

SENTIMENT ANALYSIS USING VALENCE ASSESSMENT APPROACH WITH CONJUNCTIONS

A DISSERTATION

*Submitted in partial fulfillment of the
requirements for the award of the degree
of*

INTEGRATED DUAL DEGREE

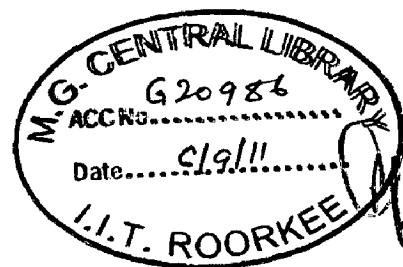
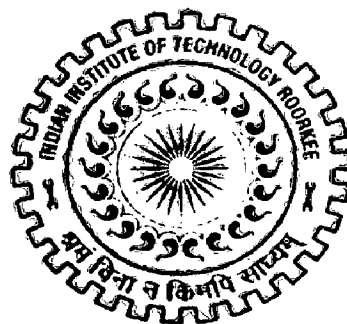
in

COMPUTER SCIENCE AND ENGINEERING

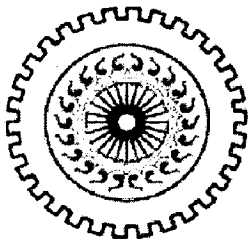
(With Specialization in Information Technology)

By

PRACHEER AGARWAL



**DEPARTMENT OF ELECTRONICS AND COMPUTER ENGINEERING
INDIAN INSTITUTE OF TECHNOLOGY ROORKEE
ROORKEE -247 667 (INDIA)
JUNE, 2011**



INDIAN INSTITUTE OF TECHNOLOGY
ROORKEE

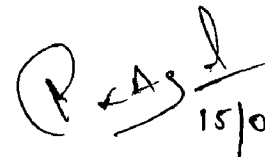
CANDIDATE'S DECLARATION

I hereby declare that the work is being presented in the dissertation work entitled "SENTIMENT ANALYSIS USING VALENCE ASSESSMENT APPROACH WITH CONJUNCTIONS" towards the partial fulfillment of the requirement for the award of the degree of **Integrated Dual Degree in Computer Science and Engineering with specialization in Information Technology**, submitted to the **Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, India** is an authentic record of my own work carried out during the period from May 2010 to June 2011 under the guidance and supervision of **Dr. R C Joshi, Professor, Department of Electronics and Computer Engineering of Indian Institute of Technology, Roorkee.**

The matter being presented in the Dissertation has not been submitted by me for the award of any other degree or diploma of this or any other Institute/University.

Date: June, 2011

Place: Roorkee



15/06/11

(PRACHEER AGARWAL)

CERTIFICATE

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

Date: June, 2011


(Dr. R C Joshi)
Supervisor
15/6

ACKNOWLEDGEMENTS

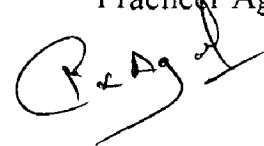
I would like to express my heartfelt gratitude to my learned mentor, Dr. R.C.Joshi, Professor, Department of Electronics and Computer Engineering, Indian Institute of Technology Roorkee, for his encouragement, able guidance, painstaking supervision and innovative suggestions. His capability to judge things beyond the written text has helped this dissertation take its present form. I feel privileged to have completed my dissertation under his supervision.

I have no words to thank my friend Bharat Kumar. He has been instrumental in shaping my approach towards accomplishing the task at hand. Also, I wish to convey my sincere thanks to all my classmates and friends. The time spent with them is unforgettable and has touched all aspects of my life. The help in manual testing had saved my lot of time. The late night interesting discussions has helped me in thinking out of the box for all the problems that I faced. Also each moment spent together is memorable.

Above all, I have no words to express my love for my family. I could not have reached this important milestone in my life without their constant support and encouragement. I take this opportunity to thank my parents for their unconditional love and motivation that shaped the confident individual in me.

I am thankful to the almighty for showering his grace that provided the courage and perseverance to overcome all obstacles that stood in my way.

Pracheer Agarwal



ABSTRACT

“What other people think” has always been an important piece of information for most of us during the decision-making process. Long before awareness of the World Wide Web became widespread, many of us asked our friends for suggestions. But the Internet and the Web have now made it possible to find out about the opinions and experiences of those in the vast pool of people that are neither our personal acquaintances nor well-known professional critics — that is, people we have never heard of. And conversely, more and more people are making their opinions available to strangers via the Internet. In recent years, there has been a rapid growth of web-content, especially on-line discussion groups, review sites and blogs. These are highly personal and typically express opinions. To organize this information, identification of sentiment polarity is very useful.

Sentiment analysis or opinion mining is a branch of natural language processing, computational linguistics and text mining. The main task in Sentiment Analysis is to find out the mood of writer or speaker with respect to some topic.

Most of the previous attempts to extract sentiment from sentence focused on the use of machine learning methods ignoring the importance of language analysis. We present an approach to find the hidden sentiment expressed in text at sentence level in the presence of conjunctions. Different approaches have been used to find sentiment, but none of those ever considered the conjunctions used in the sentence. We have formed a rule set for different conjunctions to join the sentiments expressed in different phrases of the sentence. Several experiments with datasets have been conducted. The experimental results shows significant performance gain over existing approaches.

CONTENTS

Candidate's Declaration	i
Acknowledgments	ii
Abstract	iii
Contents	iv
List of Figures	vi
List of Tables	vii
Chapter 1 Introduction and Problem Statement	1
1.1 Introduction.....	1
1.2 Motivation.....	2
1.3 Problem Statement.....	3
1.4 Thesis Organisation.....	4
Chapter 2 Background and Literature Survey	5
2.1 Features of Sentiment Analysis.....	6
2.1.1 Term Presence vs Frequency.....	6
2.1.2 Parts of Speech.....	6
2.1.3 Negation.....	7
2.2 Word Sentiment Classification.....	8
2.3 Machine Learning Techniques.....	10
2.4 Valence Assessment Approach.....	11
2.5 Research Gaps.....	13
Chapter 3 Proposed Technique for Sentiment Analysis	14
3.1 Framework for Sentiment Analysis.....	14
3.2 Text Pre-processing Module.....	16
3.3 Triplet Extraction Module.....	17
3.4 Knowledgebase Extension Module.....	20
3.5 Valence Assessment Module.....	21
3.6 Opinion Detection Module.....	24
3.6.1 Roles of Conjunctions.....	24
3.6.2 Analysis of Conjunctions.....	24
3.6.3 Conjunction Rule-set.....	25
Chapter 4 Implementation Details	29
4.1 NLP Tool Used – Stanford Parser.....	29

4.2	Linguistic Resources Used – WordNet.....	29
4.3	Design and development.....	31
4.3.1	Text Pre-processing Module.....	32
4.3.1	Triplet Extraction Module.....	33
4.3.2	Valence Assessment Module.....	34
4.3.3	Knowledgebase Extension Module.....	35
4.3.4	Opinion Detection Module.....	36
Chapter 5 Result and Discussion		37
5.1	Dataset used.....	37
5.1.1	Dataset for Triplet Extraction Module.....	37
5.1.2	Dataset for Opinion Detection Module.....	37
5.2	Performance of Triplet Extraction Module.....	38
5.3	Performance of Opinion Detection Module.....	42
5.4	Impact of different situation and parameters on algorithm.....	44
Chapter 6 Conclusion and Future Work		
6.1	Conclusion.....	45
6.2	Scope for future work.....	46
References		47

