Rule Based Sentiment Analysis System

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

Master of Engineering

in

Computer Science and Engineering

Submitted By

Sujata Rani

(801232029)

Under the supervision of:

Dr. Parteek Kumar

Assistant Professor

COMPUTER SCIENCE AND ENGINEERING DEPARTMENT

THAPAR UNIVERSITY

PATIALA – 147004

July 2014

Certificate

I hereby certify that the work which is being presented in the thesis entitled, "***Rule Based Sentiment Analysis System***", in partial fulfillment of the requirements for the award of 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 researcher's 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.

Sujata Rani

(Sujata Rani)

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

Parteek Kumar

(Dr. Parteek Kumar)
Assistant Professor,
Computer Science and Engineering Department

Countersigned by

(Dr. Deepak Garg)
Head
Computer Science and Engineering Department
Thapar University
Patiala

(Dr. S. K. Mohapatra)
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**, Assistant Professor, Computer Science and Engineering Department, Thapar University for his positive attitude, excellent guidance, constant encouragement, keen interest, invaluable co-operation, generous attitude and above all his blessings. He has been a source of inspiration for me.

I am grateful to **Dr. Deepak Garg**, 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.

(Sujata Rani)

Roll No. 801232029

Abstract

Due to the exponential enhancement in the Internet usage and replacement of public opinions, sentiment analysis becomes an important process in today's life. Sentiment analysis is a process of extracting information from user's opinions. Every person shares his or her information in social network sites, blogs, product review websites and web-forums. Thus, the thoughts of other people provide information that helps in decision making process. But, sentiment analysis is a challenging task because it is very difficult to find the exact sentiment from text as there are so many challenges like entity identification, subjectivity detection in performing sentiment analysis.

The project work that has been carried out in the thesis is focused on implementation of rule based sentiment analysis system. Earlier, only adjectives were used as feature to perform sentiment analysis. But, this system uses adverb-adjective combinations as a feature as it improves the sentiment analysis process. The system extracts the twitter posts and computes the frequency of each word in tweet. Then, it calculates the sentiment and score of each tweet. The system also computes the sentiment and score of simple English sentences. Some of the challenges like thwarted expectations, pragmatics, and entity identification have been resolved by system. The system has also used UNL (Universal Networking Language) to resolve the challenges like entity identification *etc.* At the end, the results of simple English sentences given by the system have been compared with manual testing of these sentences.

Table of Contents

Certificate……... i

Acknowledgement…….. ii

Abstract……... iii

Table of Contents……... iv

List of Figures……. vii

List of Tables…….. ix

List of Algorithms…….. x

Chapter 1: Introduction……..... 1-11

1.1 Introduction to Sentiment Analysis (SA)……... 1

1.2 Applications of Sentiment Analysis……... 2

1.2.1 Online Commerce……. 2

1.2.2 Voice of the Market (VOM)……. 3

1.2.3 Voice of the Customer (VOC)…….. 3

1.2.4 Brand Reputation Management…… 3

1.2.5 Government……... 3

1.3 Challenges of Sentiment Analysis……. 4

1.3.1 Implicit Sentiment and Sarcasm……... 4

1.3.2 Domain Dependency……. 5

1.3.3 Thwarted Expectations…….. 5

1.3.4 Pragmatics……. 6

1.3.5 World Knowledge……. 7

1.3.6 Subjectivity Detection……... 7

1.3.7 Entity Identification…….. 8

1.3.8 Negation……..... 9

1.4 Thesis Outline…… 11

Chapter 2: Literature Review……... 12-31

2.1 Approaches for Sentiment Analysis…… 12

2.1.1 Keyword-based approach…… 12

2.1.2 Concept-based approaches…… 13
2.1.3 Lexical Affinity……. 13
2.1.4 Discourse Structures……. 14
2.2 Techniques for Sentiment Analysis……... 14
2.2.1 Machine Learning Techniques…….. 14
2.2.1.1 Supervised Techniques…… 14
2.2.1.2 Unsupervised Techniques…… 19
2.2.2 Feature Extraction……. 20
2.2.2.1 Term Presence or Term Frequency…….. 20
2.2.2.2 Opinion Words or Phrases……... 21
2.2.2.3 POS (Parts of Speech) Tags……..... 21
2.2.2.4 Syntax and Negation…… 21
2.2.3 Rule Based Technique…….. 21
2.3 Role of UNL (Universal Networking Language) in Knowledge Extraction…. 22
2.3.1 Introduction to UNL…….. 24
2.3.2 UNL Format for Information Representation…… 25
2.3.2.1 Universal Words……... 25
2.3.2.2 UNL Relations…….. 27
2.3.2.3 UNL Expression Hyper-graph…….. 28
2.3.2.4 UNL Attributes……. 29
2.3.3 UNL Sentence…… 30
Chapter 3: Problem Statement…… 32-33
3.1 Objectives……... 32
3.2 Methodology…….. 32
Chapter 4: Implementation……... 34-63
4.1 Architecture of Proposed Rule Based Sentiment Analysis System……... 34
4.1.1 Tokenizer…….. 34
4.1.2 Rule set for AAC (Adverb-Adjective Combination)…… 35
4.1.2.1 Adjectives …… 35
4.1.2.2 Types of Adverbs …… 36
4.1.2.3 Types of AACs……. 37

4.1.2.4 Rules for AACs…… 37
4.1.3 Computation of Scores for AACs…… 40
4.1.3.1 Variable Scoring Algorithm (VSC)…… 40
4.1.3.2 Adjective Priority Scoring Algorithm (APS)……..... 42
4.1.3.3 Adverb Priority Scoring Algorithm (AdvPS)……..... 44
4.1.3.4 Resulting Sentiment Computing Algorithm…… 45
4.2 Process for Calculation of Final Sentiment and Score……... 46
4.3 Role of Python in Sentiment Analysis……... 48
4.4 Sentiment Analysis of Twitter Posts…….. 49
4.4.1 Extraction of Twitter Posts……... 49
4.4.2 Role of Score File…… 52
4.4.3 Computation of Term Frequency…….. 52
4.4.4 Sentiment Analysis of Tweets……... 53
4.5 Sentiment Analysis of English Sentences…….. 55
4.6 Sentiment Analysis of Natural Text using UNL…… 55
4.7 Handling of Challenges of Sentiment Analysis…… 59
4.7.1 Domain Dependency…… 59
4.7.2 Thwarted Expectations…… 59
4.7.3 Pragmatics…… 61
4.7.4 World Knowledge…… 61
4.7.5 Entity Identification…… 62
Chapter 5: Results and Discussions…….. 64-72
5.1 Testing of Tweets……... 64
5.2 Testing of Simple English Sentences…… 65
5.3 Testing of English Sentences of UC-A1 Corpus……... 67
5.4 Comparison of Results of Proposed System with Manual Testing…… 72
Chapter 6: Conclusion and Future Scope……..... 73-74
6.1 Conclusion…… 73
6.2 Limitations and Future Scope…… 74
References……... 75-78
List of Publications…… 79

List of Figures

Fig. 2.1: Demonstration of Naïve Bayes Classifier……... 15

Fig. 2.2: Classification of New Object in NBCs…… 16

Fig. 2.3(a): Example of Linear SVM……. 17

Fig. 2.3(b): Example of Hyperplane SVM……. 17

Fig. 2.4: Mapping of Objects in SVMs……..... 18

Fig. 2.5: A UNL System…… 24

Fig. 2.6: UNL graph of Example Sentence given in (2.19)……..... 31

Fig 4.1: Proposed Architecture of Rule Based Sentiment Analysis System…….. 34

Fig 4.2: Process for Computing Score and Sentiment……... 47

Fig 4.3: GUI of Sentiment Analysis System…….. 50

Fig 4.4: Clicking on "Extract Post" Button…… 50

Fig 4.5: Enter Parameter and File Name…… 51

Fig 4.6: Generated Tweet File…… 51

Fig 4.7: Clicking on "Calculate Frequency" Button……..... 53

Fig 4.8: Select Tweet File and Get Term Frequency……..... 53

Fig 4.9: Clicking on "Calculate Score" Button…….. 54

Fig 4.10: Score and Sentiment of each Tweet……... 54

Fig 4.11: Example of English Sentence given in (4.28)……..... 55

Fig 4.12: Sentiment and Score of Sentence given in (4.28)……... 55

Fig 4.13: Example of English Sentence given in (4.29)……..... 57

Fig 4.14: Output of English Sentence given in (4.29)……... 57

Fig 4.15: Example of Punjabi Sentence given in (4.31)……..... 58

Fig 4.16: Output of Punjabi Sentence given in (4.31)……... 58

Fig. 4.17: Results after handling "Domain Dependency"…….. 59

Fig. 4.18 (a): Results after handling "Thwarted Expectations" with Positive Sentiment……... 60

Fig. 4.18 (b): Results after handling "Thwarted Expectations" with Negative Sentiment……... 60

Fig. 4.19: Results after handling “Pragmatics”…….. 61
Fig. 4.20: Results after handling “World Knowledge”…….. 62
Fig. 4.21: Results after handling “Entity Identification……..... 62
Fig 5.1: Results of Tweets about “Arvind Kejriwal”……. 64

List of Tables

Table 2.1: Systems using UNL for Knowledge Extraction……... 22
Table 2.2: Syntax of UW……... 25
Table 2.3: Example of Restricted UW……... 26
Table 2.4: Examples of Extra UWs……... 26
Table 2.5: Syntax of UNL Relation……... 27
Table 2.6: Types of Relations in UNL……... 27
Table 2.7: Syntax of UNL Expression……... 28
Table 2.8: Types of UNL Attributes……..... 29
Table 2.9: Syntax of UNL Sentence in Table Format…… 30
Table 2.10: Example of UNL Sentence given in (2.19)…… 30
Table 4.1: Positive and Negative Adjectives……..... 35
Table 4.2: Types of adverbs of degree……... 36
Table 4.3: Score File……..... 52
Table 4.4: Tokens of UNL given in (4.29)…… 57
Table 5.1: Sentiment and Score of Simple English Sentences given by Proposed System……..... 65
Table 5.2: Parts of Speech contained by UC-A1 Corpus……..... 67
Table 5.3: Testing of UC-A1 Corpus…… 68
Table 5.4: Comparison of Results of Proposed System with Manual Testing…….. 72

List of Algorithms

Algorithm 4.1: Variable Scoring Algorithm…….. 41
Algorithm 4.2: Adjective Priority Scoring Algorithm……... 42
Algorithm 4.3: Adverb Priority Scoring Algorithm……... 44
Algorithm 4.4: Resulting Sentiment Computing Algorithm…….. 46
Algorithm 4.5: Computation of Sentiment and Score using UNL……..... 56

Chapter 1
Introduction

1.1 Introduction to Sentiment Analysis (SA)

Sentiment analysis is an information gathering task to attain user's feelings. By analyzing a large numbers of documents, these feelings can be expressed in positive or negative ways in the form of comments, questions and requests. Generally, sentiment analysis helps to find the attitude of a writer about any topic or the overall sentiment of a document or text. Due to the exponential enhancement in the Internet usage and replacement of public opinions, sentiment analysis becomes an important process in today's life. The Web is a huge depository of ordered and amorphous data. The analysis of this data to extract hidden public opinions and sentiment is not an easy task.

Sentiment analysis can be done at three levels that are document level, phrase level and sentence level. In document level, the entire document is summarized to check whether the sentiment about the document is positive, negative or objective. In Sentence level, analysis about the individual sentiment bearing sentences is classified. In phrase level, analysis of phrases in a sentence is carried out according to polarity [1].

SA finds the phrases in a text or document that contains some sentiment. There may be some objective facts or subjective opinions in the text. It is compulsory to distinguish between them. SA helps in determining the entities and subject from text towards which sentiment is directed. Sentiments are categorized as objective (facts), positive (represents a state of gladness, happiness, pleasure or satisfaction) or negative (represents a state of sorrow, regret, sadness or disappointment). On the basis of degree of polarity, a score can be given to the sentiments [1]. Thus, there are two research tips; first classifying the polarity of text to express the opinion in positive, negative or neutral. Second is identification of subjectivity or objectivity [2, 3]. Due to its many aspects it is often referred to with different names such as opinion mining, sentiment classification, sentiment analysis, and sentiment extraction [4].

The thoughts of other people provide information that helps in decision making process [5]. Every person shares his or her information in social network sites, blogs, product

review websites and web-forums. From all the social network sites like Twitter and Facebook, e-commerce sites, blogs; reviews and opinions of users can be found. All these reviews and opinions help the people and business organizations. These reviews and opinions also help the researchers that how to analyze and summarize the opinions expressed in this enormous amount of text data.

1.2 Applications of Sentiment Analysis

Word of Mouth (WOM) is the way of giving information from one person to another person. It helps the customers in taking decisions. Word of Mouth gives the information about the reactions, opinions or attitudes of consumers about products, business or services that they share with other persons. It imparts the information on the basis of social networking and trust. Most of the people in their social network depend on families, friends, and others. It is also indicated by research that people believe on the opinions of other people quickly that exist outside their social networks such as online reviews. Hence, this is where SA comes into play. As the online review sites, blogs, social networking sites provide huge amount of opinions, this helps in making decision-making process easier for us [6]. Some of the applications of sentiment analysis are discussed as follows.

1.2.1 Online Commerce

Most of the e-commerce activities use sentiment analysis. All the websites allow users to give their feedback about shopping and quality of products. They give the information about different features and summary of product by assigning scores and ratings to products. Then, the users can view reviews, recommendation information and opinions about products and their special features. A summary of product and its features is presented to users in graphical form. Most of the popular commercial websites like amazon.com give the information about reviews from customers with rating information and editors. By analyzing these large volumes of opinions, SA helps the websites by changing dissatisfied customers into promoters [6].

1.2.2 Voice of the Market (VOM)

Voice of the Market helps in determining the feelings of customers about products or services of competitors. VOM helps in gaining advantage at competitive part and development of new product from this accurate and timely information. To direct and target key marketing campaigns, detection of such information is required as early as possible. Sentiment Analysis helps company or group to get opinions or reviews of customers in real-time. This real-time information helps in designing new marketing strategies, to predict chances of product failure and to improve product features [6].

1.2.3 Voice of the Customer (VOC)

Voice of the Customer helps in determining what the individual customer or user is saying about products or services. It means examining the reviews, opinions and feedback of the customers. VOC is an important part of Customer Experience Management. To invent new products, VOC helps in finding such new opportunities. Extracting customer opinions also helps identify functional requirements of the products and some non-functional requirements like performance and cost [6].

1.2.4 Brand Reputation Management (BRM)

Brand Reputation Management is concerned about management of reputation in market. It focuses on product and company rather than customer. Now, one-to-many conversations are taking place online at a high rate. So, opportunities are created for organizations to manage and strengthen their brand reputation. Not only advertising, but also public relations and corporate messaging helps in determining brand perception. Brands are now a sum of the conversations about them. Sentiment analysis helps in determining how company's brand, product or service is being perceived by community online [6].

1.2.5 Government

Sentiment analysis helps government in assessing their strength and weaknesses by analyzing opinions from public. For example, "If there is situation like, The MP who is investigating 2G scam himself is deeply corrupt then how do you expect that truth will

come out?." This example shows the negative sentiment about government. Sentiment analysis also helps in identifying strengths and weaknesses in a recruitment campaign in government job, assessing success of electronic submission of tax returns and many other areas [6].

1.3 Challenges of Sentiment Analysis

Sentiment analysis approaches try to extract words from text that have positive and negative sentiment. If any sentiment bearing word is not present in the text then classify the text as objective. In this way, a text categorization task can be done. In text classification, different topics consist of many classes but there are only three broad classes in sentiment analysis. Even then, sentiment analysis is not easier than text classification [1]. The general challenges of sentiment analysis can be summarized as follows.

1.3.1 Implicit Sentiment and Sarcasm

Without the presence of any sentiment bearing words, sentence may have an implicit sentiment [1]. English sentences to illustrate the concept of implicit sentiment and sarcasm are given in (1.1) and (1.2). Punjabi sentence to demonstrate this concept is given in (1.3) while its corresponding English translation is given in (1.4).

How can anyone sit during this presentation? … (1.1)

One should question the strength of mind of the writer who had written this book. … (1.2)

Punjabi Sentence: *ਤੁਸੀਂ ਇਹ ਕੰਮ ਕਿਵੇਂ ਕਰ ਸਕਦੇ ਹੋ?* … (1.3)

Transliterated Sentence: *tusīṃ ih kamm kivēṃ kar sakadē hō?*

Equivalent English Sentence: *How can you do this work?* … (1.4)

Here, all of the above sentences given in (1.1), (1.2) and (1.3) do not have any word that has negative sentiment but all are negative sentences. Thus, identifying the semantics is very important in semantic analysis than syntax detection.

1.3.2 Domain Dependency

There is always a change in polarity of words from domain to domain [1]. English sentences to illustrate the concept of domain dependency are given in (1.5) and (1.6). Punjabi sentence to demonstrate this concept is given in (1.7) while its corresponding English translation is given in (1.8).

The story of the movie was unpredictable. … (1.5)

The steering of the bus is unpredictable. … (1.6)

Punjabi Sentence: *ਤੁਸੀਂ ਕਿੱਥੇ ਗਏ ਸੀ?* … (1.7)

Transliterated Sentence*: tusīṃ kitthē gaē sī?*

Equivalent English Sentence: *Where you had gone?* … (1.8)

In the sentence given in (1.5), the sentiment depicted is positive but the sentiment depicted in the sentence given in (1.6) is negative. In the sentence given in (1.7), sentiment conveyed is positive if someone is simply asking where you have gone but a negative sentiment when you were late to your home and your mother or father is asking in anger that where had you gone.

1.3.3 Thwarted Expectations

Sometimes the writer intentionally sets up situation only to disprove it at the end [1]. English text to illustrate the concept of thwarted expectations is given in (1.9). Similarly, Punjabi text to demonstrate this concept is given in (1.10) while its corresponding English translation is given in (1.11).

This film should be awesome. There is too much fun, the actors are first class, the supporting cast is fine as well and Hrithik Roshan is attempting to deliver an excellent performance. Still, it can't hold up. … (1.9)

As there is presence of positive words but on the whole sentiment is negative because of the last sentence. But, in earlier text classification, the overall sentiment can be positive as term frequency is very important than term presence.

Punjabi Text*: ਰਾਮ ਇੱਕ ਪੜ੍ਹਿਆ-ਲਿਖਿਆ ਅਤੇ ਹੁਸ਼ਿਆਰ ਮੁੰਡਾ ਹੈ। ਉਹ ਬਹੁਤ ਸੋਹਣਾ ਅਤੇ ਇਮਾਨਦਾਰ ਵੀ ਹੈ। ਪਰ ਉਸਦਾ ਸੁਭਾਅ ਚੰਗਾ ਨਾ ਹੋਣ ਕਰਕੇ ਉਸਨੂੰ ਕੋਈ ਪਸੰਦ ਨਹੀਂ ਕਰਦਾ।* … (1.10)

Transliterated Punjabi Text*: rām ikk paṛiā-likhiā atē hushiār muṇḍā hai. uh bahut sōhṇā atē imāndār vī hai. par usdā subhāa caṅgā nā hōṇ karkē usnū kōī pasand nahīṃ karadā.*
Equivalent English Text*: Ram is an educated and intelligent boy. He is also handsome and honest. But nobody likes him because of his nature.* … (1.11)

Here, in spite of presence of positive words but overall sentiment about '**ਰਾਮ**' rām 'Ram' is negative because the last sentence is negative.

1.3.4 Pragmatics

As the pragmatics of user opinion change the sentiment thoroughly, so it is important to detect them [1]. English sentences to illustrate the concept of pragmatics are given in (1.12) and (1.13). Similarly, Punjabi sentences to demonstrate this concept are given in (1.14) and (1.16) while their corresponding English translations are given in (1.15) and (1.17) respectively.

I have just completed watching Barca DESTROY Ac Milan. ... (1.12)
That final totally destroyed me. … (1.13)

Here, capitalization in the sentence given in (1.12) is used with delicacy to denote sentiment. This sentence presents a positive sentiment but the second sentence given in (1.13) represents a negative sentiment. There are many other methods of expressing pragmatism.

