar (2003): 1289-1305.

[39] Sharma, Parul, and Teng-Sheng Moh. "Prediction of Indian election using sentiment analysis on Hindi Twitter." Big Data (Big Data), 2016 IEEE International Conference on.IEEE, 2016.

[40] Wang, Hao, et al. "A system for real-time twitter sentiment analysis of 2012 us presidential election cycle." Proceedings of the ACL 2012 System Demonstrations.Association for Computational Linguistics, 2012.

Research Publications

Research Paper Published

Yachika Gupta and Parteek Kumar, "Customized Automated Sentiment Analysis System" in *International Journal of Information, Communication and Computing Technology (ICICCT-2017)*, JIMS, New Delhi, India.

Research Paper Accepted

Yachika Gupta and Parteek Kumar, "Real Time Sentiment Analysis System and Prediction of Punjab Elections 2017" in *International conference on Computing, Analytics & Networks* (*ICAN2017),* Chitkara University, Himachal Pradesh, India. (Selected as one of the best papers in *National Symposium on Computing Analytics & Networks (NCAN2017)*, Chitkara University, Himachal Pradesh.)

Video URL

https://youtu.be/0alZirxdJII

dss

ORIGINALITY REPORT

%9	%5	%5	%4
SIMILARITY INDEX	INTERNET SOURCES	PUBLICATIONS	STUDENT PAPERS

PRIMARY SOURCES

1	www.lexalytics.com Internet Source	%1
2	www.repustate.com Internet Source	<%1
3	knowledgecommons.lakeheadu.ca Internet Source	<%1
4	Studies in Computational Intelligence, 2016. Publication	<%1
5	Lecture Notes in Computer Science, 2015. Publication	<%1
6	Submitted to Thapar University, Patiala Student Paper	<%1
7	Submitted to CSU, San Jose State University Student Paper	<%1
8	Chen, Yi-Shin, Yi-Cheng Peng, Jheng-He Liang, Elvis Saravia, Fernando Calderon, Chung-Hao Chang, Ya-Ting Chuang, Tzu-Lung Chen, and Elizabeth Kwan. "Concept-based event	<%1