LIST OF FIGURES

Figure 3.1 Proposed Algorithm in Modular Structure 15
Figure 3.2: Parse tree generated for the sentence “A rare black squirrel has become a regular visitor to a suburban garden. 18
Figure 3.3: Functionality of Knowledgebase Extension Module 21
Figure 4.1: The parse tree generated by Stanford Parser 30
Figure 4.2: Function Calls in Text Pre-processing Module 32
Figure 4.3: Sequence of Functions Calls in triplet extraction Module 33
Figure 4.4: Sequence of functions calls in knowledgebase extension module34
Figure 4.5: Sequence of Calls and Returns in Valence Assessment Module 35
Figure 4.6: Function Calls in Opinion Detection Module 36
Figure 5.1: Triplet Similarity Measure 39
Figure 5.2: Percentage of Correct Triplets Generated : Expert Human vs Computer 41
Figure 5.3: Comparison of Triplet Production: Expert Human vs Computer 41
Figure 5.4: Accuracy of Various Approaches for Sentiment Analysis 43

LIST OF TABLES

Table 3.1: List of subtrees in which verbs are found and associated type of verb 18
Table 3.2: List of subtrees in which nouns are found and associated type of noun 19
Table 5.1: Input Datasets 38
Table 5.2: Triplet Extraction Module Testing Statistics 40
Table 5.3: Accuracy of different approaches of Sentiment Analysis 42
Table 5.4: Opinion Detection Module Testing Statistics 43

Chapter 1

Introduction and Problem Statement

“Motivation is what gets you started” – Jim Rohn

1.1 Introduction

An important part of our information-gathering behaviour has always been to find out what other people think. With the growing availability and popularity of opinion-rich resources such as online review sites and personal blogs, new opportunities and challenges arise. People now can actively use information technologies to seek out and understand the opinions of others. The sudden eruption of activity in the area of opinion mining and sentiment analysis, which deals with the computational treatment of opinion, sentiment, and subjectivity in text, has thus occurred at least in part as a direct response to the surge of interest in new systems that deal directly with opinions as a first-class object.

Sentiment analysis or opinion mining aims to determine the attitude of a speaker or a writer with respect to some topic. The basic task in sentiment analysis is classifying the *polarity* of a given text at the document, sentence, or feature/aspect level — whether the expressed opinion in a document, a sentence or an entity feature/aspect is positive, negative or neutral.

The rise of social media such as blogs and social networks has fuelled interest in sentiment analysis. As businesses look to automate the process of filtering out the noise, understanding the conversations, identifying the relevant content and actioning it appropriately, many are now looking to the field of sentiment analysis.

1.2 Motivation

The motivation for the study comes from the unique challenges offered in the varied application domain. In general terms the research aims to a linguistic approach to sentiment analysis. The motivation is to give computer programs a skill known as emotional intelligence with the ability to understand human emotion expressed in text. Application like empathic machine, online chat/e- mail clients, customer feedback/product review analysis, intelligent user interface, web-data mining etc. might benefit from this kind of research. We will now discuss the various application areas of Sentiment Analysis. The application domains are:

- Review-related websites - The same capabilities that a review-oriented search engine would have could also serve very well as the basis for the creation and automated upkeep of review- and opinion-aggregation websites. That is, as an alternative to sites like Epinions that solicit feedback and reviews, one could imagine sites that proactively gather such information. Topics need not be restricted to product reviews, but could include opinions about candidates running for office, political issues, and so forth.
- Business and Government Intelligence - Sentiment-analysis technologies for extracting opinions from unstructured human-authored documents would be excellent tools for handling many business-intelligence tasks [1]. Government intelligence is another application that has been considered. For example, it has been suggested that one could monitor sources for increases in hostile or negative communications [2].
- Sociology - Interactions with sociology promise to be extremely fruitful. For instance, the issue of how ideas and innovations diffuse [3] involves the question of who is positively or negatively disposed towards whom, and hence who would be more or less receptive to new information transmission from a given source.
- eRulemaking - Sentiment analysis has specifically been proposed as a key enabling technology in eRulemaking, allowing the automatic analysis of the opinions that people submit about pending policy or government-regulation proposals [4].
- Politics - As is well known, opinions matter a great deal in politics. Some work has focused on understanding what voters are thinking [5], whereas other projects have as a long term goal the clarification of politician position such as what public figures support or oppose.

1.3 Problem Statement

The main objective of the present research work can be described by the statement of the problem expressed as follows:

“To formulate a sentiment analysis algorithm that can work for sentences those have more than one phrase and are joined by conjunctions”.

To achieve the above objective of sentiment analysis in presence of conjunctions following smaller objectives are set:

- To extract the linguistic information from the text.
- To extract the individual polarity of each phrase in the sentence.
- To form rules for each conjunction to join the individual polarity of different phrases.

As earlier stated, we are calculating the text polarity in presence of conjunctions. Therefore the intention is to form and implement a rule set for conjunctions to join the individual sentiment expressed in different part of text.

To achieve the above objective the following design goals are set:

- To devise a data structure to store the linguistic information present in text. By linguistic information we mean subject, verb and object present in the different phrases of sentence. One set of subject, verb and object is termed as triplet. So a sentence can have one or more than one triplets.
- To analyze the output of different parsers for different input of text to record some trend in the position of subject, verb and object. In this way we can form a rule set to extract the linguistic information hidden in text.
- To formulate a strategy to find the valance of a word that is not present in the knowledgebase. By valance we mean a numerical score assigned to each word. The valance is in the range of -5 to 5. The more the valance is negative more the word is used in negative context and vice versa. By knowledgebase we mean a dictionary which contains words with their numerical valences.
- To form rule set for conjunctions to join the individual polarity of triplets. Each conjunction can have more than one rule.
- To devise testing strategy for each module. The work is done by modular approach. So for a better efficiency and results modular testing should be done.

1.4 Thesis Organisation

Remaining thesis is organised as follows:

Chapter 2 details the fundamentals and provides a literature review of the various pre-existing sentiment analysis techniques such as word sentiment classification, machine learning approaches and valence assessment approach. Each technique is further explained with the help of various approaches. Research gaps and shortcomings are identified and described.

Chapter 3 provides a detailed description of proposed scheme for sentiment analysis in presence of conjunctions. We need some rules to join the numerical valences of two phrases joined by some conjunction. So in this chapter we discuss the rules formed for each conjunction in different scenarios.

Chapter 4 gives the brief introduction of the modular implementation of the proposed scheme. The chapter discusses the design and implementation of the algorithm. The work is divided among five modules: triplet extraction module, knowledgebase extension module, valence assessment module and opinion detection module.

Chapter 5 includes the results and discussion on them. It also provides analysis of important performance parameters and requirements.

Chapter 6 concludes the thesis with a summary of contribution towards sentiment analysis in presence of conjuncts. Possible horizon for future work is also discussed.

Chapter 2

Background and Literature Survey

Today, very large amount of reviews are available on the web, as well as the weblogs are fast-growing in blogosphere. Product reviews exist in a variety of forms on the web: sites dedicated to a specific type of product such as digital camera, sites for newspapers and magazines that may feature reviews like Rolling Stone or Consumer Reports, sites that couple reviews with commerce like Amazon, and sites that specialize in collecting professional or user reviews in a variety of areas like Rottentomates.com. Users also comment on products in their personal web sites and blogs, which are then aggregated by sites such as Blogstreet.com, AllConsuming.net, and onfocus.com.