Punjabi Sentence*: ਮੈਂ ਕੱਲ "ਵਾਤਾਵਰਨ ਦਾ ਪ੍ਰਦੂਸ਼ਣ" ਕਿਤਾਬ ਖਰੀਦੀ ।* … (1.14)

Transliterated Punjabi Sentence*: maiṃ kall "vātāvran dā pradūshṇa" kitāb kharīdī .*
Equivalent English Sentence*: Yesterday I purchased a book "Vatawarn Da Pardooshan".* ... (1.15)

Punjabi Sentence*: ਵਾਤਾਵਰਨ ਦਾ ਪ੍ਰਦੂਸ਼ਣ ਦਿਨੋਂ ਦਿਨ ਵੱਧਦਾ ਜਾ ਰਿਹਾ ਹੈ ।* … (1.16)

Transliterated Punjabi Sentence*: vātāvran dā pradūshaṇ dinōṃ din vaddhdā jā rihā hai.*
Equivalent English Sentence*: Environmental pollution is increasing day by day.*… (1.17)

In sentence given in (1.14), double quotes can be can be used to denote sentiment. This sentence denotes a positive sentiment whereas the second sentence given in (1.16) denotes a negative sentiment.

1.3.5 World Knowledge

Also, world knowledge needs to be integrated in the system for determining sentiments [1]. English sentences to illustrate the concept of world knowledge are given in (1.18) and (1.19). Similarly, Punjabi sentence to demonstrate this concept is given in (1.20) while its corresponding English translation is given in (1.21).

He is a Frankenstein. … (1.18)

He has just completed Doctor Zhivago for the first time and overall conclusion is that Russia sucks. … (1.19)

Punjabi Sentence*: ਉਸਨੇ ਬਲਵੰਤ ਗਾਰਗੀ ਦੀ ਜੂਠੀ ਰੋਟੀ ਖਤਮ ਕਰ ਦਿੱਤੀ ।* … (1.20)

Transliterated Punjabi Sentence*: usnē balvant gārgī dī jūṭhī rōṭī khatam kar dittī.*
Equivalent English Sentence*: He has finished Jhuthi Roti of Balwant Gargi.* … (1.21)

The sentence given in (1.18) presents a negative sentiment but the sentence given in (1.19) presents a positive sentiment. Similarly, the sentence given in (1.20) depicts a negative sentiment. But, one has to be familiar about Frankenstein, Doctor Zhivago and ‘ਜੂਠੀ ਰੋਟੀ’ *jūṭhī rōṭ ī*’ ‘Jhuthi Roti’ to find out the sentiment.

1.3.6 Subjectivity Detection

Subjectivity Detection is used to make a distinction between opinionated and non-opinionated text. A subjectivity detection module can be included in the system to find out the objective facts. Thus, performance of a system can be increased. But, this is often

difficult to do [1]. English sentences to illustrate the concept of subjectivity detection are given in (1.23) and (1.24). Similarly, Punjabi sentences to demonstrate this concept are given in (1.25) and (1.27) while their corresponding English translations are given in (1.26) and (1.28) respectively.

I hate romantic stories and poems. … (1.22)

I do not like the movies "I hate stories". … (1.23)

Here, the sentence given in (1.22) presents an objective fact but the sentence given in (1.23) presents the opinion about a particular movie.

Punjabi Sentence*: ਮੈਂ ਰੋਜ਼ ਸੈਰ ਕਰਨ ਜਾਂਦਾ ਹਾਂ ।* … (1.24)

Transliterated Punjabi Sentence*: maiṃ rōja sair karan jāndā hāṃ.*

Equivalent English Sentence*: I go for a walk daily.* … (1.25)

Punjabi Sentence*: ਮੈਂ ਰੋਜ਼ ਸੈਰ ਕਰਨ ਜਾਂਦਾ ਹਾਂ ਕਿਉਂਕਿ ਸੈਰ ਕਰਨ ਨਾਲ ਸਿਹਤ ਚੰਗੀ ਰਹਿੰਦੀ ਹੈ ।*

… (1.26)

Transliterated Punjabi Sentence*: maiṃ rōja sair karan jāndā hāṃ kiuṅki sair karan nāl sihat caṅgī rahindī hai.*

Equivalent English Sentence*: I go for a walk daily because walking is good for health.*

… (1.27)

Here, the sentence given in (1.24) presents an objective fact whereas the sentence given in (1.26) depicts the opinion about habit of exercise.

1.3.7 Entity Identification

There may be multiple entities in a text or a sentence. It is very important to determine the entity towards which the sentiment is directed. English sentences to illustrate the concept of entity identification are given in (1.28) and (1.29). Similarly, Punjabi sentence to demonstrate this concept is given in (1.30) while its corresponding English translation is given in (1.31).

Blackberry is better than Micromax. … (1.28)

Raman defeated Hitansh in cricket. … (1.29)

Punjabi Sentence*: ਰਾਜ ਸ਼ਾਮ ਤੋਂ ਵਧੀਆ ਗਾਣਾ ਗਾਉਂਦਾ ਹੈ ।* … (1.30)

Transliterated Punjabi Sentence*: rāj shām tōṃ vadhīā gāṇā gāundā hai.*

English Sentence*: Raj sings better than Sham.* … (1.31)

The sentences given in (1.28), (1.29) and (1.30) are positive for 'Blackberry', 'Raman' and 'ਰਾਜ' *rāj 'Raj'* respectively but negative for 'Micromax*'*, 'Hitansh' and 'ਸ਼ਾਮ' *shām 'Sham'*.

1.3.8 Negation

Handling of negation is a challenging task in SA. Without the use of any negative word, negation can be expressed in many ways [1]. English sentence to illustrate the concept of negation is given in (1.32).

I do not like the movie. … (1.32)

In this sentence, polarity of all the words that appear after the negation operator (such as not) is changed. But, this method does not work for the sentence given in (1.33).

I do not like the acting but I like the direction. … (1.33)

In this sentence, scope of negation should be considered which is only up to *but* here. To solve this, polarity of all the words is changed that appear after a negation word until another negation word comes. Even then, there are problems. Consider the sentence given in (1.34).

Not only did I like the acting, but also the direction. … (1.34)

In this sentence, due to the presence of "only*"*, polarity is not reversed after "not". So, these type of combinations of "not" with the words like "only" has to be considered during designing the algorithm. Handling of negation in Punjabi sentences is more difficult than handling of negation in English sentences. Punjabi sentence to demonstrate this concept is given in (1.35) while its corresponding English translation is given in (1.36).

Punjabi Sentence*: ਮੈਨੂੰ ਖੇਡਾਂ ਖੇਡਣਾ ਪਸੰਦ ਨਹੀਂ ਹੈ।* ... (1.35)

Transliterated Punjabi Sentence*: mainūṃ krikaṭ khēḍṇā pasand nahīṃ hai.*
Equivalent English Sentence*: I don't like playing games.* … (1.36)

In sentence (1.35), negation can be handled by reversing the polarity of all the words appearing before negation operator 'ਨਹੀਂ' nahīṃ 'not'. But this method does not work accurately for Punjabi sentence given in (1.37) for which its corresponding English translation is given in (1.38).

Punjabi Sentence*: ਮੈਨੂੰ ਕ੍ਰਿਕਟ ਮੈਚ ਦੇਖਣਾ ਪਸੰਦ ਹੈ ਪਰ ਮੈਨੂੰ ਕ੍ਰਿਕਟ ਖੇਡਣਾ ਪਸੰਦ ਨਹੀਂ ਹੈ ।*… (1.37)

Transliterated Punjabi Sentence*: mainūṃ krikaṭ maic dēkhṇā pasand hai par mainūm krikaṭ khēḍṇā pasand nahīṃ hai.*
Equivalent English Sentence*: I don't like playing cricket but I like watching cricket match.* … (1.38)

Here, in sentence given in (1.37), consider the scope of negation operator 'ਨਹੀਂ' nahīṃ 'not' which extends only after conjunction 'ਪਰ' par 'but'. So, change the polarity of all the words appearing before until another negation operator 'ਨਹੀਂ' nahīṃ 'not' does not appear. But, still there will be problems for the sentences. Consider the Punjabi sentence given in (1.39) to understand these problems while its corresponding English translation is given in (1.40).

Punjabi Sentence*: ਨਾ ਸਿਰਫ਼ ਮੈਨੂੰ ਕ੍ਰਿਕਟ ਮੈਚ ਦੇਖਣਾ ਪਸੰਦ ਹੈ ਪਰ ਕ੍ਰਿਕਟ ਖੇਡਣਾ ਵੀ ਪਸੰਦ ਹੈ।*

… (1.39)

Transliterated Punjabi Sentence*: nā sirpha mainūṃ krikaṭ maic dēkhṇā pasand hai pr krikaṭ khēḍṇā vī pasand hai".*
Equivalent English Sentence*: Not only I like watching cricket match but also like playing cricket.* … (1.40)

In sentence given in (1.39), polarity is not reversed before negation operator ’ਨਾ’ nā ‘not’ due to the presence of ‘ਸਿਰਫ਼’ sirpha ‘only’. So, this type of combinations of ’ਨਾ’ nā ‘not’ with other words like ‘ਸਿਰਫ਼’ sirpha ‘only’ has to be considered during designing the algorithm.

1.4 Thesis Outline

This thesis has been divided into 6 chapters. Chapter 1 includes the introduction to sentiment analysis. It also covers applications and challenges in various domains of sentiment analysis. Chapter 2 describes different approaches of sentiment analysis like knowledge based, concepts based *etc*. It also includes techniques of sentiment analysis such as machine learning and rule based. Chapter 3 presents the problem statement, objectives and methodology for developing sentiment analysis system. In chapter 4, rule based sentiment analysis system has been proposed. Rule set and various scoring algorithms for Adverb-Adjective combinations are also described in this chapter. Results of the system are discussed in chapter 5 and chapter 6 concludes the work done in this thesis.

Chapter Summary

In this chapter, importance of sentiment analysis is introduced along with its various applications in the fields like e-commerce, voice of market, voice of customer and opinions about government. Also, challenges of sentiment analysis such as entity identification, need of world knowledge, subjectivity detection, pragmatics and thwarted expectations *etc.* has been discussed in this chapter

Chapter 2

Literature Survey

2.1 Approaches for Sentiment Analysis

Depending on the perspectives of the different persons doing the sentiment analysis, the approach can be keyword-based, concept-based, lexical affinity based, or discourse-driven. These approaches are discussed as follows.

2.1.1 Keyword-based Approach

In Keyword-based approach, main task is the construction of word lexicons. So that, text can be classified into affect category on the basis of presence of affect words like "happy", "awesome", "sad", "bored" [7]. There are number of different ways by which lexicons can be created. Lexicons can be created by starting initially with some seed words and more words can be added by using some linguistic heuristics. In another way, lexicons can also be created by starting with some seed words and other words are added to these seed words on the basis of frequency in the text [8]. But, there are two weaknesses in this approach. First weakness is poor recognition of the affect if negation is involved in the sentence [7]. For example, English sentence given in (2.1) can be correctly classified according to this approach. But, keyword-based approach fails while classifying the sentences like given in (2.2).

Today was an awesome day. … (2.1)

Today was not an awesome day at all. … (2.2)

Second weakness in this approach is that it relies on presence of affect words. There may be some sentences which can convey the affect through underlying meaning rather than the presence of affect words [7]. For example, sentence given in (2.3) describes the weakness of this approach.

My husband just applied for divorce and he wants to take charge of my children. … (2.3)

The sentence given in (2.3) conveys strong emotions but does not use any affect word. So, these types of sentences are not classified by knowledge-based approach.

2.1.2 Concept-based Approaches

The concept-based approaches use web ontologies and semantic networks to achieve semantic text analysis. Thus, these approaches help the system in extracting the conceptual and affective information from natural language opinions. These approaches mainly rely on implicit meaning or feature associated with natural language concepts. So, these approaches are better than the approaches which use keywords and word co-occurrence counts. Concept-based approaches can detect the sentiments better than syntactical techniques. These approaches can also find multi-word expressions even the expressions don't convey any emotion explicitly. The concept-based approaches mainly rely on the knowledge bases. It is difficult for the system in grasping the semantics of natural language text without the presence of comprehensive human knowledge resource. As the knowledge bases contain only typical information associated with concepts, so, it limits their capability to handle semantic variations. Thus, their fixed representation, finally, places bounds on inferences of semantic and affective features associated with concepts [9].

2.1.3 Lexical Affinity

Lexical affinity approach is slightly more advanced than keyword-based approach. This approach assigns a probabilistic 'affinity' to arbitrary words for a particular emotion rather than simply detecting affect words in the text. For example, a probability of 75% can be assigned to the word "accident" to indicate a negative affect, similar in 'car accident' or 'hurt by accident'. The probabilities assigned to words are usually trained from linguistic corpora. Though, this approach is better than keyword-based approach, but this approach has two problems. First problem is that lexical affinity approach mainly operates at the word-level and can easily be tricked by sentences like given in (2.4) and (2.5) [7].

I avoided an accident. … (2.4)

I met my girlfriend by accident. … (2.5)

In sentence (2.4), word "accident" is in negation form while in sentence (2.5) the word "accident" represents other word senses. Second problem is that lexical affinity

probabilities are influenced by a particular domain as prescribed by the source of the linguistic corpora. So, a reusable and domain-independent model can't be developed [7].

2.1.4 Discourse Structures

In discourse structures approach, discourse relations between text components are used as features for classification. For example, in case of movie reviews or product reviews, the overall sentiment of the review is usually expressed at the end of the text [2]. As a result, this approach of sentiment analysis is the discourse-driven. In this approach, the sentiment of the whole review is obtained as a function of the sentiment of the different discourse components in the review and the discourse relations that exist between them. Thus, the sentiment of a paragraph that is at the end of the review might be given more weight in the determination of the sentiment of the whole review in this approach [8].

2.2 Techniques for Sentiment Analysis

Over the past couple of years, researchers have attempted to focus on many specific tasks of sentiment analysis. Earlier many researchers have focused on assigning sentiments to documents by using different techniques like machine learning techniques, rule based techniques and feature extraction. These techniques are discussed as follows.

2.2.1 Machine Learning Techniques

Machine learning techniques are categorized into two categories, *i.e.,* supervised and unsupervised techniques. These techniques are explained as follows.

2.2.1.1 Supervised Techniques

Supervised techniques can be implemented by building a classifier. This classifier is trained by examples which can be manually labeled based on frequent terms in the documents or can be obtained from user-generated user-labeled online source [8]. Naïve Bayes Classifier (NBC), Support Vector Machines (SVM) and Maximum Entropy are mostly used supervised techniques. Supervised techniques perform better than unsupervised techniques [2]. From supervised techniques, SVMs perform better if both positive and negative words are present in the reviews. Therefore, SVMs are more

appropriate for sentiment classification [10]. However, a Naïve Bayes classifier may be more suitable when training data set is small because SVMs requires a large data set in order to build a classifier having high-quality. A brief description of Naïve Bayes Classifier and Support vector machines is given as follows.

i) Naïve Bayes Classifier

Naïve Bayes Classifier is based on Bayesian theorem and convenient when the range of the inputs is high. In spite of its simplicity, Naïve Bayes Classifier performs better than other classification methods [11].

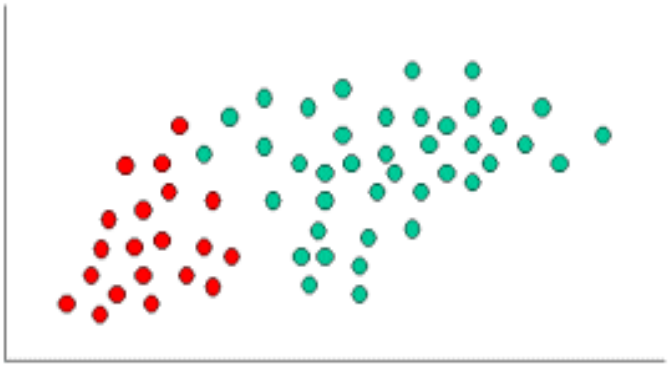

Fig. 2.1: Demonstration of Naïve Bayes Classifier [11]

As shown in the Figure 2.1, objects can be classified as either RED or GREEN. Main task is the identification of class of new objects. This decision can be taken on the basis of existing objects. Since, there are double the numbers of GREEN objects than RED as shown Figure 2.1. So, it can be thought that new objects will more likely belong to GREEN class. This faith is known as the prior probability in the Bayesian analysis. Prior probabilities work on the basis of previous experience. In this case, prior probabilities are the percentage of GREEN and RED objects. Suppose, there are 60 objects, 40 of which are GREEN and 20 are RED. Thus, prior probabilities for GREEN and RED objects will be as given in (2.6) and (2.7) [11].

Prior probability for GREEN $\propto$ *(No. of GREEN objects/Total no. of objects)*

$$\propto (40/60) \quad \ldots (2.6)$$

Prior probability for RED $\propto$ *(No. of RED objects/Total no. of objects)*

$$\propto (20/60) \quad \ldots (2.7)$$

Now, new object (X) shown as WHITE circle in Figure 2.2, can be classified using prior probability. Since all the objects are well clustered, it is logical to infer that there are more GREEN (or RED) objects in the area of X. So, there are more chances that new object will belong to GREEN class.

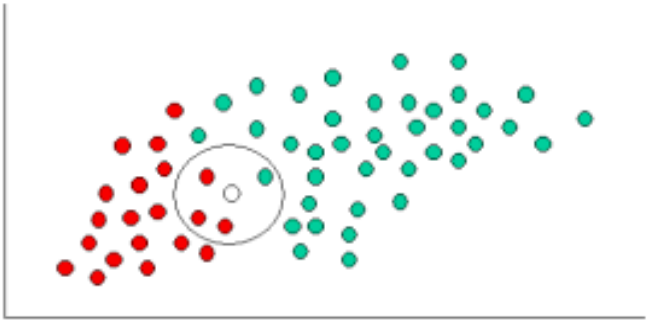

Fig. 2.2: Classification of New Object in NBCs

To measure this likelihood, a circle is drawn around X which encloses a number of points rather than their class labels. Then, number of points in the circle is calculated to compute the likelihood as given in (2.8) and (2.9) [11].

Likelihood of X given GREEN ∝ (No. of GREEN in the area of X/ Total no. of GREEN cases)

$$\propto (1/40) \quad \ldots (2.8)$$

Likelihood of X given RED ∝ (No. of RED in the area of X/ Total no. of RED cases)

$$\propto (3/40) \quad \ldots (2.9)$$

Although, prior probabilities has shown that new object (X) will belong to GREEN class but likelihood has shown that it will belong to RED class. But, in Bayesian analysis, the final classification is done by calculating a posterior probability using the Bayes' rule. A posterior probability can be calculated by combining both the prior probability and the likelihood as given in (2.10) and (2.11) [11].

Posterior probability of X being GREEN ∝

Prior probability of GREEN * Likelihood of X given GREEN

$$= (4/6) * (1/40) = (1/60) \quad \ldots (2.10)$$

Posterior probability of X being RED ∝

Prior probability of RED * Likelihood of X given RED

= (2/6) * (3/20) = (1/20) … (2.11)

Finally, new object X is classified as RED because it has achieved largest posterior probability.

ii) Support Vector Machines

Support Vector Machines work on the idea of decision planes that specify decision boundaries. A set of objects belonging to different class memberships are separated by decision planes [11]. An example to illustrate the concept of linear SVMs is shown in Figure 2.3(a). In this example, the objects either belong to GREEN class (or RED class). The separating line specifies the decision boundary. On the right hand side of the boundary, all objects are GREEN and to the left hand side of boundary, all objects are RED. A new object (white circle) will be classified as GREEN if it falls to the right side of the boundary or classified as RED if it falls to the left side of the boundary.

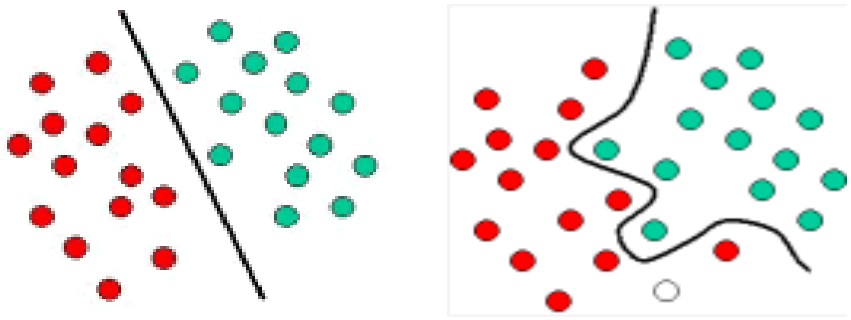

Fig. 2.3: (a) Example of linear SVM, (b) Example of hyperplane SVM [11]

A classifier that partitions a set of objects into their respective domains with a line is called linear classifier and partitioning with a curve is known as hyperplane classifier [11]. An example of hyperplane classifier is shown in Figure 2.3(b).