The information mentioned above is a rich and useful source for marketing intelligence, social psychologists, and others interested in extracting and mining opinions, views, moods, and attitudes. For example, whether a product review is positive or negative; what are the moods among Bloggers at that time; how the public reflect towards this political affair, etc.

Analysis of favourable and unfavourable opinions is a task requiring high intelligence and deep understanding of the textual context, drawing on common sense and domain knowledge as well as linguistic knowledge. The interpretation of opinions can be debatable even for humans. For example, when we tried to determine if each specific document was on balance favourable or unfavourable toward a subject after reading an entire group of such documents, we often found it difficult to reach a consensus, even for very small groups of evaluators.

Sentiment detection dates back to the late 1990s (Argamon, Koppel, & Avneri, 1998[6]; Kessler, Nunberg, & SchÄutze, 1997[7]), but only in the early 2000s did it become a major sub-field of the information management discipline (Turney, 2002[8]; Pennbaker et al., 2004[9]; Pang et al.,2002[10]).

In this chapter we will discuss various sentiment analysis techniques. But first we will discuss some features of sentiment analysis. Features are the properties of a sentence which helps us in identifying the hidden sentiment of the text.

2.1 Features

In this section we focus on features that are specific to sentiment analysis:

2.1.1 Term presence vs frequency

It is traditional in information retrieval to represent a piece of text as a feature vector wherein the entries correspond to individual terms. One influential finding in the sentiment-analysis area is as follows. Term frequencies have traditionally been important in standard IR, as the popularity of tf-idf weighting shows; but in contrast, Pang et al. [10] obtained better performance using presence rather than frequency. That is, binary-valued feature vectors in which the entries merely indicate whether a term occurs (value 1) or not (value 0) formed a more effective basis for review polarity classification than did real-valued feature vectors in which entry values increase with the occurrence frequency of the corresponding term. This finding may be indicative of an interesting difference between typical topic-based text categorization and polarity classification: While a topic is more likely to be emphasized by frequent occurrences of certain keywords, overall sentiment may not usually be highlighted through repeated use of the same terms.

2.1.2 Parts of Speech

Part-of-speech (POS) information is commonly exploited in sentiment analysis and opinion mining. One simple reason holds for general textual analysis, not just opinion mining: part-of-speech tagging can be considered to be a crude form of word sense disambiguation [11].

Adjectives have been employed as features by a number of researchers [12]. One of the earliest proposals for the data-driven prediction of the semantic orientation of words was developed for adjectives. This finding has often been taken as evidence that (certain) adjectives are good indicators of sentiment, and sometimes has been used to guide feature selection for sentiment classification, in that a number of approaches focus on the presence or polarity of adjectives when trying to decide the polarity status of textual unit. Rather than focusing on isolated adjectives, Turney [8] proposed to detect document sentiment based on selected phrases, where the phrases are chosen via a number of pre-specified part-of-speech patterns, most including an adjective or an adverb.

The fact that adjectives are good predictors of a sentence being subjective does not, however, imply that other parts of speech do not contribute to expressions of opinion or sentiment. In fact, in a study by Pang et al. [10] on movie-review polarity classification, using only adjectives as features was found to perform much worse than using the same number of most frequent unigrams. The researchers point out that noun (e.g., “gem”) and verbs (e.g., “love”) can be strong indicators for sentiment. Riloff et al. [13] specifically studied extraction of subjective nouns (e.g., “concern”, “hope”) . There have been several targeted comparisons of the effectiveness of adjectives, verbs, and adverbs.

2.1.3 Negation

Handling negation can be an important concern in opinion and sentiment related analysis. While the bag-of-words representations of “I like this book” and “I don’t like this book” are considered to be very similar by most commonly used similarity measures, the only differing token, the negation term, forces the two sentences into opposite classes. There does not really exist a parallel situation in classic IR where a single negation term can play such an instrumental role in classification (except in cases like “this document is about cars” vs. “this document is not about cars”).

However, not all appearances of explicit negation terms reverse the polarity of the enclosing sentence. For instance, it is incorrect to attach “NOT” to “best” in “No wonder this is considered one of the best”. Na et al. [14] attempt to model negation more accurately. They look for specific part-of-speech tag patterns and tag the complete phrase as a negation phrase. For their dataset of electronics reviews, they observe about 3% improvement in accuracy resulting from their modelling of negations. Further improvement probably needs deeper syntactic analysis of the sentence.

Another difficulty with modelling negation is that negation can often be expressed in rather subtle ways. Sarcasm and irony can be quite difficult to detect, but even in the absence of such sophisticated rhetorical devices, we still see examples such as “It avoids all clichés and predictability found in Hollywood movies” — the word “avoid” here is an arguably unexpected “polarity reverser”. Wilson et al. [15] discuss other complex negation effects.

The two main popular approaches to sentiment detection, especially in the real-world applications, were based on machine learning techniques and based on word sentiment classification. Later valence assessment approach was introduced which is discussed later in the chapter.

2.2 Word Sentiment Classification

The earliest approach to find the sentiment was based on word sentiment classification. Sometimes this approach is also referred as Keyword Spotting Approach. The keywords from the text are extracted and based on keywords the sentiment of text is assessed.

Classifying the semantic orientation of individual words or phrases, such as whether it is positive or negative or has different intensities, generally using a pre-selected set of seed words. Some studies showed that restricting features to those adjectives for word sentiment classification would improve performance [8][19]. However, more researches showed most of the adjectives and adverb and a small group of nouns and verbs possess semantic orientation.

Automatic methods of sentiment annotation at the word level can be grouped into two major categories:

- corpus-based approaches
- dictionary-based approaches.

The first group includes methods that rely on syntactic or co-occurrence patterns of words in large texts to determine their sentiment [8][17][18]. The second group uses WordNet [20] information, especially, synsets and hierarchies, to acquire sentiment-marked words [21] or to measure the similarity between candidate words and sentiment-bearing words such as good and bad [22].

Turney and Littman(2004) [24] presents a strategy for inferring semantic orientation from semantic association between words and phrases. It follows a hypothesis that two words tend to be the same semantic orientation if they have strong semantic association. Therefore, it focused on the use of lexical relations defined in WordNet to calculate the distance between

adjectives. Generally speaking, we can defined a graph on the adjectives contained in the intersection between a term set (For example, TL term set [24]) and WordNet, adding a link between two adjectives whenever WordNet indicates the presence of a synonymy relation between them, and defining a distance measure using elementary notions from graph theory. This approach however used only adjectives present in the text. Also there were no measures no handle the negation and conditionality. Also like all word sentiment classification techniques they touch the surface properties of text. The polarity of an adjective was judged based on the relative distance to only two other words.

Esuli et. Al(2005) [25] proposes a method that exploits the glosses or textual definitions that one term has in an online “glossary” or dictionary. Its basic assumption is that if a word is semantically oriented in one direction, then the words in it gloss tends to be oriented in the same direction. For instance, the glosses of good and excellent will both contain appreciative expressions; while the glosses of bad and awful will both contain derogative expressions.

Generally, this method can determine the orientation of a term based on the classification of its glosses. This method classifies a word more correctly as the word’s sentiment is judged based on the relative distance of its gloss.

Turney et. Al(2002) [8] formulate a strategy to infer semantic orientation from semantic association. The underlying assumption is that a phrase has a positive semantic orientation when it has good associations (e.g., “romantic ambience”) and a negative semantic orientation when it has bad associations (e.g., “horrific events”).

The semantic orientation of a given word is calculated from the strength of its association with a set of positive words, minus the strength of its association with a set of negative words. More concretely, the strength of the semantic association between words can express by calculating their pointwise mutual information (PMI) value. So, it focuses on inferring the semantic orientation of a word from its statistical association with a set of positive and negative paradigm words. Given a term t , and seed term sets S_p for positive set and S_n for negative set, the t ’s orientation value $O(t)$ (where positive value means positive orientation, and higher absolute value means stronger orientation) is given by:

$$O(t) = \sum_{t_i \in S_p} \text{PMI}(t, t_i) - \sum_{t_i \in S_n} \text{PMI}(t, t_i) \quad (2.1)$$

In addition, **Gamon and Aue (2005)** [27] described an extension to the technique for the automatic identification and labelling of sentiment terms described in Turney and Littman (2003) [24]. Besides the basic assumption in [8], Gamon and Aue (2005) adds a second assumption, namely that sentiment terms of opposite orientation tend not to co-occur at the sentence level. This additional assumption allows them to identify sentiment-bearing terms more reliably to some extent.

Hu and Liu (2006)[21] multiply or count the prior valence of opinion-bearing words of a sentence. They also consider local negation to reverse valence but they do not perform a deep analysis (e.g., semantic dependency). This method gives its way to valence assessment approach. Researchers integrate the rules of English language with the valence study to formulate the valence assessment approach.

According to a linguistic survey (**Pennebaker et al. 2003**)[9], only 4% of the words used in written texts carry affective content. This finding shows that using affective lexicons is not sufficient in recognizing affective information from text. It also indicates the difficulty of employing methods like machine-learning, keyword spotting, or lexical affinity.

2.3 Machine Learning Approaches

The first considerable contribution to sentiment analysis approach using machine learning approaches was of **Pang and Lee**[10]. The authors of that paper compare Naïve Bayes, Maximum Entropy and Support Vector Machine approaches to classify sentiment of movie reviews. It is shown that a ML algorithm outperforms a simple term counting method. The authors compared several ML algorithms and found that SVMs generally gave better results. Unigrams, bigrams, part of speech information, and the position of the terms in the text were used as features; however, using only unigrams was found to give the best results, with an accuracy of up to 72%. A variety of features was used with SVMs in an attempt to divide the data set not only into positive and negative, but also to give rankings of 1, 2, 3, and 4, where 1 means “not satisfied” and 4 means “very satisfied.” The proposed system performed fairly well at distinguishing classes 1 from 4, with about 76% accuracy. Separating classes 1, 2 from 3, 4 proved more difficult, with an accuracy of only 69%. They explain the relatively poor performance of the methods as a result of sentiment analysis requiring a deeper

understanding of the document under analysis. Linguistic components of the language were not studied in this approach resulting in poor accuracy.

The research of **Mullen and Collier (2004)[12]** introduced an approach called hybrid SVM, which brings together diverse sources of potentially pertinent information, including several favourability measures for phrases and adjectives and a knowledge of the topic of the text. The authors used a hybrid model of machine learning and word sentiment classification. Models using the features introduced are further combined with unigram models which have been shown to be effective in the past (Pang et al. 2002) and lemmatized versions of the unigram models. Experiments on movie review data from the Internet movie database demonstrated that hybrid SVMs which combine unigram-style, feature-based SVMs with those based on real-valued favourability measures, obtained superior performance. We observe that sentences typically convey affect through underlying meaning rather than affect words, and thus evaluating the affective clues is not sufficient in recognizing affective information from texts. The sentiment of the text is conveyed by the meaning rather than the words. So absence of linguistic analysis hinders the accuracy of this approach.

2.4 Valence Assessment Approach

Until 2009 word sentiment classification and machine learning approaches were the most used approaches in the sentiment analysis applications. In 2009 Mostafa Al Masum Sheikh [33] described a well-founded approach for the task of sentence level sentiment analysis by studying the relationship between sentiments conveyed through texts and structure of natural language by a method of numerical analysis.

Different approaches have been employed to “sense” sentiment, especially from the texts, but none of those ever considered the valence based appraisal structure of sentiments that was employed here.

The author describes an approach to sense sentiments contained in a sentence by applying a numerical-valence based analysis. They developed a linguistic tool, SenseNet, that provides lexical-units on the basis of each semantic verb frame obtained from the input sentence; assigns a numerical value to those based on their sense affinity; assesses the values

using rules; and finally outputs sense-valence for each input sentence. This approach changes the dynamics of sentiment analysis field.

Here first the linguistic information from the text is retrieved. Linguistic information is stored in the form of a triplet (subject, object and verb). Triplet is extracted for each occurrence of verb in the text. Here by valence we actually mean a numerical value which is assigned to each word. More the word used in negative context lesser is the value of the valence and vice versa. We have a knowledgebase which contains words and their valences. Valence for each word in the triplet is found out from the knowledgebase. Then rules are applied to evaluate valance of the triplet. In this approach the triplet which occurs first in the sentence is given more weightage in the overall valance of the sentence. Then finally valence for the sentence is evaluated. Although this approach uses some of the linguistic properties of the language, it also misses on some properties. The conjunctions used in the sentence have a huge impact on the overall polarity of text. But this approach does not take into account the effects of conjunctions.

2.5 Research Gaps

Following are the shortcoming and research that have been observed:

- While combining the valences of the individual triplets, the previous approaches give more weightage to the triplet which occurs first in the sentence. The previous approaches do not take into account the conjunctions which join the two parts of the sentences. So there is a need of rule set to combine the valences of the triplets taking into account the conjunction used in the sentence.
- The unsupervised sentiment analysis algorithms use a knowledgebase (database of words and their valences). So this database must be exhaustive such that it contains all the words. We start with a list of few words. So there is a need of extension of database.
- The words in the database are stored in their base forms. So there is a need of pre-processing of text which can address the issues like stemming, spelling mistakes etc.
- Most of the previous approaches were domain specific. A domain specific classifier was built for each domain. However a generalised classifier is required.

Chapter 3

Proposed Technique for Sentiment Analysis

The increasing interest in opinion mining and sentiment analysis is partly due to its potential applications which we have just discussed. Equally important are the new intellectual challenges that the field presents to the research community. The interpretation of opinions is usually debatable affair even for humans. However our approach is an attempt towards this task.

3.1 Framework for Sentiment Analysis

In this chapter we discuss the algorithm for sentiment analysis at sentence level in presence of conjunctions. We suggest a way to join the individual polarities of phrases present in the text. This will help us to find the hidden sentiment present in the mixed reviews. Some reviewers use terms that have negative connotations, but then write an equivocating final sentence explaining that over- all they were satisfied. Mixed reviews introduce considerable noise to the problem of scoring words. We also suggest a modified approach for linguistic information extraction from the text. This approach output is worked upon to get the hidden sentiment present in the text. As shown in Fig 3.1 the proposed work can be divided among five different modules, each of which is explained below.