Figure 2.4 shows the basic concept behind Support Vector Machines. In this figure, original objects are mapped applying a set of mathematical functions known as kernels. This process of reorganizing the objects is known as mapping or transformation. The figure shows that the mapped objects are linearly separable [11]. Thus, find an optimal line rather than constructing the complex curve that can divide the GREEN and the RED objects.

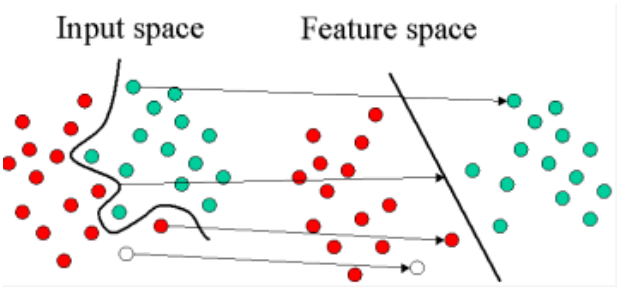

Fig. 2.4: Mapping of objects in SVMs [11]

Existing Work using Supervised Techniques

Pang *et al.* (2002) used three machine learning supervised methods such as Support Vector Machines, Naïve Bayes classifier and Maximum Entropy for sentiment classification. Out of three, SVM showed the best performance as comparison to NBCs. An accuracy of 82.9% was achieved for the movie reviews collected from Internet Movie Database (IMDb) [2]. It was concluded that sentiment classification is more challenging task than topic based categorization. As in sentiment classification, the most important task is selecting an appropriate set of features. Term presence and there frequency are the most commonly used features of sentiment classification. These features include frequency or presence of uni-grams or *N*-grams. It was claimed that uni-grams give better results than bi-grams [6].

Dave *et al.* (2003) have reported that bi-grams and tri-grams give better results for product reviews. Thus, an accuracy of 87% was achieved using NB for product reviews [12].

Pang *et al.* (2004) proposed a novel machine learning method and performed subjectivity identification as a pre-processing step to do sentiment analysis. A better accuracy was achieved than SVM and NBC by using graph based formulation method [13].

Cui *et al.* (2006) have argued that SVMs are more appropriate for sentiment classification because these can perform better when review contains both positive and negative words. However, when the set of training data is small, then Naïve Bayes classifier might be more appropriate because SVMs requires a huge set of data to make a high-quality classifier [10].

Chen *et al.* (2006) used decision trees, SVM and NBC for sentiment based classification. An accuracy of 84.59% was achieved for reviews of book, *The Da Vinci Code,* collected from Amazon.com [14].

Boiy *et al.* (2007) performed experiments using SVM, NBC and maximum entropy to perform automatic sentiment analysis in on-line text and an accuracy of 90.25% was achieved for movie and car brands reviews [15]. Annett *et al.* (2008) used same dataset of movie reviews for experiment. Different approaches like SVM, NBC, alternating decision tree and lexical (WordNet, Web Search) based approaches were used for sentiment analysis and greater than 75% accuracy was achieved for movie reviews [16].

Ye *et al.* (2009) performed sentiment analysis for data set retrieved from tourists' reviews in the travel column of *Yahoo.com* using NBC, SVM and the character based *N*-gram model. It was reported that SVM performed best in the study and gave 85.14% best accuracy [17].

Paltoglou *et al.* (2010) also proposed a combined approach for sentiment analysis based on SVM which has given an accuracy of 96.90% for movie reviews [18].

2.2.1.2 Unsupervised Techniques

In unsupervised technique, sentiment classification is done by comparison. In this technique, the features of a given text are compared against word lexicons whose sentiment values are decided prior to their use [8]. Hierarchical clustering and partial clustering are mostly utilized algorithms of unsupervised technique. Both algorithms are discussed as follows.

i) Hierarchical Clustering

Hierarchical clustering algorithms partition the objects into tree like structure where each node represents a cluster. There are zero or more child nodes in each node of the tree. Hence, tree grows down by its nature [19].

ii) Partial Clustering

In partial clustering algorithm, objects are partitioned. Objects can change the clusters on the basis of dissimilarity. K-means clustering algorithm is mostly used algorithm of partial clustering algorithm [19].

Existing Work using Unsupervised Techniques

Turney (2002) used unsupervised method for sentiment detection. He used "poor" and "excellent" seed words for computing the semantic orientation of phrases. Semantic

orientation is measured by point-wise mutual information. An accuracy of 66% was achieved for movie review domain and 84% for automobile reviews [3].

Hu *et al.* (2004) proposed a simple method by using the adjective synonym set and antonym set in WordNet to find the semantic orientations of adjectives [20].

Peng *et al.* (2010) used part-of-speech (POS) patterns as features on websites to extract the sentiment phrases of each review. Then, unknown sentiment phrase was used as a query term to receive top-*N* appropriate phrases from a search engine. After this, sentiments of unknown sentiment phrases are computed on the basis of sentiments of nearby known appropriate phrase using lexicons [21].

Li *et al.* (2010) developed an approach by using k-means clustering algorithm to cluster documents into positive group and negative group and achieved an accuracy of 70% [22].

While the machine learning based approaches provided better classification accuracy, but required a lot of training time and pre-classified training corpus. Moreover semantic orientation based approaches did not gave good performance, but returned results quickly [23].

2.2.2 Feature Extraction

Since most of sentiment analysis approaches depend on machine learning techniques and the salient features of text or documents are represented as feature vector [8]. These important features used in sentiment analysis are discussed as follows.

2.2.2.1 Term Presence or Term Frequency

In standard information retrieval and text classification, term frequency is preferred over term presence. However, in sentiment analysis for movie reviews, show that this is not the case in sentiment analysis [2]. It has been claimed that this is one indicator that sentiment analysis is different from standard text classification where term frequency is taken to be a good indicator of a topic.

Term can be either uni-grams, bi-grams or other higher-order *N*-grams. Out of these terms, it is not clear that which one gives better results. It has been claimed that uni-grams give better results than bi-grams in case of movie review sentiment analysis [2].

But, it also has been reported that bi-grams and tri-grams give better results in case of polarity classification of product-reviews [12].

2.2.2.2 Opinion Words and Phrases

Opinion words and phrases such as “like”, “nice”, “hate”, “I'd suggest that...” are words or phrases that convey positive or negative opinions. Statistical-based or Lexicon-based are the main approaches which identify the semantic orientation (positive or negative) or polarity of opinion words [8].

2.2.2.3 POS (Part of Speech) Tags

POS is used to remove the ambiguity [25]. For example, adjectives and adverbs can be identified by using POS tags which are usually used as sentiment indicators. But, it has been found that adjectives performed worse than the same number of uni-grams selected on the basis of frequency [3].

2.2.2.4 Syntax and Negation

Collocations and other syntactic features can be employed to enhance performance. In some short text (sentence-level) classification tasks, algorithms using syntactic features and algorithms using *N*-gram features were found to give same performance. Because, negation reverses the potential of sentiment, so it is also an important feature to bear in mind [5]. There are attempts to model negation for better performance. It has been reported that 3% accuracy improvement for electronics product reviews by handling negation.

2.2.3 Rule-based Technique

In rule based technique, if a rule has “if-then” relation then it consists of an antecedent and its associated consequent as given in (2.12).

antecedent → *consequent* … (2.12)

An antecedent describes a condition and can be either a token or a series of tokens that are concatenated by the “∧” operator. A token can be either ‘?’ denoting a proper noun, a word or ‘#’ denoting a target term. A target term denotes the perspective in which a set of

documents occurs like the name of a politician or company, a brand of a product or a title of the movie. A consequent denotes a sentiment that can be either positive or negative and it is the result of the condition described by the antecedent as given in (2.13) [24].

$\{token_1 \wedge token_2 \wedge ... \wedge token_n\} \Rightarrow \{+|-\}$ … (2.13)

The two simple rules given in (2.14) and (2.15) depend on two sentiment bearing words, each of which denotes an antecedent.

$\{excellent\} \Rightarrow \{+\}$ … (2.14)

$\{absurd\} \Rightarrow \{-\}$ … (2.15)

Assume two sentences given in (2.16) and (2.17).

i) Mobile-A is more expensive than Mobile-B. … (2.16)

ii) Mobile-A is more expensive than Mobile-C. … (2.17)

The target word of sentences (2.16) and (2.17) is Mobile-A. The rule deduced from these sentences is as given in (2.18).

$\{\# \wedge more \wedge expensive \wedge than \wedge ?\} \Rightarrow \{-\}$ … (2.18)

The analysis of this rule is as the target word, *i.e.,* Mobile-A is less favorable than the other two mobiles because of its price that is expressed by the rule (2.18). Here, the center of attention is on the price attribute of the Mobile-A.

2.3 Role of UNL (Universal Networking Language) in Knowledge Extraction

Knowledge extraction involves morphological, syntactic, semantic and pragmatic analysis of the text of a source language. Researchers have explored the use of UNL contexts such as knowledge representation, knowledge management, multilingual search engine, language independent Universal Digital Library *etc.* The description of various systems developed by researchers to extract knowledge using UNL is given in Table 2.1.

Table 2.1: Systems using UNL for Knowledge Extraction

System Name	Description	Developers
VoiceUNL	Speech to Speech Machine Translation	Tomokiyo *et al.* (2003) [26]

Table 2.1: Systems using UNL for Knowledge Extraction

System Name	Description	Developers
Agro-Explorer	Meaning based, interlingua search engine	Surve *et al*. (2004) [27]
Question Answering system	Used UNL as a tool for language-independent semantics	Mukerjee *et al.* (2005) [28]
CELTA	Multilingual business-to-business web platform	Bértoli *et al.* (2005) [29]
'Pivot' XML based architecture	For multilingual, multiversion documents using UNL	Hajlaoui *et al.* (2005) [30]
eXtended Markup Language (XML) UNL model	Used for knowledge-based annotation	Cardeñosa *et al.* (2005) [31]
Universal Communication Language (UCL) (derived from UNL)	Used as the language for communication among software agents and among humans	Montesco *et al.* (2005) [32]
aAQUA (an improved form of Agro-Explorer system)	An online multilingual and multimedia agricultural portal for distributing information from and to rural regions	Ramamritham *et al.* (2006) [33]
Library Information System (LIS)	Provides access to the books of different languages into the native language of the user	Alansary *et al.* (2006) [34]
A multilingual search engine	Requires EnConverter to convert the contents of source language to UNL	Karande (2007) [35]
LOOK4	For enhancement of web search results with Universal Words (UWs) and WordNet	Avetisyan *et al.* (2010) [36]

Thus, UNL helps in extracting knowledge from text. The extracted knowledge helps in performing sentiment analysis process. To understand the UNL, the framework of UNL is described as follows.

2.3.1 Introduction to UNL

UNL is a language that enables computers to process information across different languages. It is used to replicate the functions of natural language used by humans for communication. It has the capability to represent information and knowledge provided by natural language. The incentive behind the development of UNL is to offer an Interlingua representation such that there is same representation for semantically equivalent sentences of all languages.

The UNL Programme was started in 1996 in the Institute of Advanced Studies (IAS) of United Nations University (UNU) in Tokyo, Japan. In January 2001, UNU established an autonomous organization, the Universal Networking Digital Language (UNDL) Foundation. It is responsible for the development and management of the UNL Programme. From the very beginning, researchers from all over the world have been working on creation of linguistic resources and software for UNL systems [37].

The UNLization and NLization are the main processes in a UNL system as shown in Figure 2.5. The process of conversion of source language expression into the UNL expression is known as UNLization. The process of conversion of UNL expressions into a target language expression is called NLization.

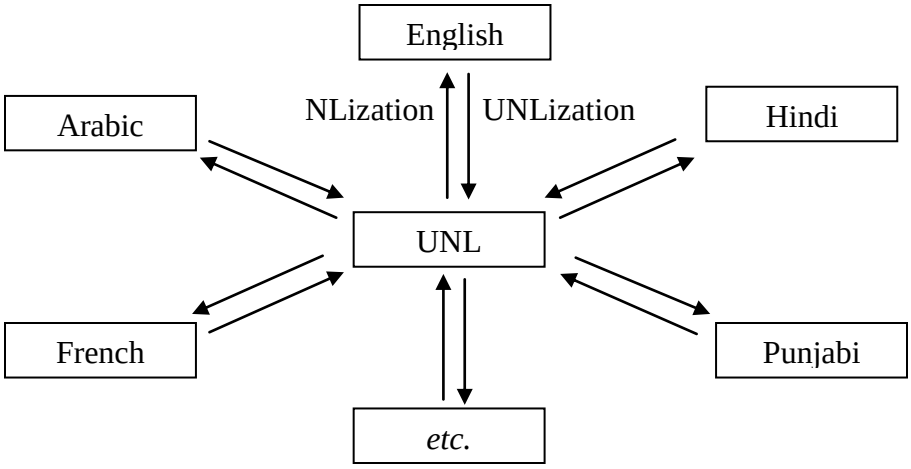

Fig. 2.5: A UNL system

2.3.2 UNL Format for Information Representation

UNL information can be represented in three different types of semantic units. These units are Universal Words (UWs), Relations and Attributes. UNL depicts information sentence by sentence [38]. Each sentence is changed into a hyper-graph (also known as UNL graph) having concepts represented as nodes and relations as directed arcs. The concepts are represented by UWs and UNL relations are used to describe the role of UW in a sentence. The subjective information is described by UNL attributes [39].

2.3.2.1 Universal Words

Universal Words form the vocabulary of UNL. These words are represented as nodes in a UNL graph. The nodes in the graph are interlinked by relations and modified by attributes. A UW is a character string followed by a list of constraints [39]. The syntax of a UW is given in Table 2.2:

Table 2.2: Syntax of UW

Universal Word
<UW> ::= <headword> [<constraint list>]

Here, the headword can be an English word or a sentence or a phrase. It is considered as a label. This is also known as basic UW. The constraint list is used to describe the sense of word. This list contains the UNL relations that represent the objective knowledge of concept in headword. The constraint list is also used to restrict the range of the concepts that an English expression represents and thus also helps in disambiguation.

The UWs are divided into four categories as follows:

- Basic UWs
- Restricted UWs
- Extra UWs
- Temporary UWs

i) Basic UWs

The headwords having no constraint list are known as Basic UWs. For example, “water”, “go”, “run”, “come” *etc.*

ii) Restricted UWs

A UW with constraint list is known as Restricted UW. For example, the basic UW "run", with no constraint list, denotes the concepts of "management of an organization by a person" (*e.g.*, "He is running the organization very well"), "driving of vehicle" (*e.g.*, "He runs the car very fast"), "move at a speed faster than walking" (*e.g.*, "He runs very fast in the race"), "short visit" (*e.g.*, "He runs up to Mumbai"), and so on. The UW "run" is restricted with constraint list to represent these concepts unambiguously as shown in Table 2.3.

Table 2.3: Example of Restricted UW

Unambiguous forms of UW "run"
run(agt>human, equ>manage, obj>organization)
run(agt>human, equ>drive, obj> vehicle)
run(agt>human, pur> short visit)

Here, "agt", "obj", "equ" and "pur" are UNL relations.

iii) Extra UWs

Extra UWs are special type of restricted UWs that are used to represent the concepts that are not available in English language. These words are represented as headwords using English characters and constraint list gives idea about the nature of the concept. The examples of extra UWs are shown in Table 2.4.

Table 2.4: Example of Extra UWs

Extra UWs
vaisakhi (icl>festival)
bhangra (icl>dance)

iv) Temporary UWs

Temporary UWs may not have a definition. For example, a number or an e-mail or URL need not be necessarily defined. If these UWs appear in a UNL document then these are treated as temporary UWs.

2.3.2.2 UNL Relations

Relations are basic building blocks of UNL sentences. UNL expressions are formed using binary relations. A binary relation includes a UNL relation and two UWs. Each UNL relation has different label according to different role played by it. A relation is chosen by considering a number of factors [39, 40]. The format of binary relation of UNL is shown in Table 2.5.

Table 2.5: Syntax of UNL Relation

UNL Relation
<relation> (<uw1>, <uw2>)

In UNL relation, <uw1> acts as parent of the relation and <uw2> acts as the child of the relation whereas <relation> defines the relation between these universal words <uw1> and <uw2>. There are 46 relations in all that are defined for UNL [39]. The description of these relations is given in Table 2.6 [42].

Table 2.6: Types of Relations in UNL

UNL relation	**Description**	**Constituent elements**	**Examples**
1. agt	defines a thing that initiates an action	Agent, Unergative verbs (intransitive verb semantically have an agent subject) like "sleep", "cough", "run", *etc.*	John slept … agt(slept, John) John killed Mary. agt(killed, John)
2.aoj	defines a thing that is in a state or has an attribute	Stative verbs like "believe", "like", "understand", "know", "dislike", "love", *etc.*	John believes in Mary. aoj(believes, John) John knows Mary. aoj(knows, John)
		aoj (general attribute)	John is sad. aoj(sad, John) John looks sad …. aoj(sad, John)

Table 2.6: Types of Relations in UNL

UNL relation	Description	Constituent elements	Examples
3.obj	defines a thing in focus that is directly affected by an event or state	Unaccusative verbs like "die", "fall", "melt", *etc.*	John died. obj(died, John) The snow melts. obj(melts, snow)
		obj (direct object)	John killed Mary. obj(killed, Mary) John knows Mary. obj(knows, Mary)
4.rsn	defines a reason why an event or a state happens	Reason	*He goes … because of … illness.* rsn(go(icl>do), illness(icl>thing)) … known for … beauty. rsn(known(aoj>thing), beauty(icl>abstract thing))

All binary relations that create a UNL expression posses directions and the semantic network of a UNL expression is known as directed hyper-graph.

2.3.2.3 UNL Expression Hyper-graph

Each UNL expression can be represented by semantic hyper-network. Each node in the graph and <uw1> and <uw2> of a binary relation can be exchanged with a semantic network. In the UNL expression, a node that consists of a semantic network is known as "scope". A scope can also be connected with other universal words or scopes. The format of binary relations of UNL expression is in Table 2.7.

Table 2.7: Syntax of UNL Expression

UNL Expression
<relation>.<scope-id> (<node1>, <node2>)

Here, <scope-id> is the ID for distinguishing a scope. <node1> and <node2> can be a UW or a <scope node>. A <scope node> is given in the format ": <scope-id>".

2.3.2.4 UNL Attributes

Attributes are used to describe subjective information in a sentence. These attributes exhibit the point of view of speaker in a given sentence [40]. There are 87 attributes to convey the semantic content of a sentence. UNL attributes are divided into eight groups. Descriptions of some of these groups with some of the corresponding UNL attributes are given in Table 2.8 [39].

Table 2.8: Types of UNL Attributes

Concept	**Attributed as**
Speaker's view on aspects of Even	@begin, @complete, @continue, @experience, @progress,
Speaker's view of reference to Concepts	@def, @indef, @not
Speaker's attitudes	@affirmative, @confirmation, @exclamation, @imperative, @polite, @request, @respect
Speaker's feelings and judgments	Attributes to represent ability: @ability
	Attributes to represent possibility: @certain, @may, @possible, @probable
	Attributes to represent emotion: @admire, @blame, @regret, @surprised

Attributes are thus used to include information about time and aspect of the event, modality of the predication, reference of the entities mentioned, number and/or gender, *etc.* For example, in the sentence, "The boy eats potatoes in the kitchen", attributes are needed to express plurality in the object ("*potato", i.e., "@pl"*), to indicate definite reference to agent ("*boy", i.e., "@def"*), to indicate definite reference to place ("*kitchen", i.e., "@def"*) and to denote the head of expression ("*eat", i.e., ".@entry"*) [41].

2.3.3 UNL Sentence

UNL sentence is the fundamental unit of representation inside the UNL framework. It consists of nodes (UWs) interlinked with binary semantic relations and modified by attributes. UNL sentences can be represented by two ways, *i.e.*, table format and list format. In the table format, universal words and relations constitute a single structure, while in list format, universal words and relations are represented separately [39]. The syntax of table format is given in Table 2.9.

Table 2.9: Syntax of UNL Sentence in Table Format

<UNL sentence>	::=	<list of relations>
<list of relations>	::=	<binary relation>[<binary relation>...]
<binary relation>	::=	<relation>[“:”<Scope-ID>]“(”<sourcenode>, <target node>“)”
<source node>	::=	<UW+attributes>
<target node>	::=	<UW+attributes>
<UW+attributes>	::=	<UW>{“:”<Scope-ID}[<attribute list>] “:” <UW-ID>

Here, ‘<’ and ‘>’ represent a non-terminal symbol; ‘[’ and ‘]’ denotes an omissible part; ‘{’ and ‘}’ specify a range; ‘::=’ indicates that left hand side can be replaced by the right hand side; ‘...’ indicates that there can be more than 0 times repetition of the front part and predefined delimiters are enclosed within “ ”. This format is illustrated below with the help of an example sentence.