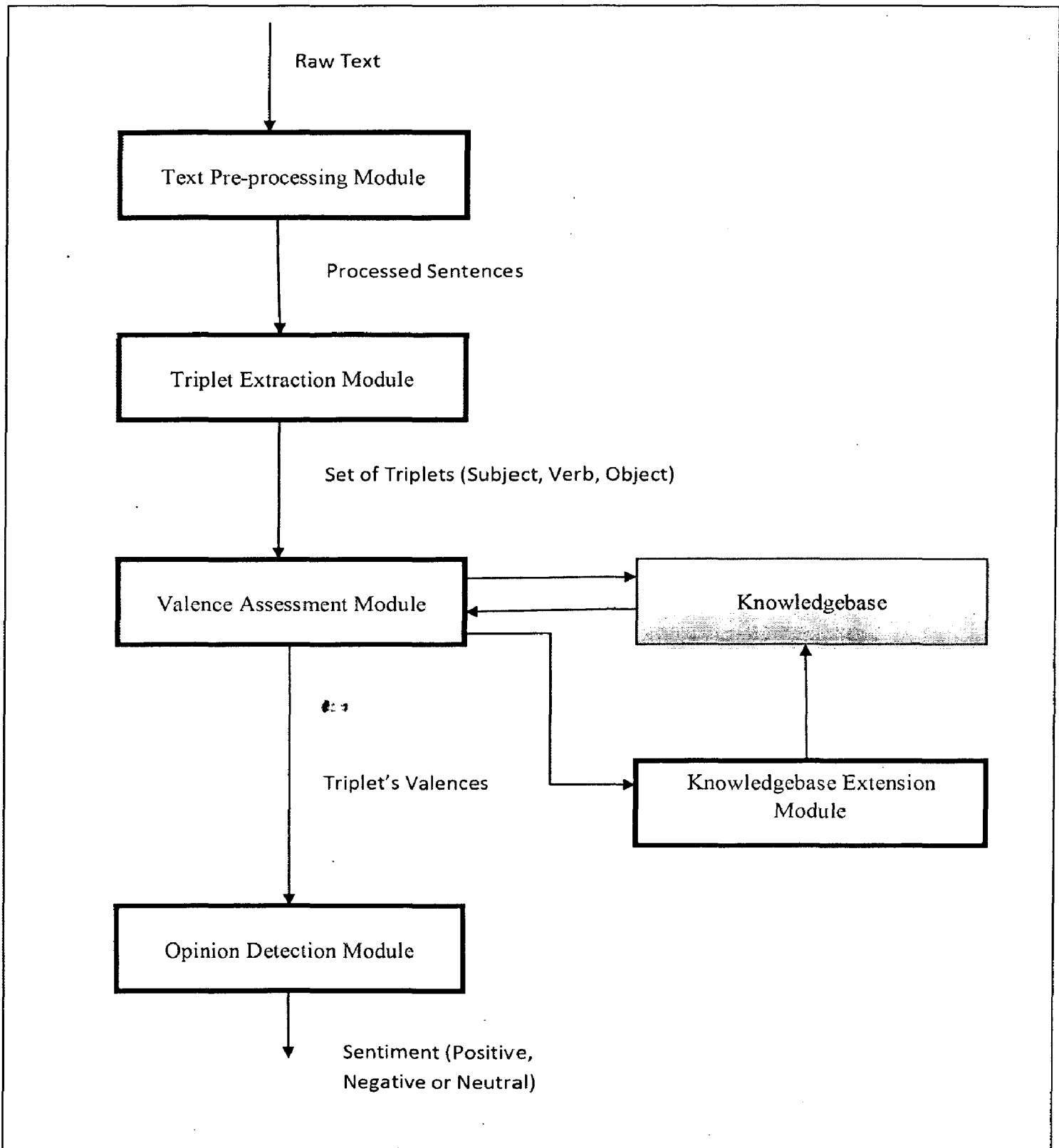


Figure 3.1 Framework for Sentiment Analysis

3.2 Text Pre-processing Module

This module deals with the first and foremost step of any text mining task that is to clean the data and prepare it for use in the algorithm. There are three steps in pre-processing strategy we employed.

1. Change to lower case

Words are store in the lower case in the knowledgebase. When we search a word in the knowledgebase, sometimes it happens, the result is NULL and the word is present is the lexicon. This happens due to the difference in the case of word present and the word searches. So the text is converted to all lower case first.

2. Spelling Mistake

We have a list of words which are misspelled frequently with their misspelled spellings. So each word in the text is searched in the list and if found is replaced by the correct spelling.

3. Lemmatization

Words are store in the base form(lemma) in the knowledgebase. So before computation text is lemmatized. In computational linguistics, lemmatisation is the algorithmic process of determining the lemma for a given word. In many languages, words appear in several inflected forms. For example, in English, the verb 'to walk' may appear as 'walk', 'walked', 'walks', 'walking'. The base form, 'walk', that one might look up in a dictionary, is called the lemma of the word.

Lemmatisation is closely related to stemming. The difference is that a stemmer operates on a single word without knowledge of the context, and therefore cannot discriminate between words which have different meanings depending on part of speech. However, stemmers are typically easier to implement and run faster, and the reduced accuracy may not matter for some applications.

For instance:

1. The word "better" has "good" as its lemma. This link is missed by stemming, as it requires a dictionary look-up.

2. The word "meeting" can be either the base form of a noun or a form of a verb ("to meet") depending on the context, e.g., "in our last meeting" or "We are meeting again tomorrow". Unlike stemming, lemmatisation does select the right lemma depending on the context.

The following is an example of lemmatisation and stemming. Given the following sentence: "The quick brown fox jumps over the lazy dogs." the lemmas from the words in the sentence would be as follows: "the quick brown fox jump over the lazy dog".

3.3 Triplet Extraction Module

After the pre-processing of the text, our next step is to convert the text into machine format. We have to store the linguistic information of the text into a data structure so that we can apply some functions to calculate the hidden sentiment. Our approach to these problems relies on breaking sentences into three pieces consisting of a subject, a relation, and an object. In this project we will call this as a triplet. We use natural language processing tools to for transformation of natural language text into a into or representation that facilitates an improvement on the processing of information.

Natural language processing in our system relies on a parse-tree produced by an existing NLP parse engine. We chose Stanford's JavaNLP [34] parsing engine because it represents an established code base as well as for its log-linear run time. JavaNLP parses all entered text into a tree structure that begins at a root node, denoted as root and containing no information, and progresses downwards to leaf nodes based on phrasal dependence.

A sentence (S) is represented by the parser as a tree having three children: a noun phrase (NP), a verbal phrase (VP) and the full stop (.). The root of the tree will be S.

Once the sentence tree has been created, our program parses it triplet formation. The three entities of each triplet are subject, verb and object. A triplet may also contain adjectives and adverbs related to the entities. A triplet is extracted for each occurrence of verb in the sentence. So multiple triplets can be extracted for a single snippet. Conversion from a natural language sentence to triplet format occurs in three phases:

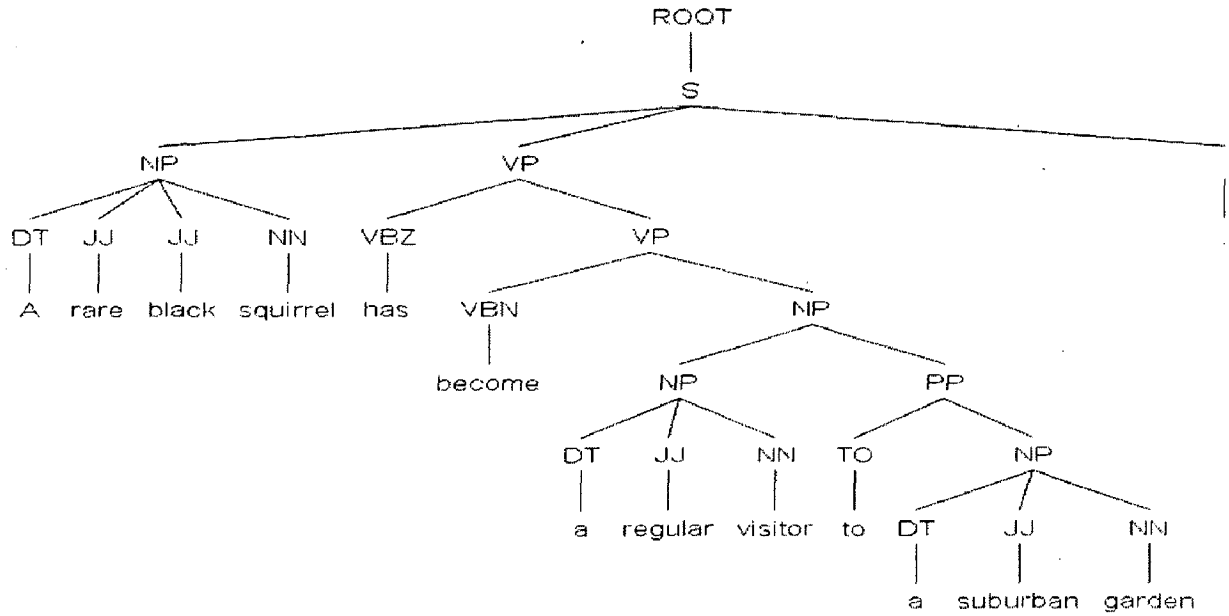


Figure 3.2: Parse tree generated for the sentence “A rare black squirrel has become a regular visitor to a suburban garden.”

1. Action Recognition

First, for determining the action in the sentence, a search will be performed in the VP subtree. The deepest verb descendent of the verb phrase will give the second element of the triplet. Verbs are found in the following subtrees:

SUBTREE	TYPE OF VERB
VB	Verb, Base Form
VBD	Verb, Past Tense
VBG	Verb, Present Participle
VBN	Verb, Past Participle
VBP	Verb, not 3rd Person Singular
VBZ	Verb, 3rd Person Singular

Table 3.1: List of subtrees in which verbs are found and associated type of verb

Procedure EXTRACT-VERB

```

1  EXTRACT-VERB(VP_subtree) returns a solution, or NULL
2  verbe ← deepest verb found in VP_subtree
3  verbAttributes ← EXTRACT-ATTRIBUTES(predicate)
4  result ← predicate U predicateAttributes
5  if result ≠ NULL then return result
6  else return NULL

```

2. Subject Recognition

Secondly we intend to find the subject of the sentence. In order to find it, we are going to search in the NP subtree. The subject will be found by performing breadth first search and selecting the first descendent of NP that is a noun. Nouns are found in the following subtrees:

SUBTREE	TYPE OF NOUN
NP	Noun, Common Singular
NNP	Noun, Proper Singular
NNPS	Noun, Proper Plural
NNS	Noun, Common Plural

Table 3.2: List of subtrees in which nouns are found and associated type of noun

Procedure EXTRACT-SUBJECT

```

1  EXTRACT-SUBJECT(NP_subtree) returns a solution, or NULL
2  subject ← first noun found in NP_subtree
3  subjectAttributes ← EXTRACT-ATTRIBUTES(subject)
4  result ← subject U subjectAttributes
5  if result ≠ NULL then return result
6  else return NULL

```

3. Object Recognition

Thirdly, we look for objects. These can be found in three different subtrees, all siblings of the VP subtree containing the predicate. The subtrees are: PP (prepositional phrase), NP and

Procedure EXTRACT-OBJECT

```
1  EXTRACT-OBJECT(VP_sbtree) returns a solution, or NULL
2  siblings ← find NP, PP and ADJP siblings of VP_subtree
3  for each value in siblings do
4      if value = NP or PP
5          object ← first noun in value
6      else
7          object ← first adjective in value
8  objectAttributes ← EXTRACT-ATTRIBUTES(object)
9  result ← object ∪ objectAttributes
10 if result ≠ NULL then return result
11 else return NULL
```

ADJP (adjective phrase). In NP and PP we search for the first noun, while in ADJP we find the first adjective. For each element, attributes are found. For example attributes of a noun are mainly adjectives and attributes of verb are mainly adverb. Negation in each phrase is also checked at this step.

Procedure EXTRACT-ATTRIBUTES

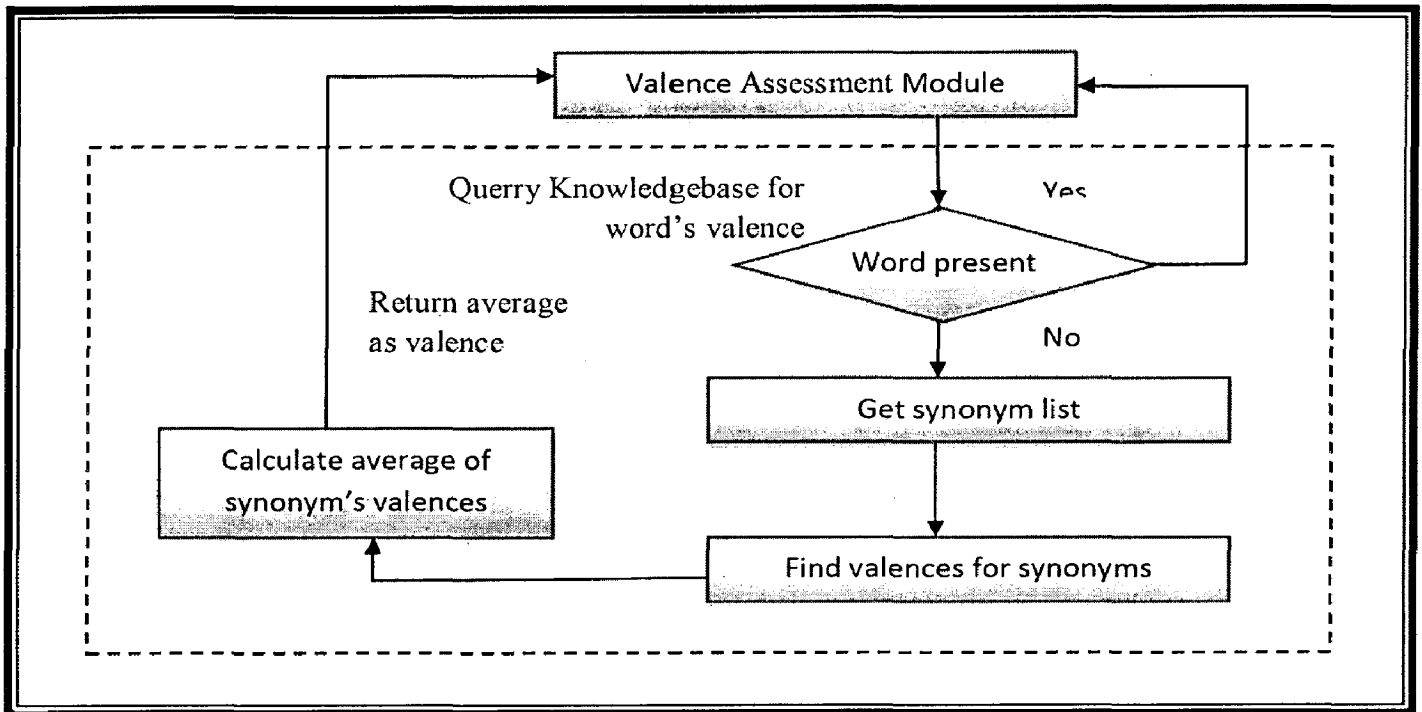
```
1  EXTRACT-ATTRIBUTES(word) returns a solution, or NULL
2  if adjective(word)
3      result ← all RB siblings
4  else if noun(word)
5      result ← all JJ, CD, ADJP siblings
6  else if verb(word)
7      result ← all ADVP siblings
8  if result ≠ NULL then return result
9  else return NULL
```