Example Sentence: Ram killed Mary. … (2.19)

Corresponding UNL of the example sentence given in (2.19) is shown in Table 2.10.

Table 2.10: Example of UNL Sentence given in (2.19)

UNL Relation
{org:en}
Ram killed Mary
{/org}
{unl}
obj(kill.@past, Mary)
agt(kill.@past, Ram)
{/unl}

As discussed earlier, UNL sentence can also be represented in the form of UNL graph. The UNL graph of example sentence given in (2.18) is shown in Figure 3.2.

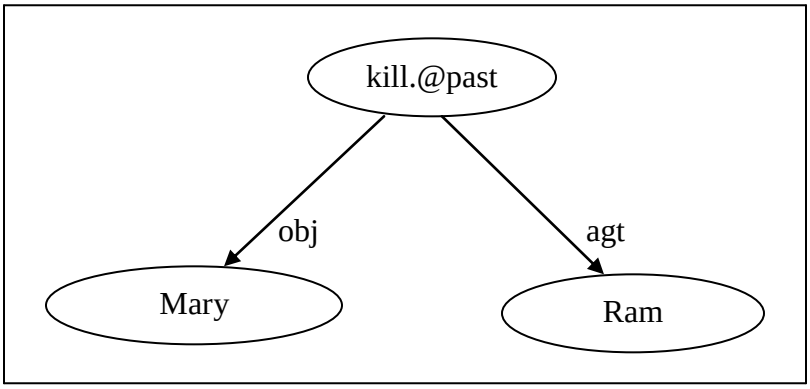

Fig. 2.6: UNL Graph of Example Sentence given in (2.19)

Chapter Summary

In this chapter, various approaches for sentiment analysis like keyword based, concept based approaches *etc.* have been discussed. Also, the different techniques of sentiment analysis such as machine learning techniques, feature extraction and rule based techniques have been described. A brief description of UNL and its importance in knowledge extraction also has been included in this chapter.

Chapter 3

Problem Statement

Sentiment analysis is a process of extracting information from user's opinions. The decisions of the people get affected by the opinions of other people. Today, if any person wants to buy a product or wants to watch a movie then he or she will first search the reviews and opinions about that product or movie on social media, blogs *etc.* As there is a huge explosion of user's opinion on social media like Twitter, Facebook and other user forums, then identification of sentiment becomes very difficult from this huge data manually. So, there is a need of automated sentiment analysis system.

3.1 Objectives

The main objective of this research work is to perform the sentiment analysis for English sentences. In order to perform this task, following objectives were proposed to be carried out.

i) To study the existing approaches and techniques for sentiment analysis.

ii) To extract the twitter posts by user defined parameters using twitter APIs.

iii) Creation of Adjective and Adverb Score files.

iv) Creation of rule set and scoring algorithms for Adverb-Adjective Combinations (AACs).

v) To explore the use of Universal Networking Language for sentiment analysis.

vi) Development of sentiment analysis system for analysis of twitter posts and English documents.

3.2 Methodology

To achieve all the objectives discussed in section 3.1, the following methodology has been used.

i) Literature survey has been carried out by study of existing approaches like knowledge based, concept based approaches *etc.* and techniques like

supervised, unsupervised and rule-based to perform the sentiment analysis process.

ii) Python language has been used with twitter APIs to extract the twitter posts from twitter.

iii) Sentiment bearing adjectives and adverbs are identified. And positive or negative scores are assigned according to sentiment beard by them.

iv) Rule set for AACs and scoring algorithms like variable scoring algorithm, adjective priority scoring algorithm *etc.* has been created. The scoring algorithms adjust the score of adjectives according to type of adverb which adjoins it.

v) UNL has been explored for resolving the challenges like entity identification *etc.*

vi) A system has been developed which extracts the twitter posts from twitter and compute sentiment and score of each tweet. It also performs sentiment analysis of English documents.

Chapter 4

Implementation

4.1 Architecture of Proposed Rule Based Sentiment Analysis System

To perform sentiment analysis, simple English text file or tweets extracted from twitter are inputted by user. Then, system works on it and computes its sentiment and score. The architecture shown in Figure 4.1, illustrates the working of rule based sentiment analysis system.

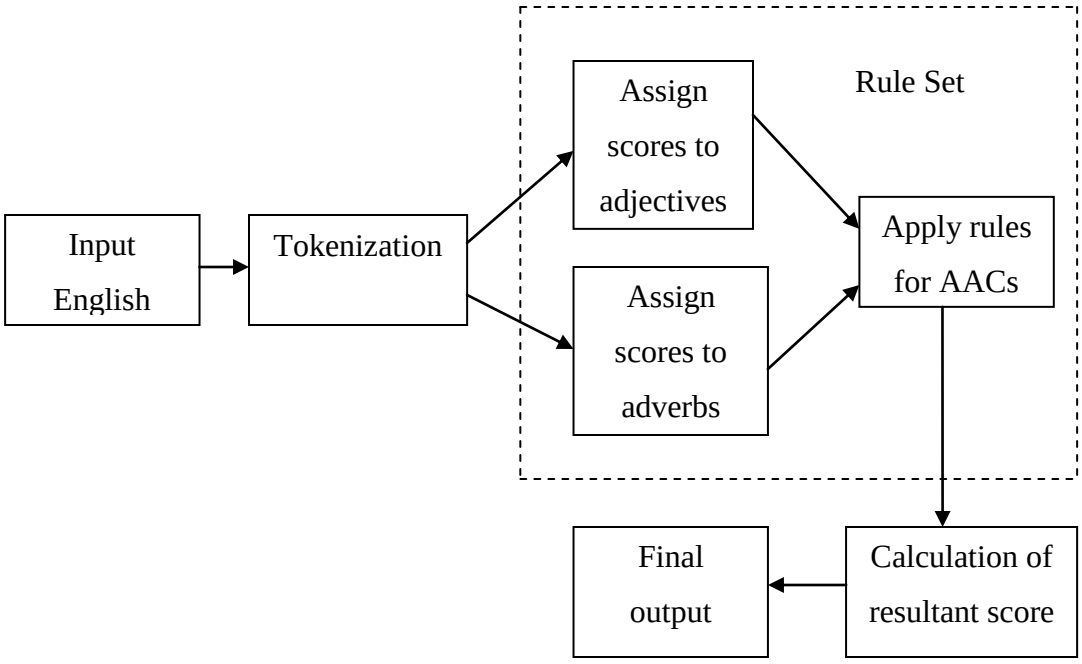

Fig. 4.1: Proposed Architecture of Rule Based Sentiment Analysis System

The main components of the system are explained as follows.

4.1.1 Tokenizer

The system takes English text file as input. The inputted file is splitted into tokens by tokenizer. A token is a part of a sequence of characters in a text that are combined together as a valuable semantic unit for processing. The tokenizer converts a sentence into word level tokens consisting of words, punctuation marks, and other symbols. The output of tokenizer for the example sentence given in (4.1) is shown in (4.2).

Example Sentence: *She is very beautiful girl.* … (4.1)

Tokenizer:

---

i. *She*

ii. *is*

iii. *very*

iv. *beautiful*

v. *girl* … (4.2)

At this stage there is no part-of-speech information resolved for the tokens.

4.1.2 Rule set for AAC (Adverb-Adjective Combination)

Parts of Speech information is most commonly utilized in all NLP tasks. It is used to disambiguate sense which in turn is used to guide feature selection [5]. Researchers mainly use adjectives and adverbs as features to find the sentiment in a text or document. A general description of adjectives, adverbs and AACs is given as follows.

4.1.2.1 Adjectives

Adjectives are most commonly used as features amongst all parts of speech. There is strong correlation between adjectives and subjectivity of text. Even all the parts of speech play an important role, but only adjectives as features depict the sentiments with high accuracy. An accuracy of around 82.8% has been achieved in movie review domains by using adjectives only as features [2]. Some examples of positive and negatives are shown in Table 4.1.

Table 4.1: Positive and Negative Adjectives

Positive Adjectives		**Negative Adjectives**	
dazzling	awesome	suck	boring
brilliant	mesmerizing	terrible	bad
phenomenal	Cool	awful	stupid
excellent	spectacular	unwatchable	slow
fantastic	exciting	hideous	clichéd

4.1.2.2 Types of Adverbs

Adverbs have no prior polarity. But, adverbs can play a major role in identification of sentiment of sentence when used with sentiment bearing adjectives. The sentiment value of adjectives gets altered when adverbs are used [27]. On the basis of the level to which adverbs can modify the sentiment value; these can be classified as shown in Table 4.2.

Table 4.2: Types of Adverbs of Degree

Type of adverb	Description	Examples	Example Sentence
Adverb of affirmation	defines an adverb which is used to affirm a sentence as true and used to answer the questions raised by others	absolutely, alright, surely, certainly, clearly, exactly, totally, obviously, positively, really, very	Is Anne coming today? Yes, she is.
Adverbs of doubt	indicate doubt or suspicion	possibly, may be, probably	He may probably come today.
Strongly Intensifying Adverbs	Defines those adverbs which increase the strength of another adjective or adverb in the sentence	exceedingly, immensely, extremely	The student is extremely bright.
Weak Intensifying Adverbs	defines those adverbs which decrease the strength of another adjective or adverb in the sentence	barely, slightly, mildly	She mildly disapproved of his actions.
Negation and Minimizers	defines an adverb which is used to negate a sentence	Almost, contradictorily, invalidly, never, no, not , rarely	Can Anne speak Spanish? No, she can't.

4.1.2.3 Types of AACs

There are two types of AACs as follows.

i) Unary AACs

ii) Binary AACs

i) Unary AACs

In unary AACs, there is one adjective and one adverb. The sentiment score of the adjective get altered when an adverb adjoins it. The syntax for unary AACs is given in (4.3).

<adverb><adjective> … (4.3)

For example, "very good" and" really wonderful"

ii) Binary AACs

In binary AACs, there is one adjective and more than one adverb. By repeatedly modifying the score of the adjective when an adverb gets added to it, the sentiment score of the binary AACs is calculated. So, binary AACs can also be defined in terms of two unary AACs repeatedly defined. The syntax for binary AACs is given in (4.4).

$<adverb_i , adverb_j> <adjective>$ … (4.4)

For example, "very very good" and "very less expensive

4.1.2.4 Rules for AACs

On the basis of score of adjective (adj) and adverb (adv), a score is assigned to AAC to compute its sentiment. Certain axiomatic rules are explained to specify the way with which the scores of the adjectives get modified when adverbs are used. One such axiom can be stated as [27]:

"The score of weakly intensifying adverb or adverb of doubt will be less than or equal to score of strongly intensifying adverb or adverb of affirmation."

Then, certain functions are discussed in order to evaluate the axioms. A function ***f*** takes an AAC and returns its final score [27].

i) Functions to Quantify Axioms

Affirmative (AFF) and Strongly Intensifying Adverbs (STRONG)
If score(adj) > 0 and adv ∈ AFF ∪ STRONG, then: f (adv, adj) ≥ score (adj). If score(adj) < 0 and adv ∈ AFF ∪ STRONG, then: f (adv, adj) ≤ score (adj).

Description of "AFF" and "STRONG" Function:

If score of an adjective is greater than 0, *i.e.*, adjective is positive and adverb belongs to affirmative or strongly intensifying adverb then score of combination of both adjective and adverb will be greater than or equal to score of adjective only in a sentence [1].

For example, "wonderful" is a positive adjective and "really" belongs to affirmative/strongly intensifying adverb. So, ***f*** value for "really wonderful" is more positive than the positive adjective "wonderful",

If score of an adjective is less than 0, *i.e.*, adjective is negative and adverb belongs to affirmative or strongly intensifying adverb then score of combination of both adjective and adverb will be less than or equal to score of adjective only in a sentence [1].

For example, "bad" is a negative adjective and "very" belongs to affirmative/strongly intensifying adverb. So, ***f*** value for "very bad" is more negative than the score of the negative adjective "bad".

Weakly Intensifying Adverbs (WEAK)
If score (adj) > 0 and adv ∈ WEAK, then: f (adv, adj) ≤ score (adj). If score (adj) < 0 and adv ∈ WEAK, then: f (adv, adj) ≥ score (adj).

Description of "WEAK" Function:

If score of an adjective is greater than 0, *i.e.*, adjective is positive and adverb belongs to weakly intensifying adverb then score of combination of both adjective and adverb will be less than or equal to score of adjective only in a sentence [1].

For example, “good” is a positive adjective and “barely” belongs to weak intensifying adverb. So, ***f*** value for “barely good” is more negative than the score of the positive adjective “good”.

If score of an adjective is less than 0, *i.e.,* adjective is negative and adverb belongs to weakly intensifying adverb then score of combination of both adjective and adverb will be greater than or equal to score of adjective only in a sentence [1].

For example, “bad” is a negative adjective and “slightly” belongs to weakly intensifying adverb. So, ***f*** value for “slightly bad” is more positive than the score of the negative adjective “bad”.

Adverbs of Doubt (DOUBT)
If score (adj) > 0 , adv ∈ DOUBT and adv’ ∈ AFF ∪ STRONG, then: f (adv, adj) ≤ f (adv’, adj). If score (adj) < 0 is negative, adv ∈ DOUBT and adv’ ∈ AFF ∪ STRONG, then: f (adv, adj) ≥ f (adv’, adj).

Description of “DOUBT” Function:

If score of an adjective is greater than 0, *i.e.,* adjective is positive, one adverb (adv) belongs to adverbs of doubt and there is another adverb (adv’) that belongs to weakly intensifying adverb then score of combination of both adjective and adverb (adv) will be less than or equal to score of combination of both adjective and adverb (adv’) in a sentence [1].

For example, ***f*** value for “probably good” is less than “very good”. As, “probably” and “very” belong to adverb of doubt and adverb of affirmative or strongly intensifying adverbs respectively. Here, “good” is a positive adjective.

If score of an adjective is less than 0, *i.e.,* adjective is negative, one adverb *(adv*) belongs to adverbs of doubt and there is another adverb (*adv’*) that belongs to weakly intensifying adverb then score of combination of both adjective and adverb (*adv*) will be greater than or equal to score of combination of both adjective and adverb (*adv’*) in a sentence [1].

For example, ***f*** value for "probably bad" is more than "very bad". As, "probably" and "very" belong to adverb of doubt and adverb of affirmative or strongly intensifying adverb respectively. Here, "bad" is a negative adjective.

Minimizers (MIN)
If score (adj) > 0 and adv ∈ MIN, then: f (adv, adj) ≤ score (adj). If score (adj) < 0 and adv ∈ MIN, then: f (adv, adj) ≥ score (adj).

Description of "MIN" Function:

According to this function, if score of an adjective is greater than 0, *i.e.*, adjective is positive and an adverb belongs to Minimizers then score of combination of both adjective and adverb will be less than or equal to score of adjective only in a sentence [1].

For example, "hardly good" is less positive than the positive adjective "good". Here, "hardly" belongs to adverb of Minimizers and "good" is a positive adjective.

If score of an adjective is less than 0, *i.e.*, adjective is negative and adverb belongs to Minimizers then score of combination of both adjective and adverb will be greater than or equal to score of adjective only in a sentence [1].

For example, "hardly bad" is less negative than the positive adjective "bad". Here, "hardly" belongs to adverb of Minimizers and "bad" is a negative adjective.

4.1.3 Computation of Scores for AACs

Scoring algorithms are used to calculate the final score of inputted English text. Scores are assigned to sentiment bearing adjectives and adverbs by using following algorithms given in (4.1), (4.2) and (4.3). These algorithms are discussed as follows.

4.1.3.1 Variable Scoring Algorithm (VSC)

The algorithm 4.1 changes the score of the AAC by using the function ***f*** defined as follows [1].

Algorithm 4.1: Variable Scoring Algorithm

If adv $\in$ AFF $\cup$ STRONG, then:

$$f_{VS}(adv, adj) = score(adj) + (5\text{-}score(adj)) * score(adv). \quad \ldots (4.5)$$

if score(adj) > 0. If score(adj) < 0,

$$f_{VS}(adv, adj) = score(adj) - (5\text{-}score(adj)) * score(adv). \quad \ldots (4.6)$$

If adv $\in$ WEAK $\cup$ DOUBT, VS reverses the above and returns

$$f_{VS}(adv, adj) = score(adj) - (5\text{-}score(adj)) * score(adv). \quad \ldots (4.7)$$

if score(adj) > 0. If score (adj) < 0, it returns

$$f_{VS}(adv, adj) = score(adj) + (5\text{-}score(adj)) * score(adv). \quad \ldots (4.8)$$

Description of Algorithm:

If an adverb belongs to affirmative or strongly intensifying adverb and score of adjective is greater than 0, *i.e.*, adjective is positive then the final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.5).

For example, suppose score of positive adjective "good" is 3 and score of adverb "really" belonging to affirmative or strongly intensifying adverbs is 0.4. Then, final score of "really good" will be as given in (4.9).

$$\begin{aligned} \text{Score (really good)} &= \text{Score (good)} + (5\text{-score (good)})*\text{score (really)} \\ &= 3 + (5\text{-}3)*0.4 \\ &= 3.8 \end{aligned} \quad \ldots (4.9)$$

Similarly, if the score of adjective is less than 0, *i.e.*, adjective is negative then final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.6).

But, if an adverb belongs to weakly intensifying adverb or adverb of doubt and score of adjective is greater than 0, *i.e.*, adjective is positive then the final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.7).

For example, suppose score of adjective "good" is 3 and score of adverb "very" belonging to weak or doubt intensifying adverbs is 0.3. Then, final score of "very good" will be as given in (4.10).

$$\begin{aligned} \text{Score (very good)} &= \text{Score (good)} + (5\text{-score (good)})*\text{score (very)} \\ &= 3 + (5-3)*0.3 \\ &= 3.6 \end{aligned} \quad \ldots (4.10)$$

Similarly, if the score of adjective is less than 0, *i.e.,* adjective is negative then final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.8).

Thus, the score of "very good" is slightly lower than the score of "really good" because score of adverb "very" is less than the score of adverb "really".

4.1.3.1 Adjective Priority Scoring Algorithm (APS)

The algorithm 4.2 gives priority to adjectives over the adverbs and alters the score of the adjective by weight ***r***. Then this weight ***r*** determines the extent to which an adverb affects the score of an adjective. It denotes the importance of adverb compared to an adjective that it modifies. The larger the value of ***r***, the greater is the effect of the adverb [1].

Algorithm 4.2: Adjective Priority Scoring Algorithm

If adv $\in$ AFF $\cup$ STRONG, then:

$$f_{APS}^{r}(adv, adj) = \min(5, score(adj) + r * score(adv)). \quad \ldots (4.11)$$

if score(adj) > 0. If score (adj) < 0,

$$f_{APS}^{r}(adv, adj) = \min(5, score(adj) - r * score(adv)). \quad \ldots (4.12)$$

If adv $\in$ WEAK $\cup$ DOUBT, APS^{r} reverses the above and returns

$$f_{APS}^{r}(adv, adj) = \max(0, score(adj) - r * score(adv)). \quad \ldots (4.13)$$

if score (adj) > 0. If score (adj) < 0, it returns

$$f_{APS}^{r}(adv, adj) = \max(0, score(adj) + r * score(adv)) \quad \ldots (4.14)$$

Description of Algorithm:

If an adverb belongs to affirmative or strongly intensifying adverb and score of adjective is greater than 0, *i.e.,* adjective is positive then the final score of the Adverb-Adjective

combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.11).

For example, consider the weight ***r*** between 0 and 1, *i.e.*, 0.1. An adverb "really" belonging to affirmative or strongly intensifying adverb has a score of 0.4 and positive adjective "good" has a score of 3. Then, final score of "really good" according to adjective priority algorithm is as given in (4.15).

$$\begin{aligned} \text{Score of (really good)} &= \min(5, \text{score (good)} + r * \text{score (really)}) \\ &= \min(5, 3+0.1*0.4) \\ &= \min(5, 3.04) \\ &= 3.04 \end{aligned} \quad \ldots (4.15)$$

Similarly, if the score of adjective is less than 0, *i.e.*, adjective is negative then final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.12).

If an adverb belongs to weakly intensifying adverb or adverb of doubt and score of adjective is greater than 0, *i.e.*, adjective is positive then the final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.13).