3.4 Knowledgebase Extension Module

A common approach to sentiment assessment is to start with a set of lexicons whose entries are assigned a prior valence indicating whether a word, independent of context, evokes something positive or something negative. We are calling our lexicon as knowledgebase. The initial database does not include an exhaustive list of all words in English language. So there are situations in which numerical valence for some words is not found out in the lexicon. This module is developed to handle these types of situations.

If a word is not found in the knowledgebase, WordNet and ConceptNet are utilized to find the synonyms to the word. The valence for those synonyms is then found out in the

knowledgebase, and the average of the numerical valence of the found words in the knowledgebase is assigned to the word. A new entry with word and its valence is also entered in the knowledgebase.




 -- Knowledgebase Extension Module

Figure 3.3: Functionality of Knowledgebase Extension Module

3.5 Valence Assessment Module

A basic idea of valence assessment technique for sentiment analysis was discussed in the literature review. Mostafa Al Masum(2009)[33] described a well-founded approach for the task of sentence level sentiment analysis by studying the relationship between sentiments conveyed through texts and structure of natural language by a method of numerical analysis. We have adopted the same approach have used their rule set to obtain triplet level numerical valence.

The words in knowledgebase are classified in certain categories. The verbs are classified into two groups, the affective verb (AV) group and the non-affective verb (V) group. The verbs having the tag <affect> in the knowledgebase are members of AV. Both AV and V are

further partitioned into positive (AVpos, Vpos) and negative (AVneg, Vneg) groups on the basis of their prior valence. Similarly, adjectives (ADJ), adverbs (ADV), concepts (CON) also have positive and negative groups indicated by ADJpos, ADJneg, ADVpos, ADVneg, CONpos, and CONneg, respectively. For a named entity (NE) the system creates three kinds of lists, namely ambiguous named entity (NEambi), positive named entity (NEpos) and negative named entity (NEneg).

All the triplets obtained from the input sentence are processed to assign a valence value to the sentence. This procedure involves the following steps:

- Rules are applied to assign contextual valence to the subject, verb and object of the triplet considering their attributes.
- Conditionality, negation, and previously assigned contextual valence values are considered to assign a contextual valence to the triplets.

At this point of time we have valences of each word in the triplet. Pronouns (e.g. I, he, she etc.) and proper names (not found in the listed named entity) are considered as positive valenced actors with a score 1 out of 5 for simplicity. Here are some example rules to compute contextual valence using attributes.

- ADJpos + (CONneg or NEneg) → neg. Valence (e.g., strong cyclone; nuclear weapon)
- ADJpos + (CONpos or NEpos) → pos. Valence (e.g., brand new car; final exam)
- ADJneg + (CONpos or NEpos) → neg. Valence (e.g., broken computer; terrorist group)
- ADJneg + (CONneg or NEneg) → neg. Valence (e.g., ugly witch; scary night)

For adverbs the following rules are applied. We have some adverbs tagged as <except> to indicate exceptional adverbs (e.g., hardly, rarely, seldom etc.) in the list. For these exceptional adverbs we have to deal with ambiguity as explained below.

- ADVpos + (AVpos or Vpos) → pos. Valence (e.g., write nicely; sleep well)
- ADVpos + (AVneg or Vneg) → neg. Valence (e.g., often miss; always fail)
- ADVneg + (AVpos or Vpos) → neg. Valence (e.g., rarely complete; hardly make)
- ADVneg + (AVneg or Vneg) → ambiguous (e.g., hardly miss; kill brutally)

Hence, the rules to resolve the ambiguity are:

- ADVneg-except + (AVneg or Vneg) → pos. Valence (e.g., rarely forget; hardly hate)
- ADVneg-not except + (AVneg or Vneg) → neg. Valence (e.g., suffer badly; be painful)

taking the contextual valence of action and object into consideration.

- Neg. Action Valence + Pos. Object Valence → Neg. Action-Object Pair Valence (e.g., kills innocent people, miss morning lecture, fail the final examination, etc.)
- Neg. Action Valence + Pos. Object Valence → Pos. Action-Object Pair Valence (e.g., quit smoking, hang a clock on the wall, hate the corruption, etc.)
- Pos. Action Valence + Pos. Object Valence → Pos. Action-Object Pair Valence (e.g., buys a brand new car, listen to the teacher, look after you family, etc.)
- Pos. Action Valence + Neg. Object Valence → Neg. Action-Object Pair Valence (e.g., buys a gun, patronize a famous terrorist gang, make nuclear weapons, etc.)

The above rules are naive and there are exceptions to the rules. In the sentences “I like romantic movies” and “She likes horror movies” the rules fail to detect both as conveying positive sentiment because “romantic movies” and “horror movies” are considered positive and negative, respectively. In order to deal with such cases we have a list of affective verbs (AVpos, AVneg) that uses the following rules to assign contextual valence for an affective verb.

- AVpos + (pos. or neg. Object Valence) = pos. Action-Object Pair Valence (e.g., I like romantic movies. She likes horror movies.)
- AVneg + (neg. or pos. Object Valence) = neg. Action-Object Pair Valence (e.g., I dislike digital camera. I dislike this broken camera.)

The rules for computing valence of a triplet are as follows.

- (CONpos or NEpos) + Pos. Action-Object Pair Valence = Pos. Triplet Valence (e.g., the professor explained the idea to his students.)
- (CONpos or NEpos) + Neg. Action-Object Pair Valence = Neg. Triplet Valence (e.g., John rarely attends the morning lectures.)
- (CONneg or NEneg) + Pos. Action-Object Pair Valence = Tagged Negative Triplet Valence (e.g., the robber appeared in the broad day light.) to process further.
- (CONneg or NEneg) + Neg. Action-Object Pair Valence = Neg. Triplet Valence (e.g., the strong cyclone toppled the whole city.)

3.6 Opinion Detection Module

3.6.1 Role of Conjunctions

A conjunction is a word that links words, phrases, or clauses, and it may be used to indicate the relationship between the ideas expressed in a clause and the ideas expressed in rest of the sentence. They play a vital role in deciding the overall polarity of a sentence. They often change the sentiment into the opposite orientation or add in the strength of the sentiment.

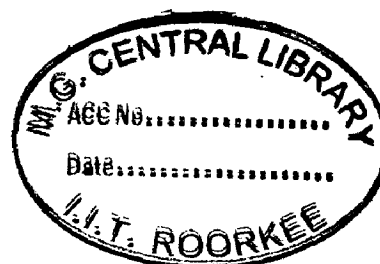
For example, Ram is a exceptionally brilliant student but somehow he did not get admission in IIT. If we only consider the word exceptionally, we will mistake the sentiment for positive. However, the word but in the sentence changes its sentiment orientation, actually it is negative. The difficulty with conjunctions is that they can occur almost anywhere in the structure of a sentence and therefore demands a thorough analysis of the sentence construct as we need to find the main clause in a sentence in order to decide the sentence level polarity.