For example, consider the weight ***r*** between 0 and 1, *i.e.*, 0.1. An adverb "very" belonging to weakly intensifying adverb or adverb of doubt has a score of 0.3 and positive adjective "good" has a score of 3. Then, final score of "very good" according to adjective priority algorithm is as given in (4.16).

$$\begin{aligned} \text{Score of (very good)} &= \min(5, \text{score (good)} + r * \text{score (very)}) \\ &= \min(5, 3+0.1*0.3) \\ &= \min(5, 3.03) \\ &= 3.03 \end{aligned} \quad \ldots (4.16)$$

Similarly, if the score of adjective is less than 0, *i.e.*, adjective is negative then final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.14).

Thus, the final score of "very good" is less than the score of "really goo

4.1.3.3 Adverb Priority Scoring Algorithm (AdvPS)

The algorithm 4.3 is also similar to previous algorithm 4.2 except the parameter weight r is applied to adjective rather than adverb [1].

Algorithm 4.3: Adverb Priority Scoring Algorithm

If adv ∈ AFF ∪ STRONG, then:

$$f_{AdvPS}^{r}(adv, adj) = \min(5, score(adv) + r * score(adj)). \quad \ldots (4.17)$$

if score (adj) > 0. If score (adj) < 0,

$$f_{AdvPS}^{r}(adv, adj) = \max(0, score(adv) - r * score(adj)). \quad \ldots (4.18)$$

If adv ∈ WEAK ∪ DOUBT, then:

$$f_{AdvPS}^{r}(adv, adj) = \max(0, score(adv) - r * score(adj)). \quad \ldots (4.19)$$

if score (adj) > 0. If score (adj) < 0,

$$f_{AdvPS}^{r}(adv, adj) = \min(5, score(adv) + r * score(adj)). \quad \ldots (4.20)$$

Description of Algorithm:

If an adverb belongs to affirmative or strongly intensifying adverb and score of adjective is greater than 0, *i.e.,* adjective is positive then the final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.17).

For example, consider the weight ***r*** between 0 and 1, *i.e.,* 0.1. An adverb "really" belonging to affirmative or strongly intensifying adverb has a score of 0.4 and positive adjective "good" has a score of 3. Then, final score of "really good" according to adverb priority algorithm is as given in (4.21).

$$\begin{aligned} \text{Score (really good)} &= \min(5, score(really) + r * score(good)) \\ &= \min(5, 0.4+0.1*3) \\ &= \min(5, 0.7) \\ &= 0.7 \end{aligned} \quad \ldots (4.21)$$

So, final score of "really good" is 0.7.

Similarly, if the score of adjective is less than 0, *i.e.*, adjective is negative then final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.18).

If an adverb belongs to weakly intensifying adverb or adverb of doubt and score of adjective is greater than 0, *i.e.*, adjective is positive then the final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.19).

For example, consider the weight ***r*** between 0 and 1, *i.e.*, 0.1. An adverb "very" belonging to weakly intensifying adverb or adverb of doubt has a score of 0.3 and positive adjective "good" has a score of 3. Then, final score of "very good" according to adverb priority algorithm is as given in (4.22).

Score (very good) = max (0, score (very) + r * score good))

= max (0, 0.3+0.1*3)

= max (0, 0.6)

= 0.6 … (4.22)

Similarly, if the score of adjective is less than 0, *i.e.*, adjective is negative then final score of the Adverb-Adjective combination is the score of the adjective that is accordingly modified with the effect of the adverb as given in (4.20).

Thus, the final score of "very good" is less than the score of "really good".

4.1.3.4 Resulting Sentiment Computing Algorithm

The algorithm 4.4 assigns the sentiment to text on the basis of final score calculated. The sentiment assigned to text can be positive, moderately positive, highly positive, negative, moderately negative, highly negative or neutral.

Description of Algorithm:

The final sentiment to text is assigned according to algorithm 4.4. According to this algorithm, if final score is greater than 0 but less than or equal to 0.3 then text represents positive sentiment. But, if final score is greater than 0.3 but less than or equal to 0.5 then

text represents moderately positive sentiment. Otherwise, if final score is greater than 0.5 then text represents highly positive sentiment.

Algorithm 4.4: Resulting Sentiment Computing Algorithm

```
If score > 0:
   If score <=0.3:
         then text represents positive sentiment,
   else if (score > 0.3) and (score <=0.5):
         then text represents moderately positive sentiment,
   else:
         then text represents highly positive sentiment,
else if score < 0:
    If score >=-0.3:
         then text represents  negative sentiment,
   else if (score < -0.3) and (score >=-0.5):
         then text represents moderately negative sentiment,
   else:
         then text represents highly negative sentiment,
else:
     the text represents no sentiment or neutral sentiment.
```

Similarly, if final score is less than 0 and also greater than or equal to -0.3 then text represents negative sentiment. But, if final score is less than -0.3 and greater than – 0.5 then text represents moderately negative sentiment. Otherwise, text represents highly positive sentiment.

But, if final score is equal to 0 then text represents no sentiment or neutral sentiment.

4.2 Process for Calculation of Final Sentiment and Score

As shown in Figure 4.2, first of all simple English text is tokenized by system. Tokenization process splits the text into tokens. Then, pattern-matching process takes place. In this process, tokens are matched with adjective and adverb score files.

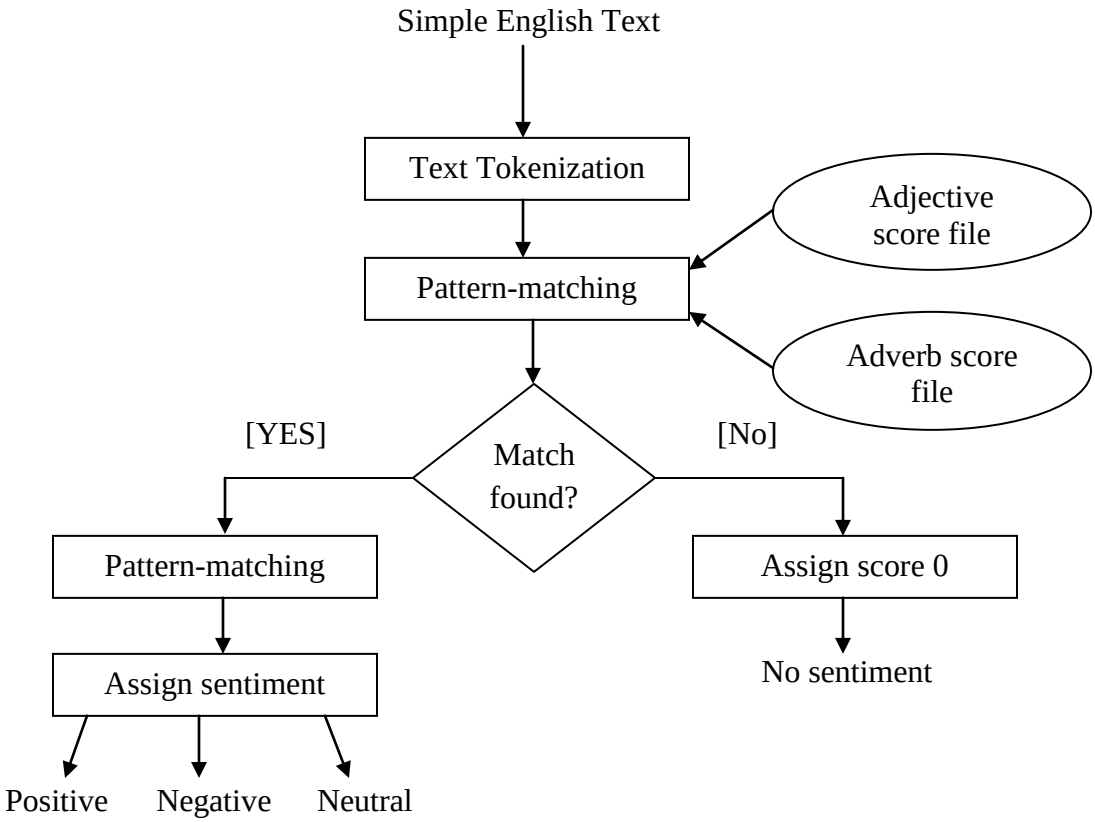

Fig. 4.2: Process for Computing Score and Sentiment

The scores of adjectives are assigned between -5 and 5 and scores of adverbs are assigned between 0 and 0.5. If any of the token matches with terms of adjective and adverb score files then their corresponding scores are computed. And final score of text is computed using algorithms given in (4.1), (4.2) and (4.3). Then, final sentiment is assigned to text as positive, negative or neutral using algorithm 4.4. But, if any of the token in the text does not matches with terms of adjective and adverb score files. Then, 0 score is assigned to that text and finally no sentiment are assigned to that text.

For example: *He is very bad boy.* … (4.23)

By tokenization process, sentence given in (4.23) is splitted into tokens as "He", "is", "very", "bad" and "boy". Now, these tokens are matched with adjective and adverb score files. Here, "very" and "bad" are found in adverb and adjective score files respectively. So, corresponding scores of "very" and "bad" are assigned as 0.3 and -3 respectively. And then final score and sentiment is computed by system using algorithm 4.4.

4.3 Role of Python in Sentiment Analysis

Python is a powerful programming language and easy to learn. It has high-level data structures that are very effective and a simple but effective approach to OOPS. Python's refined syntax and dynamic typing, makes it an ideal language for scripting on most platforms. New functions and data types that are implemented in C or C++ are also extended with interpreter of Python [42]. As Python is an extension language so it is suitable for personalized applications.

Python plays an important role in sentiment analysis. There are number of modules in python to access Internet and processing Internet protocols. One of them is *urllib2* which is used for retrieving data from URLs and other is *smtplib* that is used for sending mail. Python also provides us a special module named as *oauth2*. *OAuth* stands for Open Authentication Protocol. This protocol acts on a user's behalf and do things to a website. Before *OAuth*, user name and password of the person was required to give this access. When application sends a request on the user's interest, it gives user's credentials to website and then makes the request. Security is the major problem with this approach. It became a real concern when sites state that password will not be shared for other than their intended purposes were breaking their part of the Terms of Service. But with *OAuth*, there is no need to share username and password with an application. Instead, the application switches to the actual site (such as Twitter) and asks for credentials, which the application never sees. The site sends the access token to application which it uses to make requests on user's behalf. The important thing about this approach is that the endpoint site (Twitter) also stores this access token and account can be logged in on the endpoint and access token is revoked, thereby disallowing the application from making any further requests on user's behalf–without even touching the application. *JSON* (JavaScript Object Notation) module is also provided by Python. As the data returned by many sites is in *json* format so *json.loads* function of *json* takes the *json* as a string and returns data in Python structure [42].

Python provides various modules for development of graphical user interface (GUI), *e.g., Tkinter*, *wxPython* and *JPython*. Rule based sentiment analysis system uses *Tkinter* for the sentiment analysis of twitter posts and English documents. Python when used with *Tkinter* helps in creating GUI applications. *Tkinter* gives a powerful and object-oriented

interface to the *Tk* GUI toolkit. It also provides various controls for GUI application, such as buttons, labels and text boxes. Python provides 15 types of controls in *Tkinter* [42].

4.4 Sentiment Analysis of Twitter Posts

Twitter is a micro-blogging site. It enables users to send updates or tweets in the form of messages. These tweets can be sent to a group of friends or followers. Twitter has become a research area for sentiment classification or opinion mining due to its integration with many web-based applications. On the basis of opinion reflected, a tweet can be classified as highly positive, moderately positive, positive, highly negative, moderately negative, negative or neutral [8].

4.4.1 Extraction of Twitter Posts

To extract the posts from Twitter, install the *oauth2* library for authentication and *urllib2* for fetching URLs (Uniform Resource Locators) as given in (4.24) and (4.25). *Urlopen* function of *urllib2* provides us a capability of fetching urls using a variety of different protocols.

import oauth2 as oauth ... (4.24)

import urllib2 as urllib … (4.25)

Then, a twitter account on site *https://dev.twitter.com/apps* to access the data has been created. On twitter account that is created, an application by putting a dummy website to generate access token key, access token secret, consumer key and consumer secret for authentication of twitter data is created. Copy all these credentials. Token class of module *OAuth* is an *OAuth* credential used to request authorization. Tokens in *OAuth* consist of a "key" and a "secret". To identify the tokens that are used, the key is included in requests but the secret is used only in the signature, to prove that the server has given the token to requester. Consumer class of an *Oauth* module is a consumer of *OAuth*-protected services. The *OAuth* consumer is a "third-party" service which on behalf of an end user wants to access protected resources from an *OAuth* service provider. It's kind of the *OAuth* client. The consumer must be registered with the service provider. As part of that process, the service provider gives the consumer a "key" and a "secret" with which the consumer software can identify itself to the service. The consumer will include its key in

each request to identify it, but will use its secret only when signing requests, to prove that the request is from that specific registered consumer. A new request signature *octet* string is generated; the Service Provider verifies the request and compares it to the signature provided by the Consumer. By using the request parameters that are provided by the Consumer, signature is generated. The Consumer Secret and Token Secret are stored by the Service Provider. *HTTPHandler* and *HTTPSHandler* classes of module *urllib2* are used to handle the fetching of *HTTP* URLs and *HTTPS* URLs respectively [42]. The GUI of the sentiment analysis system is shown in Figure 4.3.

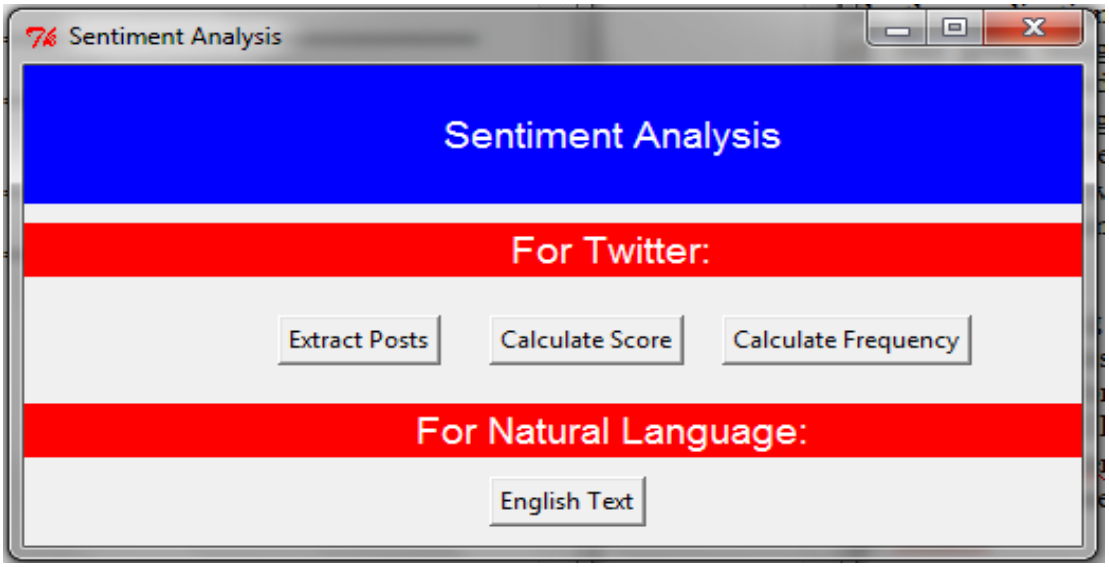

Fig. 4.3: GUI of Sentiment Analysis System

When "Extract Post" button on main window is clicked then a new window appears and asks for the parameter as shown in Figure 4.4 to enter for which twitter posts will be extracted.

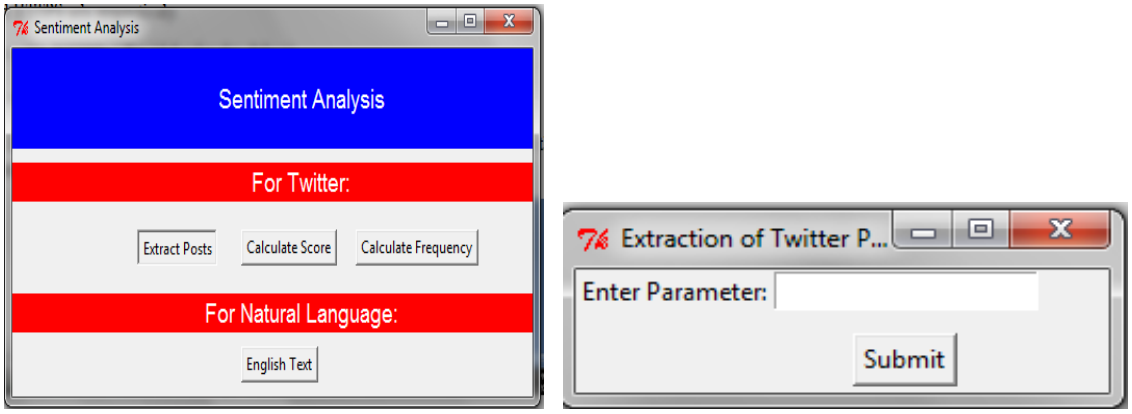

Fig. 4.4: Clicking on "Extract Post" Button

Suppose, parameter "BJP" is entered and clicked on button Submit. This parameter is passed to URL which gives the data in *json* format. When submit button is clicked, program asks for file name at Python Shell to which extracted tweets are stored as shown in Figure 4.5.

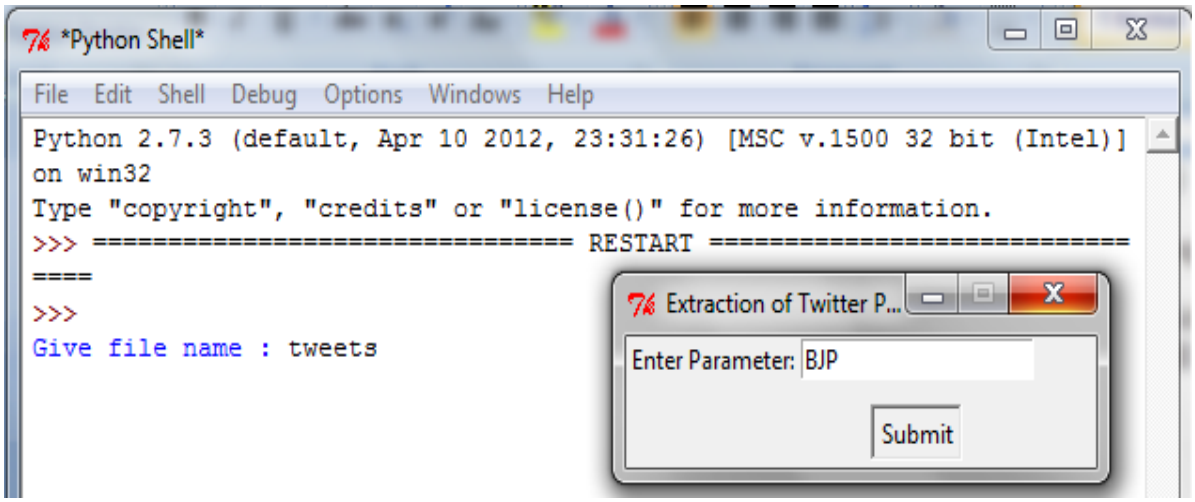

Fig. 4.5: Enter Parameter and File Name

Thus, a tweet file "tweets.txt" is shown in Figure 4.6.

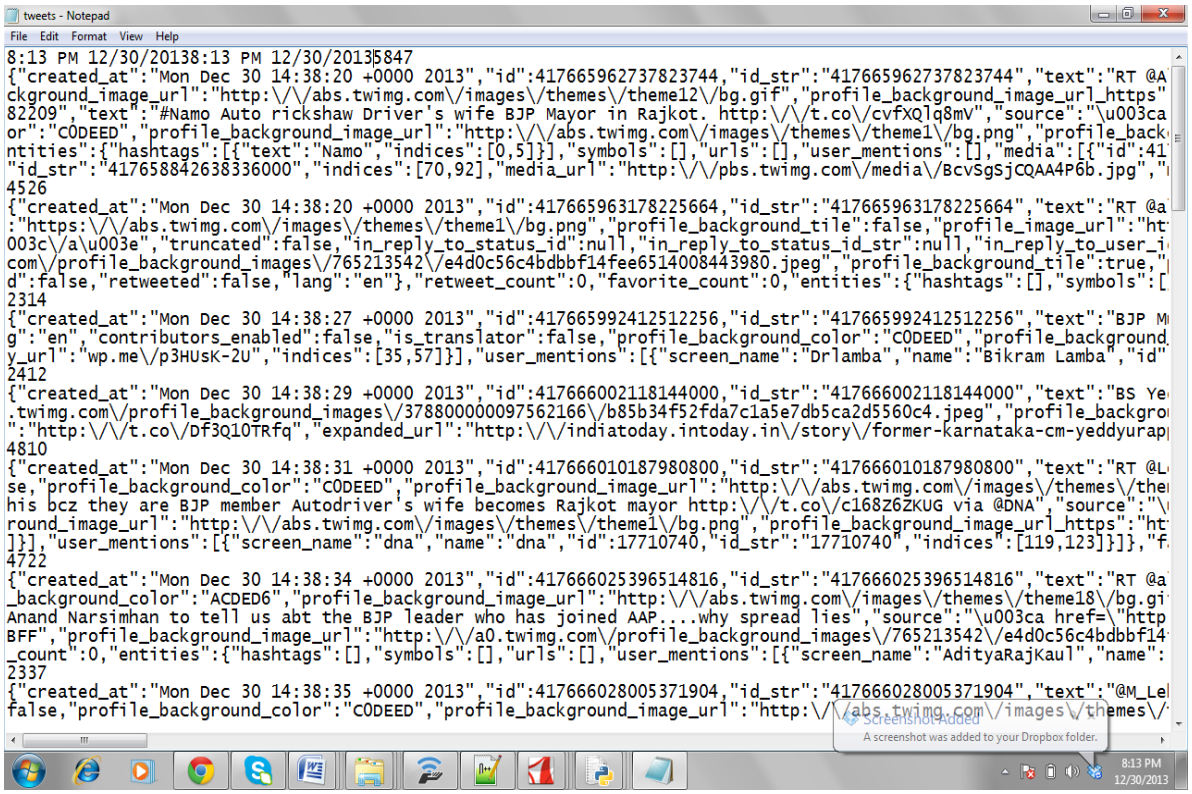

Fig. 4.6: Generated Tweet File

4.4.2 Role of Score File

The score file contains the list of pre-computed sentiment scores as shown in Table 4.3. Each line of the score file contains a word or phrase along with its sentiment score. If a word or phrase which is found in text but not found in score file then word or phrase is given a sentiment score of 0. The file format of score file is TAB limited means that term and score are separated by TAB character. Then, the sentiment of text on the basis of sentiment scores of the terms in the text is computed. The sentiment of text is equal to the sum of the sentiment scores for each term in the text.

Table 4.3: Score File

Term	Score	Term	Score
sad	-2	Better	2
safe	1	Bitter	-2
rainy	-1	Charming	3
rash	-2	Confused	-2
degrade	-2	Enjoy	2
delay	-1	Exploit	-2
commit	1	Faithful	3
accept	1	fearing	-2
accomplish	2	fun	4
Attack	-1	guilt	-3
best	3	ability	2
cancel	-1	die	-3
inadequate	-2	broke	-1
lazy	-1	favor	2
pretty	1	joke	2

4.4.3 Computation of Term Frequency

To calculate the frequency of each term in tweet, tweet file is passed as argument to program. So, when the button “Calculate Frequency” on main window is clicked, it asks for selecting the tweet file as shown in Figure 4.7.

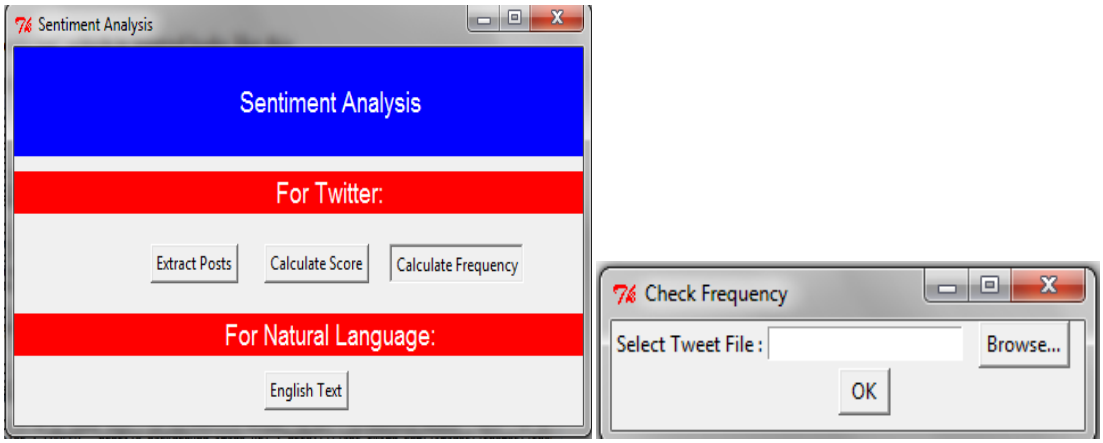

Fig. 4.7: Clicking on "Calculate Frequency" button

When the path of tweet file is selected and "OK" button is clicked. Then, it returns the each term of tweet file with its score as shown in Figure 4.8.

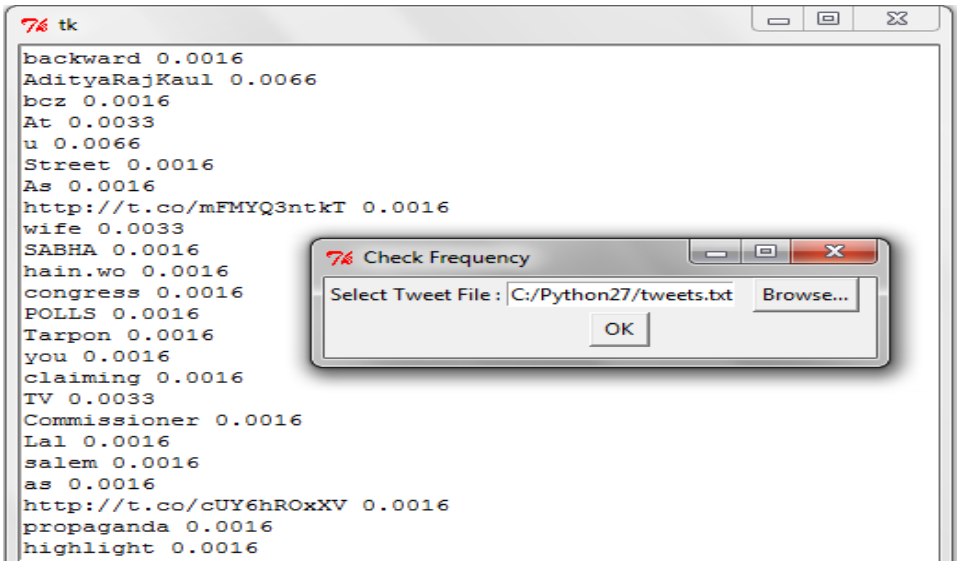

Fig. 4.8: Select Tweet File and Get Term Frequency

4.4.4 Sentiment Analysis of Tweets

To do the sentiment analysis of tweets, import two libraries *sys* and *json* as given in (4.26) and (4.27). Most of sites return data in *json* format. So *json.loads* function of *json* library takes *json* string and returns data in python structure.

import sys … (4.26)

import json … (4.27)

Now, to calculate sentiment and score of tweets extracted in "tweets.txt" file, click on "Calculate Score" button on main window, then a new window will appear as shown in Figure 4.9. This window prompts to select the tweet file and score file.

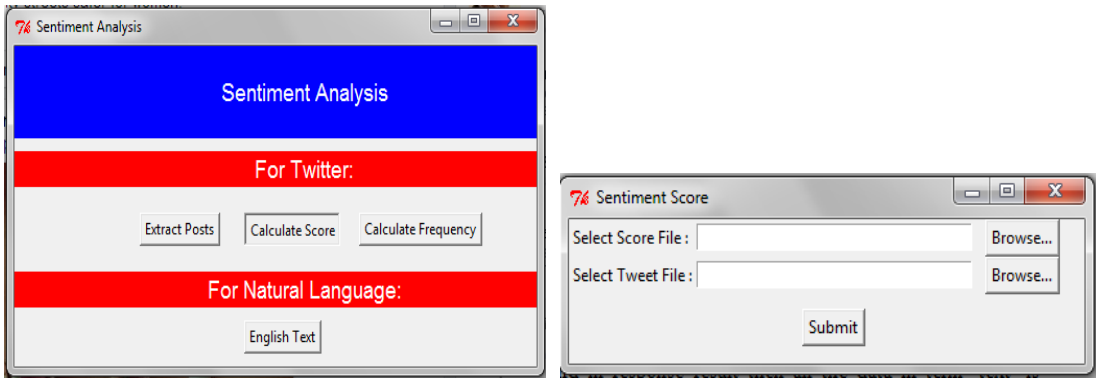

Fig. 4.9: Clicking on "Calculate Score" Button

Then, the path of score file and tweet file is selected through browsing by using "Browse" buttons. Both of these score file and tweet file are passed as arguments to program. When the "Submit" button is clicked, it shows the output about sentiment and score of tweets as shown in Figure 4.10.

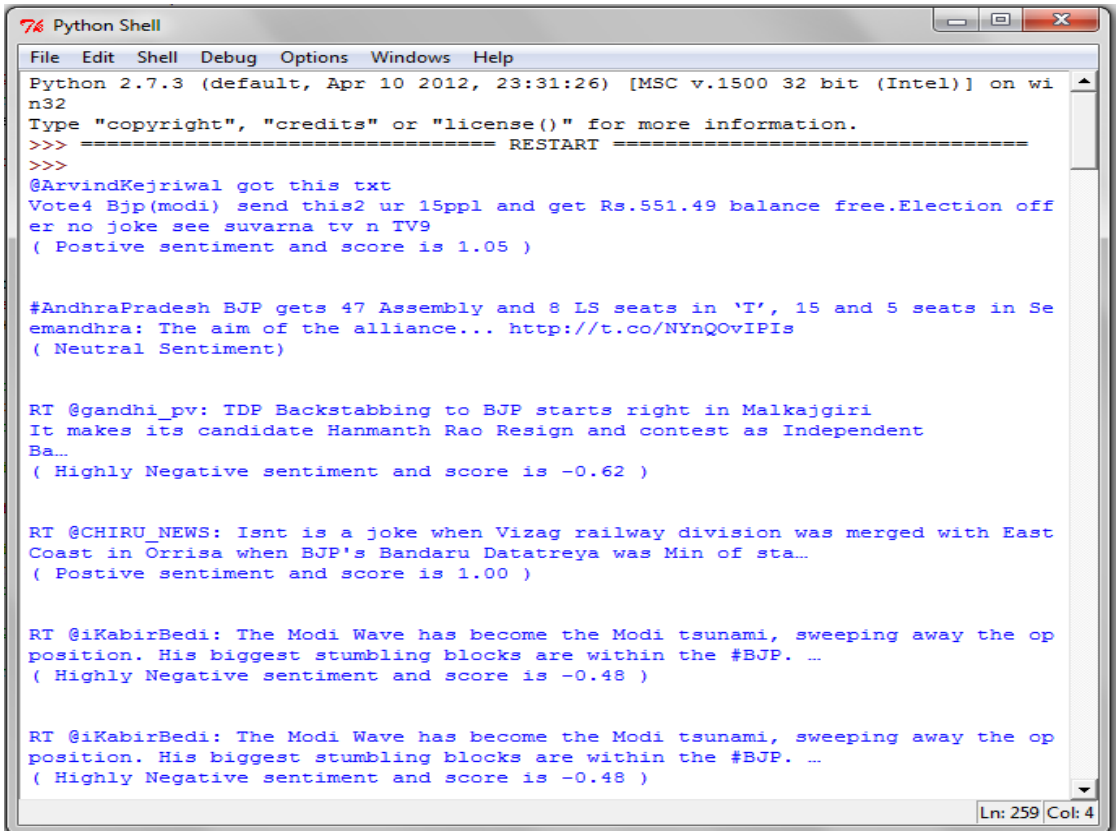

Fig. 4.10: Score and Sentiment of each Tweet

4.5 Sentiment Analysis of English Sentences

The system also takes simple English sentences as input. By using rules and algorithms for AACs, it computes score and sentiment of sentences. For example, when on main window, "English Text" button is clicked. It prompts a new window and sentence given in (4.28) is entered as shown in Figure 4.11.

Example Sentence: *He is not a good boy.* … (4.28)

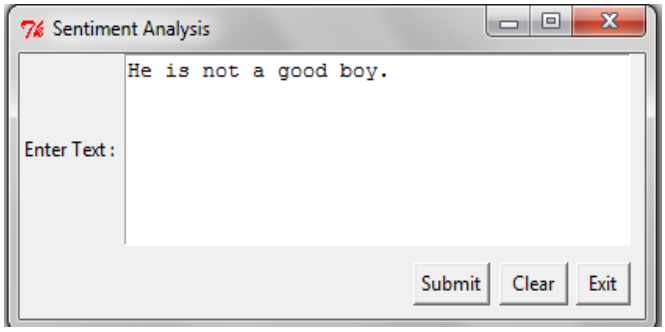

Fig. 4.11: Example of English Sentence given in (4.28)

On clicking the button "submit", it gives the overall sentiment and score of that sentence as shown in Figure 4.12.

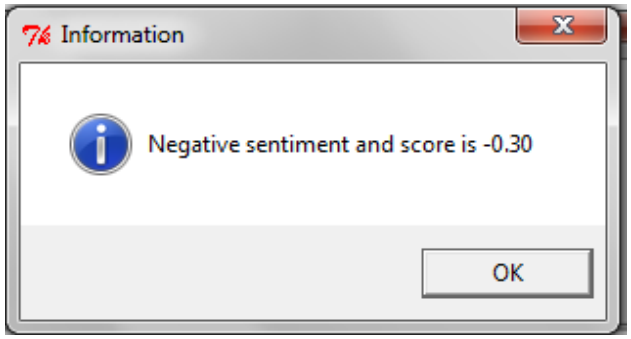

Fig. 4.12: Sentiment and Score of Sentence given in (4.28)

4.6 Sentiment Analysis of Natural Text using UNL

In the sentiment analysis process, identification of entity in the sentence is important. Entity identification is a difficult task without using UNL. But, it becomes easy to identify the entity in the sentence with UNL. Because, UNL provides different type of relations like "agt (agent)", "obj (object)" and "man (manner)" *etc*. which help in identification of entity and object *etc.* in the given sentence.

For example, English sentence to illustrate the concept of sentiment analysis is given in (4.29) and its corresponding UNL is given in (4.30).

English Sentence: *He killed her.* … (4.29)

Corresponding UNL to English sentence:

obj (kill.@past, 00.@3.@female)

agt(kill.@past, 00.@3.@male) … (4.30)

In the given UNL, "obj" and "agt" are relations. Here, "kill" is the first universal word and "00" is the second universal word which represents third person. "@past", "@3", "@female" and "@male" are attributes.

Algorithm 4.5: Computation of sentiment and score using UNL

Step 1: Parsing of UNL by the system.

Step 2: Identification of relations, universal words and attributes of universal words from UNL.

Step 3: Assigning score to sentiment bearing universal words and their attributes using score files.

Step 4: Calculation of final score.

Step 5: Assignment of sentiment to sentence according to relations identified as follows.

If final score > 0 and relation is "agt",

Then sentiment is positive for second universal word of relation "agt".

If final score < 0 and relation is "agt",

Then sentiment is negative for second universal word and positive for second universal word of relation "obj".

If final score is 0,

Then sentence is objective.

Description of Algorithm:

First of all, UNL is parsed by system. For example, system parses the UNL given in (4.30) and splits the UNL into tokens. In second step, system identifies the scope, relation, universal words and their attributes from tokens splitted in first step as shown in Table 4.4.

Table 4.4: Tokens of UNL given in (4.30)

Scope	Relation	First universal word	Attributes of first universal word	Second universal word	Attributes of second universal word
00	obj	kill	@past	00	@3.@female
00	agt	kill	@past	00	@3.@male

In third step, score is assigned, *e.g.*, score is assigned to first universal word "kill" here. In fourth step, final score of whole sentence is computed and then sentiment is assigned to sentence according to fifth step.

For example, the system processes the example sentence given in (4.29) as shown in Figure 4.13 and its output is shown in Figure 4.14.

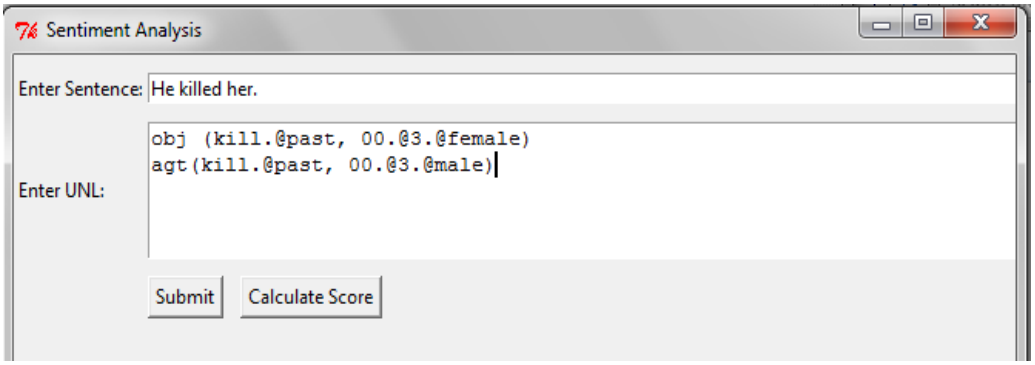

Fig.4.13: Example of English Sentence given in (4.29)

The output shows the score of the sentence and for which sentiment is negative or positive. As in the given example, Male person had killed female person. So, sentiment is negative for "Male Person" and positive for "Female Person".

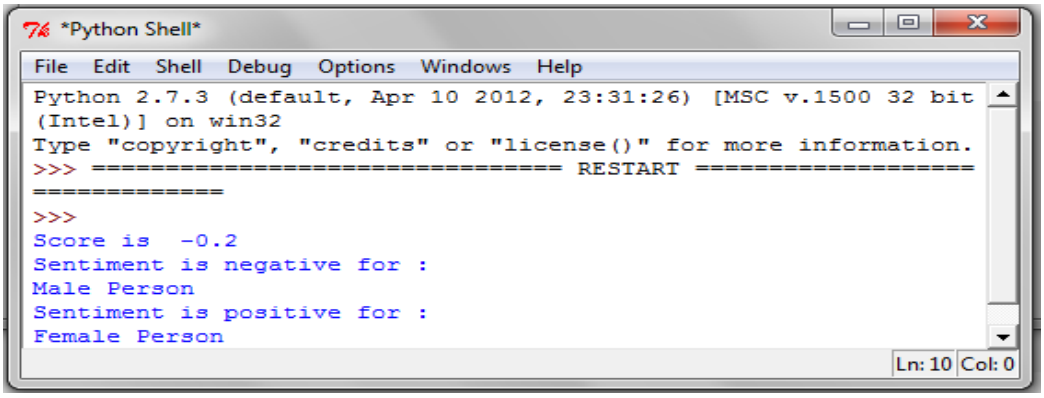

Fig.4.14: Output of Example Sentence given in (4.29)

Punjabi sentence to illustrate the concept of sentiment analysis is given in (4.31) and its corresponding UNL is given in (4.32).

Punjabi Sentence**:** *ਇੱਕ ਬਹੁਤ ਸੋਹਣੀ ਕਾਰ* … (4.31)

Corresponding Transliteration**:** *ikk bahut sōhṇī kār*

Corresponding UNL of Punjabi sentence**:**

mod(car.@indef, beautiful.@plus) … (4.32)

In the given UNL, "mod" is the relation. "car" and "beautiful" are first and second universal words respectively. "@indef" and "@plus" are attributes corresponding to first and second universal words respectively.

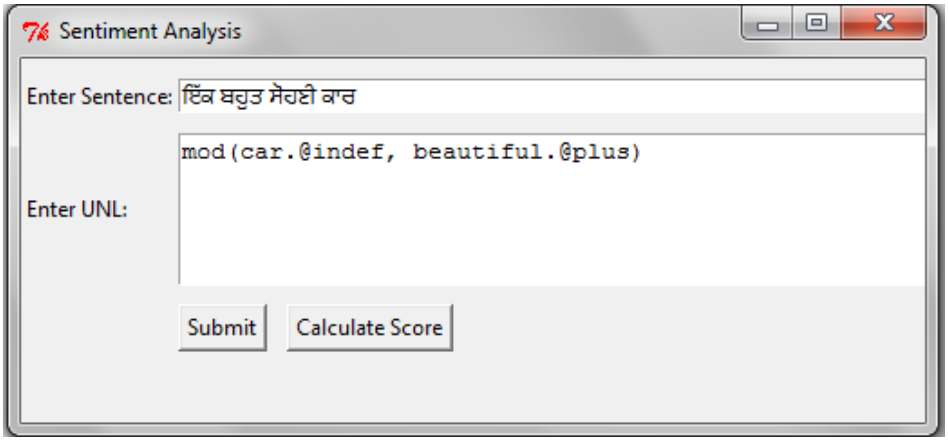

Fig. 4.15: Example of Punjabi Sentence given in (4.31)

The UNL is parsed by system as shown in Figure 4.15 and output is shown in Figure 4.16.

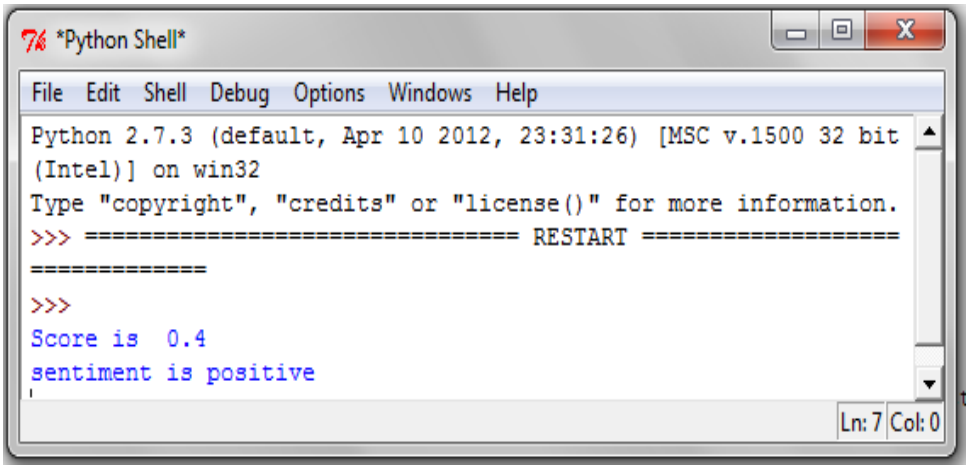

Fig. 4.16: Output of Punjabi Example Sentence given in (4.31)

4.7 Handling of Challenges of Sentiment Analysis

As there are many challenges in sentiments analysis. So, there was need to overcome these challenges to improve sentiment analysis process. Some of these challenges are solved by creating different databases files for these special cases as follows.

4.7.1 Domain Dependency

As polarity of words changes from domain to domain, therefore separate database files have been created for these domains to handle it as discussed in chapter 1. English sentences to illustrate the concept of domain dependency are given in (4.33) and (4.34).

The story of the movie was unpredictable. … (4.33)

The steering of the bus is unpredictable. … (4.34)

The results after handling the English sentence given in (4.33) and (4.34) are shown in Figure 4.17.

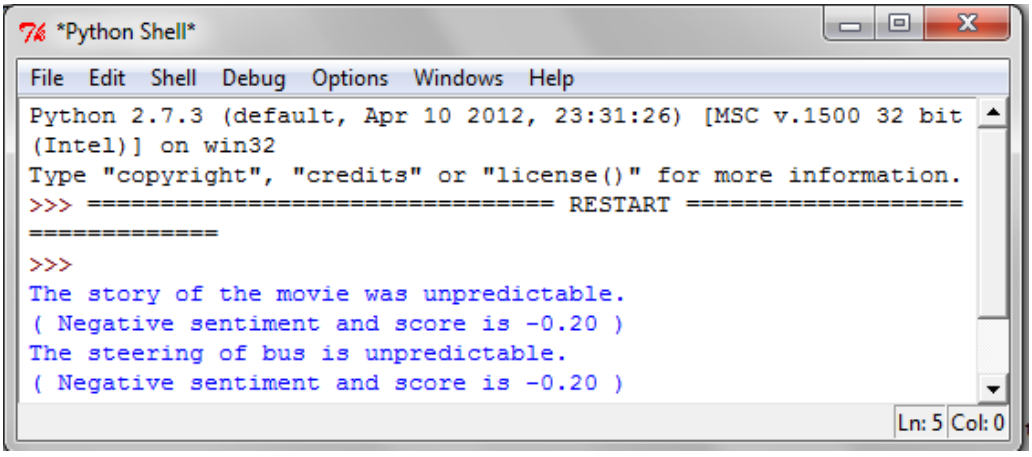

Fig. 4.17: Results after handling "Domain Dependency"

In the sentence given in (4.33), the sentiment depicted is positive but the sentiment depicted in the sentence given in (4.34) is negative.

4.7.2 Thwarted Expectations

In case of thwarted expectations, sometimes sentiment of whole text or review is present at its end. So, this challenge can be resolved by analyzing last one or two sentences of any review. English text to illustrate the concept of thwarted expectations is given in (4.35).

This film should be awesome. There is too much fun, the actors are first class, the supporting cast is fine as well and Hrithik Roshan is attempting to deliver an excellent performance. … (4.35)

As shown in Figure 4.18, the last sentence depicts the positive sentiment about movie review domain. So, overall sentiment will be positive.

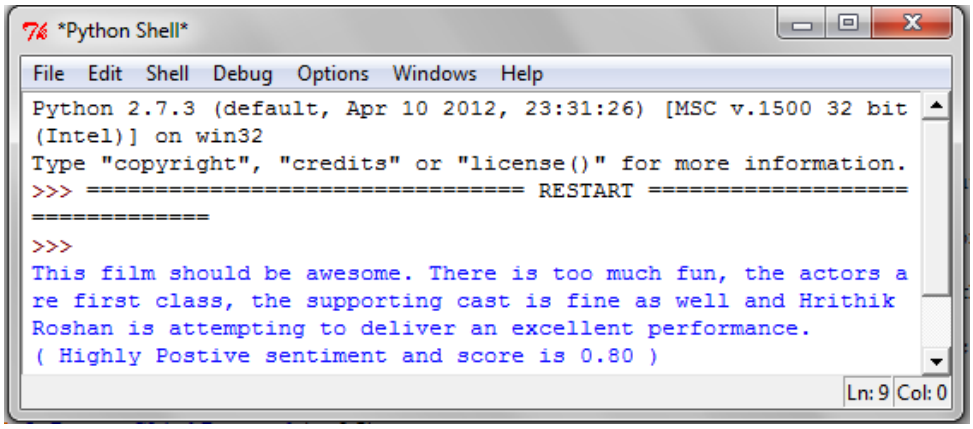

Fig. 4.18: (a) Results after handling "Thwarted Expectations" with Positive Sentiment

This film should be awesome. There is too much fun, the actors are first class, the supporting cast is fine as well and Hrithik Roshan is attempting to deliver an excellent performance. Still, it can not hold up. … (4.36)

The results of the movie review given in (4.36) are shown in Figure 4.18(b). In this review, the last sentence depicts the negative sentiment about movie review domain. So, resultant sentiment will be negative.

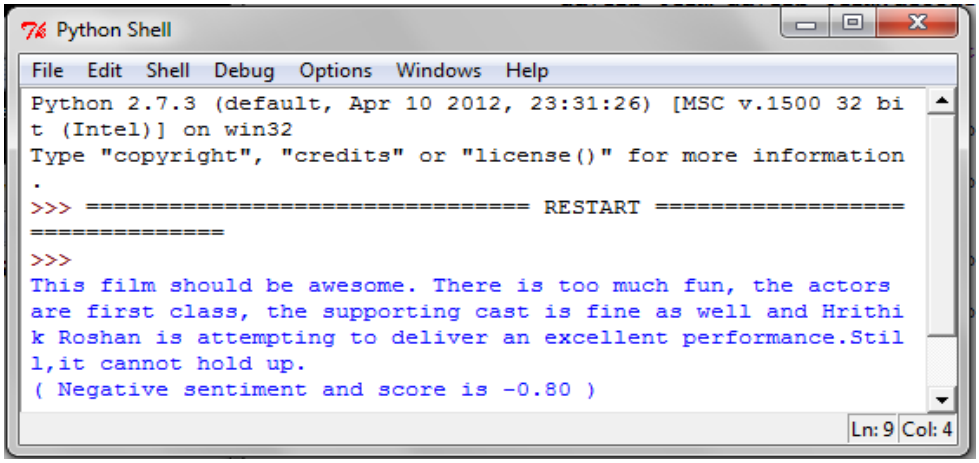

Fig. 4.18: (b) Results after handling "Thwarted Expectations" with Negative Sentiment

4.7.3 Pragmatics

Pragmatics of user opinion (such as capitalization of words *etc.*) changes the sentiment of words. English sentences to illustrate the concept of pragmatics are given in (4.37) and (4.38).

I have just completed watching Barca DESTROY Ac Milan. ... (4.37)

That final totally destroyed me. … (4.38)

As discussed in chapter 1, capitalization in the sentence given in (4.37) is used with delicacy to denote sentiment and presents a positive sentiment but the sentence given in (4.38) represents a negative sentiment.

The results of the sentences given in (4.37) and (4.38) after resolving this challenge are shown in Figure 4.19.

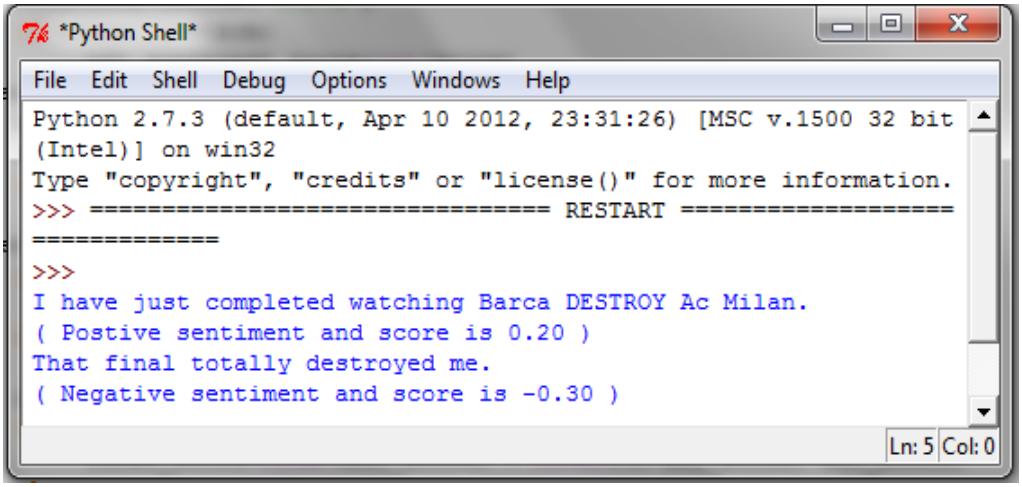

Fig. 4.19: Results after handling "Pragmatics"

4.7.4 World Knowledge

World knowledge is also important for detecting sentiments. Separate database files have been created to handle this challenge also. English sentences to illustrate the concept of world knowledge are given in (4.39) and (4.40).

He is a Frankenstein. … (4.39)

He has just completed Doctor Zhivago for the first time and overall conclusion is that Russia sucks. … (4.40)

As discussed in chapter 1, one has to be familiar about Frankenstein and Doctor Zhivago to determine the sentiment of sentences given in (4.39) and (4.40). After resolving this

challenge, sentences given in (4.39) presents negative sentiment and sentence given in (4.40) presents a positive sentiment as shown in Figure 4.20.

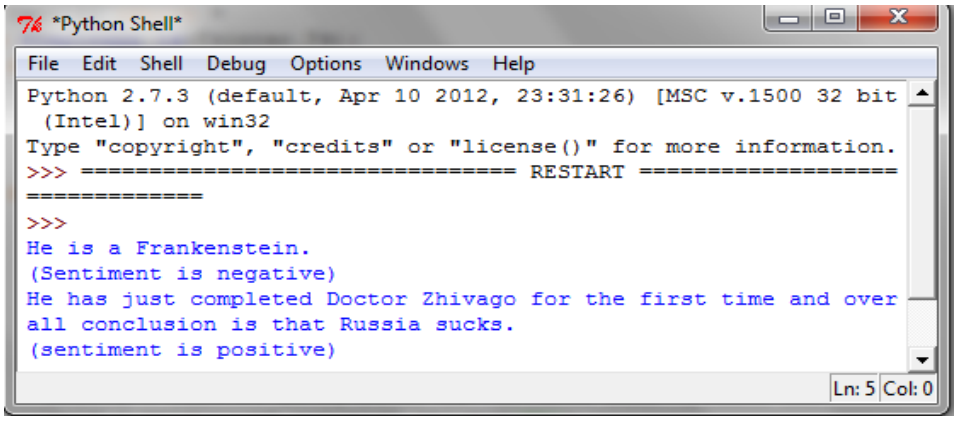

Fig. 4.20: Results after handling "World Knowledge"

4.7.5 Entity Identification

Identification of an entity is also an important task to perform sentiment analysis process as discussed in chapter 1. There may be multiple entities in a text or a sentence. English sentences to illustrate the concept of entity identification are given in (4.41) and (4.42).

Blackberry is better than Micromax. … (4.41)

Raman defeated Hitansh in cricket. … (4.42)

After resolving this challenge, sentences given in (4.41) and (4.42) depict positive sentiment for "Blackberry" and negative for "Hitansh" respectively as shown in Figure 4.21.

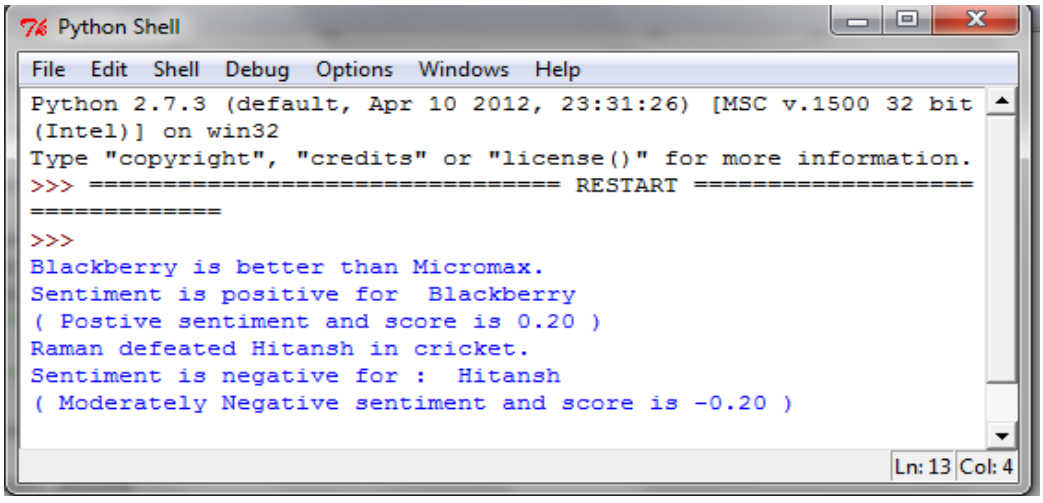

Fig. 4.21: Results after handling "Entity Identification"

This challenge can also be resolved very easily using UNL of sentences. Because, UNL provides relations like “agt (agent)”, “man (manner)”, “obj (object)”, which help in identification of entity in a sentence.

Chapter Summary

In this chapter, different rules on the basis of type of adverbs of degree and scoring algorithms for AACs are discussed. A rule based sentiment analysis system has also been explained in this chapter. The system performs the sentiment analysis of twitter posts and English documents. It extracts the twitter posts from twitter by user defined parameter using twitter APIs and computes the frequency of each term in tweet. Then, it calculates the sentiment the score of each tweet using rule set and scoring algorithms. This system also calculates the sentiment and score of English sentences on the basis of AACs. To resolve the challenges like entity identification, it uses UNL for performing sentiment analysis. Some of the challenges which are resolved by creating special database files are also included in this chapter.

Chapter 5

Results and Discussions

5.1 Testing of Tweets

The proposed system has been tested on tweets extracted from twitter about "Arvind Kejriwal". The sentiment, score of tweets and overall sentiment about "Arvind Kejriwal" is shown in Figure 5.1.

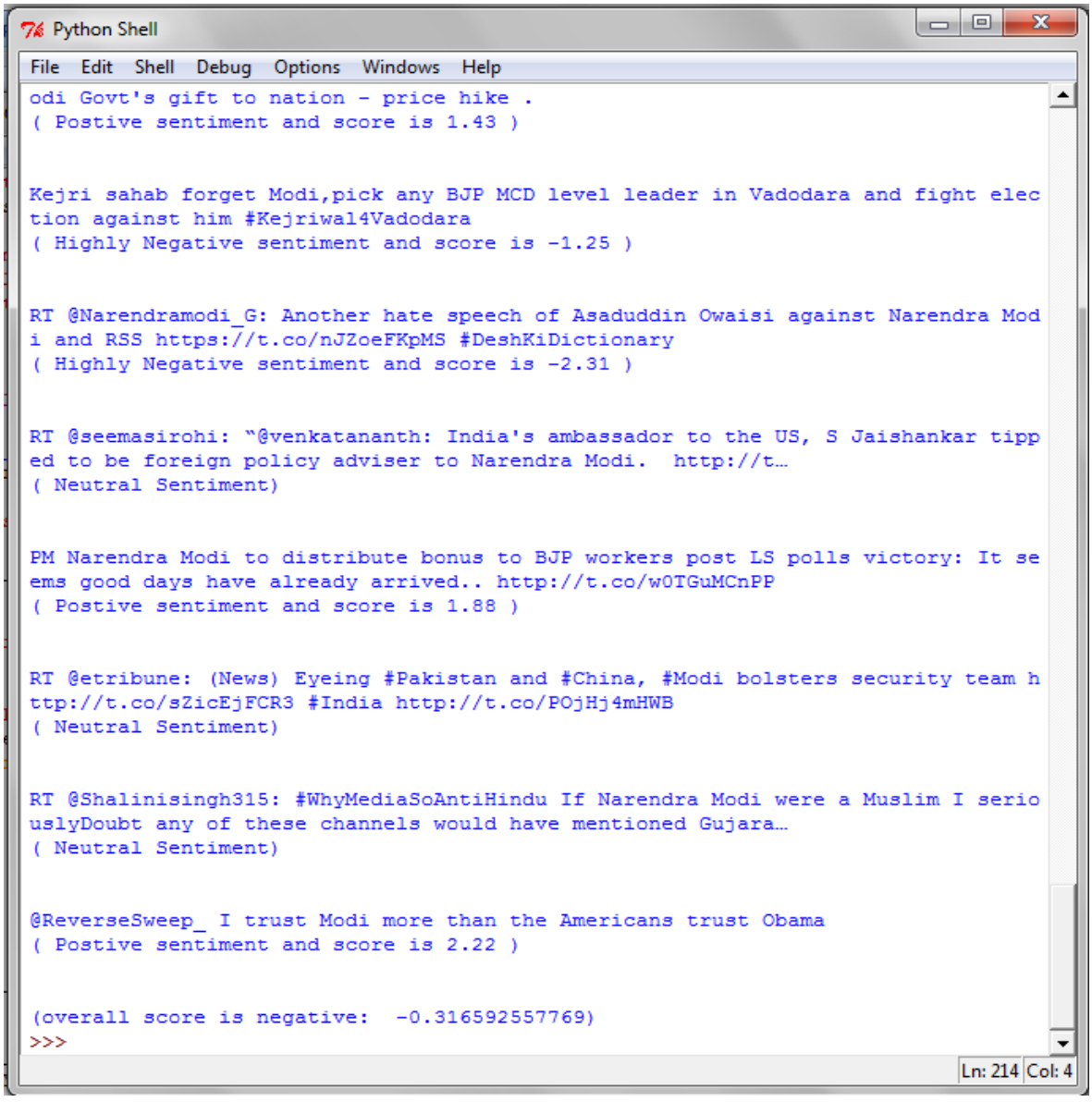

Fig 5.1: Results of Tweets about "Arvind Kejriwal"

5.2 Testing of Simple English Sentences

The rule based sentiment analysis system has been tested on 100 simple English sentences containing adjectives and adverbs. The proposed system uses AACs in sentences, so better accuracy has been achieved by system in comparison to adjectives only. Out of these 100 sentences, the results of 45 sentences given by the proposed system are shown in Table 5.1.

Table 5.1: Sentiment and Score of Simple English Sentences given by Proposed System

Sr. No.	Sentence	Sentiment	Score
1.	Ram is a good boy.	Positive	0.30
2.	Ram is not a good boy.	Negative	-0.30
3.	Meeting maybe postponed.	Positive	0.02
4.	Ram is a bad boy.	Negative	-0.30
5.	Ram is a very bad boy.	Highly negative	-0.54
6.	Luckily I cleared the GATE exam in 2012.	Positive	0.10
7.	Sushma is a very beautiful girl.	Moderately Positive	0.36
8.	This work is hardly good to do.	Positive	0.30
9.	Everybody was praising her dress in the party.	Positive	0.30
10.	She is too beautiful girl.	Moderately Positive	0.38
11.	Her childish nature impresses everybody.	Positive	0.10
12.	Naveen always uses abused words while talking.	Negative	-0.30
13.	The movie was awesome.	Moderately Positive	0.40
14.	He cannot pay his fees because his father is very poor.	Highly Negative	-0.54
15.	He defeated me in Candy crush game.	Negative	-0.30
16.	I am very determined to achieve goals in my life.	Positive	0.29
17.	Her beauty distracts everyone.	Negative	-0.20

Table 5.1: Sentiment and Score of Simple English Sentences given by Proposed System

Sr. No.	Sentence	Sentiment	Score
18.	Sita is confident girl.	Positive	0.20
19.	Katrina is famous for her elegant smile.	Moderately Positive	0.40
20.	Sunita betrayed me.	Negative	-0.30
21.	I am feeling burdened due to too much work in the office.	Highly Negative	-0.34
22.	Tina misbehaved with her teacher during class.	Negative	-0.20
23.	An extraordinary miracle happened in her life.	Moderately Positive	0.40
24.	She always remains confused about her future.	Negative	-0.20
25.	He is a very big liar.	Highly Negative	-0.41
26.	I like him too much.	Moderately Positive	0.32
27.	Everyone likes her because of her loving nature.	Moderately Positive	0.40
28.	I was very nervous during interview.	Highly Negative	-0.41
29.	Palvi is very moody regarding her studies.	Moderately negative	-0.28
30.	I wished her good luck for the exam.	Highly positive	0.60
31.	Rupinder always give natural pose while photography.	Positive	0.10
32.	He is my best buddy.	Positive	0.30
33.	I am feeling tension free after exams.	Positive	0.10
34.	Coward always blames others.	Negative	-0.20
35.	Charanpreet's voice is very bold.	Positive	0.29
36.	Aurangzeb was an evil king.	Negative	-0.30
37.	I am fan of Shahrukh Khan.	Positive	0.30

Table 5.1: Sentiment and Score of Simple English Sentences given by Proposed System

Sr. No.	Sentence	Sentiment	Score
38.	Karina Kapoor looks fantastic in sari.	Moderately positive	0.40
39.	She blocked him from Facebook.	Negative	-0.10
40.	I am getting bored from studies now.	Negative	-0.20
41.	He is fond of sweets.	Positive	0.20
42.	She is very fun loving girl.	Highly positive	0.57
43.	They greeted me very well when I reached at their home.	Positive	0.29
44.	I am feeling guilty for my mistake.	Highly Negative	-0.40
45.	She always feels jealous with me.	Negative	-0.20

5.3 Testing of English Sentences of UC-A1 Corpus

The proposed system has been tested on English sentences of UC-A1 corpus given by UNDL foundation. The UC-A1 corpus consists of 100 UNL sentences containing different parts of speech. The parts of speech contained in this corpus are given in Table 5.2.

Table 5.2: Parts of Speech contained by UC-A1 Corpus

Type	Number of sentences
Temporary entries	5
Determiners	10
Possessive determiners	10
Prepositions	20
Conjunctions	10
Noun phrase structure	20
Verb forms	10
Sentence structures	15
Total	**100**

The sentiment and score of 20 UNL sentences assigned by proposed system is shown in Table 5.3.

Table 5.3: Testing of UC-A1 Corpus

Sr. No.	Sentence	UNL	Sentiment	Score
1.	beautiful car	{unl} mod(car, beautiful) {/unl}	Positive	0.3
2.	the new beautiful expensive car	{unl} mod(car.@def, beautiful) mod(car.@def, expensive) mod(car.@def, new) {/unl}	Highly Positive	0.9
3.	many new beautiful expensive cars	{unl} mod(car.@multal, beautiful) mod(car.@multal, expensive) mod(car.@multal, new) {/unl}	Highly Positive	0.9
4.	the beautiful glass mug	{unl} mod(mug.@def, glass) mod(mug.@def, beautiful) {/unl}	Positive	0.3
5.	the new beautiful expensive car of John	{unl} mod(car.@def, John) mod(car.@def, beautiful) mod(car.@def, expensive) mod(car.@def, new) {/unl}	Highly Positive	0.9
6.	an extremely beautiful car	{unl} mod(car.@indef, beautiful) man(beautiful, extremely) {/unl}	Highly Positive	0.6

Table 5.3: Testing of UC-A1 Corpus

Sr. No.	Sentence	UNL	Sentiment	Score
7.	a new beautiful car from Switzerland and a very expensive book for John	{unl} and(:02, :01) ben:02(book.@indef, John.@for) mod:02(book.@indef, expensive.@plus) plc:01(car.@indef, Switzerland:0B.@from) mod:01(car.@indef, beautiful) mod:01(car.@indef, new) {/unl}	Highly Positive	1.0
8.	John breaks the rules.	{unl} agt(break(icl>do).@entry, "John") obj(break(icl>do).@entry, rule.@generic.@pl) {/unl}	Negative for : "John" Positive for : Rule	-0.3
9.	The explosion broke the windows.	{unl} agt(break(icl>do).@entry.@past, explosion(icl>event).@def) obj(break(icl>do).@entry.@past, window.@def.@pl) {/unl}	Negative for : Explosion Positive for : Window	-0.3
10.	The leaf is red.	{unl} aoj(red(aoj>thing).@entry, leaf(icl>thing).@def) {/unl}	Objective	0
11.	It is not beautiful.	{unl} aoj(beautiful.@not.@present, 00.@3) {/unl}	Negative	-0.3

Table 5.3: Testing of UC-A1 Corpus

Sr. No.	Sentence	UNL	Sentiment	Score
12.	The child runs while crying.	{unl} agt(run(icl>do).@entry.@present, child(icl>person).@def) coo(run(icl>do).@entry.@present, cry(icl>do).@past) {/unl}	Negative for : Child	-0.1
13.	She is very beautiful.	{unl} aoj(beautiful(aoj>thing).@entry, she) man(beautiful(aoj>thing).@entry, very) {/unl}	Moderately Positive	0.4
14.	I won the prize in fair competition.	{unl} agt(win(icl>do).@entry.@past, i(icl>person).@def) obj(win(icl>do).@entry.@past, prize.@def) scn(win(icl>do).@entry.@past, competition(icl>event)) mod(competition(icl>event), fair) {/unl}	Positive	0.3
15.	He killed her with a knife in the kitchen yesterday because of Peter.	{unl} agt(kill.@past, 00.@3.@male) obj(kill.@past, 00.@3.@female) man(kill.@past, knife.@indef.@with) plc(kill.@past, kitchen.@def) tim(kill.@past, yesterday) rsn(kill.@past, Peter) {/unl}	Negative for : Male Person Positive for : Female Person	-0.2

Table 5.3: Testing of UC-A1 Corpus

Sr. No.	Sentence	UNL	Sentiment	Score
16.	The train for London was cancelled.	{unl} obj(cancel(icl>ocurr).@entry.@past, train(icl>thing).@def) to(train(icl>thing).@def, london(icl>city)) {/unl}	Negative	-0.1
17.	John died with Mary.	{unl} cob(die(icl>occur).@entry.@past, "Mary") obj(die(icl>occur).@entry.@past, "John") {/unl}	Negative	-0.3
18.	He looks sad since this morning.	{unl} obj(look(icl>ocurr).@entry, i(icl>person)) man(look(icl>ocurr).@entry, sad) tmf(look(icl>ocurr).@entry, morning(icl>time)) tmf(morning(icl>time), this) {/unl}	Negative	-0.2
19.	He killed himself.	{unl} agt(kill.@past.@reflexive, 00.@3.@male) {/unl}	Negative for : Male Person	-0.2
20.	Ram killed Mary.	{unl} obj(kill.@past, Mary) agt(kill.@past, Ram) {/unl}	Negative for : Ram Positive for : Mary	-0.2

5.4 Comparison of Results of Proposed System with Manual Testing

The proposed system has been tested on 200 simple English sentences. These sentences are also manually tested by 3 users. Each sentence is classified as positive, negative, moderately positive, moderately negative, highly positive, highly negative and objective by each user. The final sentiment of each sentence is taken as average of sentiments assigned by each one of the user. The results of polarity of sentiment given by system and by manual testing are shown in Table 5.4. It has been observed that system has identified sentiments for 194 sentences out of 200 sentences. So, system has correctly identified the sentiments of 188 sentences, *i.e.*, the sentiments of 188 sentences assigned by system agree with sentiments assigned through manual testing. Thus, an accuracy of 74% has been achieved by system.

Table 5.4: Comparison of Results of Proposed System with Manual Testing

Polarity of sentence	**Number of sentences assigned to corresponding polarity by proposed system**	**Number of sentences assigned to corresponding polarity during manually testing**	**Number of sentences whose score given by proposed system agrees with manual testing**
Positive	40	36	36
Moderately positive	14	16	14
Highly positive	20	24	20
Negative	44	48	44
Moderately negative	4	6	4
Highly negative	6	4	4
Objective	66	66	66
Total	**194**	**200**	**188**

Chapter Summary

In this chapter, results of testing on tweets of "Arvind Kejriwal", 45 simple English sentences and 20 UNL sentences given by proposed system are shown. Also, comparison of results of proposed system with manual testing is included in this chapter.

Chapter 6

Conclusion and Future Scope

6.1 Conclusion

Sentiment analysis is the process which helps in identifying people's attitudes and emotional states. The feelings of the people can be expressed in positive or negative ways. Mostly, parts of speech are used as feature to extract the sentiment of the text. From parts of speech, only adjectives play an important role in indentifying sentiment. But, when adverbs are used with adjectives then sentiment of text get altered. It is also important to use Adverb-Adjective combination as a feature to do sentiment analysis. So, a rule based sentiment analysis system has been purposed. In this system, Adverb-Adjective combinations are taken into account for performing sentiment analysis as it gives better results than adjectives only. The purposed system helps in computing the sentiment and score of tweets or English documents.

To do the sentiment analysis of tweets, the proposed system first extracts the twitter posts from twitter about any parameter entered by user. The system can also computes the frequency of each term in tweet. At last, it calculates score and sentiment of each tweet using rule set and scoring algorithms for AACs.

The system computes the sentiment and score of simple English sentences. These sentences are splitted into tokens by tokenization process. Then, with the help of adjective and adverb score files; sentiment and score of these sentences is found. Also, some challenges of sentiment analysis like entity identification, pragmatics, world knowledge *etc.* have been resolved by creating special database files for special cases. The system also calculates the sentiment and score of sentences using UNL. Because, UNL provides different type of relations like "agt (agent)", "obj (object)" and "man (manner)" *etc*. which help in identification of entity, identification of object and detection of subjectivity in a sentence.

The system has been tested on 100 simple English sentences and 100 English sentences of UC-A1 corpus given by UNDL foundation. It has been observed that system has identified sentiments for 194 sentences out of 200 sentences. So, system has correctly

identified the sentiments of 188 sentences, *i.e.*, the sentiments of 188 sentences assigned by system agree with sentiments assigned through manual testing. Thus, an accuracy of 74% has been achieved by the system.

6.2 Limitations and Future Scope

Some of the limitations of the proposed system and future scope to resolve these limitations are given as follows.

- The proposed system works for simple sentences only. It can be extended to work for complex sentences also.
- Adjective and adverb score files can be improved by adding more sentiment bearing words to it.
- The proposed system does not use any parser. So, parser can be embedded into system in future to improve sentiment analysis.
- As there should be availability of UNL for performing sentiment analysis. So in future, UNL can be made available by calling APIs of IAN.
- The system works for simple universal words while using UNL for sentiment analysis. It can be extended to work for compound universal words also for resolving challenges of sentiment analysis.
- The proposed system is not web based, so it can be extended as web-based application in future.

References

[1] Mukherjee S, Bhattacharyya P 2013 Sentiment Analysis: A Literature Survey, IIT Bombay, Mumbai.

[2] Pang B, Lee L, Vaithyanathan S 2002 Thumbs up? Sentiment classification using machine learning techniques. *Proc. ACL-02 Conf. on Empirical methods in natural language processing,* 10: 79-86.

[3] Turney P D 2002 Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *Proc. 40th Annual Meeting on Association for Computational Linguistics (ACL),* Philadelphia, 417-424.

[4] Karamibekr M, Ghorbani A A 2012 Sentiment analysis of social issues. *Int. Conf. on Social Informatics,* 215-221.

[5] Pang B, Lee L 2008 Opinion mining and sentiment analysis. *Foundations and trends in information retrieval,* 2(1-2):1-135.

[6] Vohra S, Teraiya J 2013 Applications and Challenges for Sentiment Analysis: A Survey. *Int. Journal of Engineering Research & Technology (IJERT),* 2(2):1-5.

[7] Cambria E 2013 An Introduction to Concept-Level Sentiment Analysis. *Advances in Soft Computing and Its Applications*, *Springer Berlin Heidelberg*, 478-483.

[8] Gebremeskel G 2011 Sentiment Analysis of Twitter posts about news. Master's Thesis, University of Malta.

[9] Cambria E, Schuller B, Xia Y, Havasi C 2013 New avenues in opinion mining and sentiment analysis. *IEEE Intelligent Systems,* 28(2):15-21.

[10] Cui H, Mittal V, Datar M 2006 Comparative experiments on sentiment classification for online product reviews. *American Association for Artificial Intelligence*, 6:1265-1270.

[11] Lewicki P, Hill T 2006 Statistics: methods and applications. *Tulsa, OK. Statsoft*.

[12] Dave K, Lawrence S, Pennock D M 2003 Mining the peanut gallery: Opinion extraction and semantic classification of product reviews. *Proc. 12th Int. Conf. on World Wide Web*, 519-528.

[13] Pang B, Lee L 2004 A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. *Proc. 42nd Annual Meeting on Association for Computational Linguistics*, no. 271.

[14] Chen C, Ibekwe-SanJuan F, SanJuan E, Weaver C 2006 Visual analysis of conflicting opinions. *IEEE Symposium on Visual Analytics Science and Technology*, Baltimore, Maryland: United States, 59-66.

[15] Boiy E, Hens P, Deschacht K, Moens M F 2007 Automatic Sentiment Analysis in On-line Text. *Proc. ELPUB2007 Conference on Electronic Publishing,* Vienna, Austria, 349-360.

[16] Annett M, Kondrak G 2008 A comparison of sentiment analysis techniques: Polarizing movie blogs. *Advances in Artificial Intelligence, Springer Berlin Heidelberg*, 25-35.

[17] Ye Q, Zhang Z, Law R 2009 Sentiment classification of online reviews to travel destinations by supervised machine learning approaches. *Expert Systems with Applications,* 36(3):6527-6535.

[18] Paltoglou G, Thelwall M 2010 A study of information retrieval weighting schemes for sentiment analysis. *Proc.48th Annual Meeting of the Association for Computational Linguistics*, 1386-1395.

[19] Bair E 2013 Semi-supervised clustering methods. *Wiley Interdisciplinary Reviews: Computational Statistics*, 5(5):349-361.

[20] Yang Y, Pedersen J O 1997 A comparative study on feature selection in text categorization." *Int. Conf. on Machine Learning*, 97:412-420.

[21] Peng T C, Shih C C 2010 An Unsupervised Snippet-based Sentiment Classification Method for Chinese Unknown Phrases without using Reference Word Pairs. *IEEE/WIC/ACM Int. Conf. on Web Intelligence and Intelligent Agent Technology*, 3: 243-248.

[22] Li G, Liu F 2010 A clustering-based approach on sentiment analysis. *Int. Conf. on Intelligent Systems and Knowledge Engineering (ISKE),* 331-337.

[23] Sharma A, Dey S 2012 Performance Investigation of Feature Selection Methods and Sentiment Lexicons for Sentiment Analysis. *Special Issue of Int. Journal of Computer*

Applications on Advanced Computing and Communication Technologies for HPC Applications - ACCTHPCA, 3:15-20.

[24] Prabowo R, Thelwall M 2009 Sentiment analysis: A combined approach. *Journal of Informetrics*, 3(2):43-157.

[25] Witten I H, Frank E 2005 Data Mining: Practical machine learning tools and techniques, Morgan Kaufmann.

[26] Tomokiyo M, Chollet G 2003 VoiceUNL: a proposal to represent speech control mechanisms within the Universal Networking Digital Language. *Proc. Int. Conf. on the Convergence of Knowledge, Culture, Language and Information Technologies*, Alexandria, Egypt, 1-6.

[27] Surve M, Singh S, Kagathara S, Venkatasivaramasastry K, Dubey S, Rane G, Saraswati J, Badodekar S, Iyer A, Almeida A, Nikam R, Perez C G, Bhattacharyya P 2004 AgroExplorer: a meaning based multilingual search engine. *Proc. Int. Conf. on Digital Libraries (ICDL)*, Delhi, India, 1-13.

[28] Mukerjee A, Achla M R, Kumar K, Goyal P, Shukla P 2003 Universal Networking Language: a tool for language independent semantics? *Proc. Int. Conf. on the Convergence of Knowledge, Culture, Language and Information Technologies*, Egypt, 145-154.

[29] Bértoli L, Luz R P, Bastos R C 2003 A web platform using UNL: CELTA's showcase. *Proc. Int. Conf. on the Convergence of Knowledge, Culture, Language and Information Technologies*, Egypt, 276-285.

[30] Hajlaoui N, Boitet C 2005 A Pivot XML-based architecture for multilingual, multiversion documents: parallel monolingual documents aligned through a central correspondence descriptor and possible use of UNL. *Universal Network Language: Advances in Theory and Applications, Ed(s) Cardeñosa J, Gelbukh A, Tovar E*, México, Research on Computing Science, 309-325.

[31] Cardeñosa J, Gallardo C, Iraola L 2005 An XML-UNL model for knowledge based annotation." *Universal Network Language: Advances in Theory and Applications, Ed(s) Cardeñosa J, Gelbukh A, Tovar E*, México, Research on Computing Science, 300-308.

[32] Montesco C E, Moreira D A 2005 UCL-universal communication language. *Universal Network Language: Advances in Theory and Applications, Ed(s) Cardeñosa J, Gelbukh A, Tovar E*, México, Research on Computing Science, 326-336.

[33] Ramamritham K, Bahuman A, Duttagupta S 2006 aAqua: a database-backended multilingual, multimedia community forum. *Proc. Int. Conf. on Management of Data*, Chicago, USA, 784-786.

[34] Alansary S, Nagi M, Adly N 2006 Towards a language-independent Universal Digital Library. *Proc. 2nd Int. Conf. on Universal Digital Library,* Alexandria, Egypt, 1-10.

[35] Karande J B 2007 Multilingual search engine: implementation using UNL. *Proc. Int. Conf. on Semantic Web and Digital Libraries*, Bangalore, India, 1-7.

[36] Avetisyan A, Avetisyan V 2010 LOOK4: Enhancement of web search results with Universal Words and WordNet. *Proc. 5th Int. Conf. on Global WordNet,* Mumbai, India, 1-5.

[37] UNDL Foundation 2010, [Online] Available: http://www.undl.org.

[38] Uchida H, Zhu M 2001 The Universal Networking Language beyond Machine Translation. *Proc. Int. Symposium on Language in Cyberspace*, Seoul, Korea, 1-15.

[39] Uchida H, Zhu M 2005 UNL2005 for providing knowledge infrastructure. *Proc. Semantic Computing Workshop*, Chiba, Japan, 1-12.

[40] Uchida H, Zhu M 1993 Interlingua for multilingual machine translation. *Proc. 4th MT Summit*, Japan, 157-169.

[41] Cardeñosa J, Gallardo C, Tovar E 2003 Standardization of the generation process in a multilingual environment. *Proc. Int. Conf. on the Convergence of Knowledge, Culture, Language and Information Technologies*, Egypt, 10-24.

[42] An Introduction to Python v2.7.7 2012. [Online] Available: https://docs.python.org/2/.

Research Publications

Research Paper Published

Sujata Rani and Parteek Kumar, "Challenges of Sentiment Analysis and Existing State of Art" in *International Journal of Innovation and Research in Computer Science (IJIRCS),* special issue 2014.

Research Paper Accepted

Sujata Rani and Parteek Kumar, "Rule Based Sentiment Analysis System" in *Second Elsevier International Conference on Emerging Research in Computing, Information, Communication and Applications (ERCICA-2014),* NMIT, Bangalore, India.