3.6.2 Conjunction Analysis

We start by passing the current sentence to the JavaNLP parser [35], the output of the parser is the dependency tree with POS tagging and the typed dependencies of the words. For conjunction analysis typed dependency representation output is used.

For eg, for the sentence “Neither the orchestra nor the chorus was able to overcome the terrible acoustics in the church.” The typed dependency output is as follows:

```
preconj(orchestra-3, Neither-1)
det(orchestra-3, the-2)
nsubj(able-8, orchestra-3)
cc(orchestra-3, nor-4)
det(chorus-6, the-5)
conj(orchestra-3, chorus-6)
cop(able-8, was-7)
aux(overcome-10, to-9)
xcomp(able-8, overcome-10)
det(acoustics-13, the-11)
amod(acoustics-13, terrible-12)
dobj(overcome-10, acoustics-13)
prep(acoustics-13, in-14)
det(church-16, the-15)
pobj(in-14, church-16)
```



By the analysis of the output, we can find the conjunctions used in the sentence. The conjunctions usage is tagged with “aux” or “conj” tag in the output. Correlative conjunctions start is tagged with “preconj” tag and end is tagged with “cc” tag. For the above sentence with the help of tags we can show that two dependencies are present in the text, “neither.. nor” and “to”.

3.6.3 Conjunction Rule-set

In the previous sub-section it was described how valence is assigned to triplets. Now we explain how sentiment is assessed for a sentence. After conjunction analysis we have a set of triplet’s valences and the conjunctions which join them to form a sentence. The algorithm of this function is described below.

There are three types of conjunctions: coordinating conjunctions, correlative conjunctions, and subordinating conjunctions. The preposition “to” also acts as conjunction in many cases. We will study each of them one by one. Let there are two triplets T1 and T2.

Coordinating Conjunctions – for, and, but, or, so

All coordinating conjunctions except but act similarly. So rule for coordinating conjunctions except for 'but' is to add the valences of the two triplets. The triplets of opposite polarity can result in neutral sentiment.

$$\text{Valence Value} = ((\text{valence of T1}) + (\text{valence of T2}))/2$$

Example: I wanted to sit in the front of the balcony, so I ordered my tickets early.

When 'but' conjunctions is used in the text, the second triplet dominates over the first triplet. For example in the sentence "John was not a regular student but he finally scored good grades.", we can assess a positive sentiment although the first triplet indicates a strong negative sentiment. So while joining the two triplets joined by the conjunction 'but' the first triplet valence plays no role in the final sentiment.

$$\text{Valence Value} = \text{valence of T2}$$

Correlative Conjunctions – both..and, either...or, neither...nor, whether...or

The nature of all correlative conjunctions is almost similar. They join two qualities or two suggestions.

$$\text{Valence Value} = ((\text{valence of T1}) + (\text{valence of T2}))/2$$

But there is internal negation present in "neither.. nor" conjunction. So triplet valence must be negated. So in case of "neither..nor" the final sentiment is the opposite of sentiment calculated by the rule.

Subordinating Conjunctions – after, before, because, since, while, etc..

The subordinating conjunctions plays no differently from the correlative conjunctions. The rule remains the same for them.

To

The functionality of 'to' changes with the polarity-pair of the triplets. So there are four sub-cases. The valence value is the average of the absolute values of the triplet's valences.

$$\text{Valence Value} = (\text{abs}(\text{valence of T1}) + \text{abs}(\text{valence of T2})) / 2$$

The polarity of valence value is decided as follows:

- Pos. valence of T1 + Pos. valence of T2 → Pos. Contextual Valence
Example - I am interested to go for a movie.
- Neg. valence of T1 + Pos. valence of T2 → Neg. Contextual Valence
Example - It was really hard to swim across this lake.
- Pos. valence of T1 + Neg. valence of T2 → Neg. Contextual Valence
Example - It is easy to catch a cold at this weather.
- Neg. valence of T1 + Neg. valence of T2 → Pos. Contextual Valence
Example - It is difficult to take bad photo with this camera.

The above idea is further explained by an example of sentiment is assessed for the sentence "John was not a talented student but he finally scored good grades."

Step1 – Text Pre-processing

The text prep-processing module first changes the whole text to lower case and lemmatizes the text. The output of the module is as follows:

"john is not a talented student but he finally score good grade"

Step 2 – Triplet Extraction

There are two verbs present in the sentence. So this module generates two triplets for the sentence.

T1 : { john, is , student [talented] }

T2 : {he, score , grade[good]}

This module also indicates that negation is present in the first triplet.

Step 3 – Valence Assessment

From the knowledgebase we get the following prior valence for the words found in the example sentence.

is : 1.0, student : 2.37, talented : 5.0, score : 3.85, good : 5.0, grade : 4.32

The nouns and pronouns are assigned a valence of 1 for simplicity. The valence for the word grade was not found out in the knowledgebase, so the synonyms for the word “grade” were found out with the help of WordNet. The average of synonyms’ valences was assigned to the valence field of “grade”.

According to algorithm the valence for the two triplets were calculated as 5.685 and 9.01. Due to the presence of negation the polarity of the first triplet’s valence is reversed.

Step 4 – Opinion Detection

The first job of this module is to find the conjunction used in the text. The conjunction “but” is spotted in the typed dependency output of the Stanford parser.

The rule for the conjunction “but” is applied on the valence of the two triplets and a positive sentiment is sensed in the text.

Chapter 4

Implementation Details

4.1 NLP Tool Used – Stanford Parser

4.1.1 Description

Stanford Parser is a natural language parser developed by Dan Klein and Christopher D. Manning from The Stanford Natural Language Processing Group [34][35]. The package contains a Java implementation of probabilistic natural language parsers; a graphical user interface is also available, for parse tree visualization. The module we developed uses version 1.6.3, released on 09.07.2010.

The parser can read various forms of plain text input and can output various analysis formats, including part-of-speech tagged text, phrase structure trees, and a grammatical relations (typed dependency) format. For example, consider the text: “A rare black squirrel has become a regular visitor to a suburban garden.”

The following output shows part-of-speech tagged text, then a context-free phrase structure grammar representation, and finally a typed dependency representation. All of these are different views of the output of the parser.

Tagging

```
A/DT
rare/JJ
black/JJ
squirrel/NN
has/VBZ
become/VBN
a/DT
regular/JJ
visitor/NN
to/TO
a/DT
suburban/JJ
garden/NN
./.
```

Parse

```
(ROOT
(S
(NP (DT A) (JJ rare) (JJ black) (NN squirrel))
(VP (VBZ has)
(VP (VBN become)
(NP (DT a) (JJ regular) (NN visitor))
(PP (TO to)
(NP (DT a) (JJ suburban) (NN garden))))))
(. .)))
```

Typed dependencies

```
det(squirrel-4, A-1)
amod(squirrel-4, rare-2)
amod(squirrel-4, black-3)
nsubj(visitor-9, squirrel-4)
aux(visitor-9, has-5)
cop(visitor-9, become-6)
det(visitor-9, a-7)
amod(visitor-9, regular-8)
prep(visitor-9, to-10)
det(garden-13, a-11)
amod(garden-13, suburban-12)
pobj(to-10, garden-13)
```

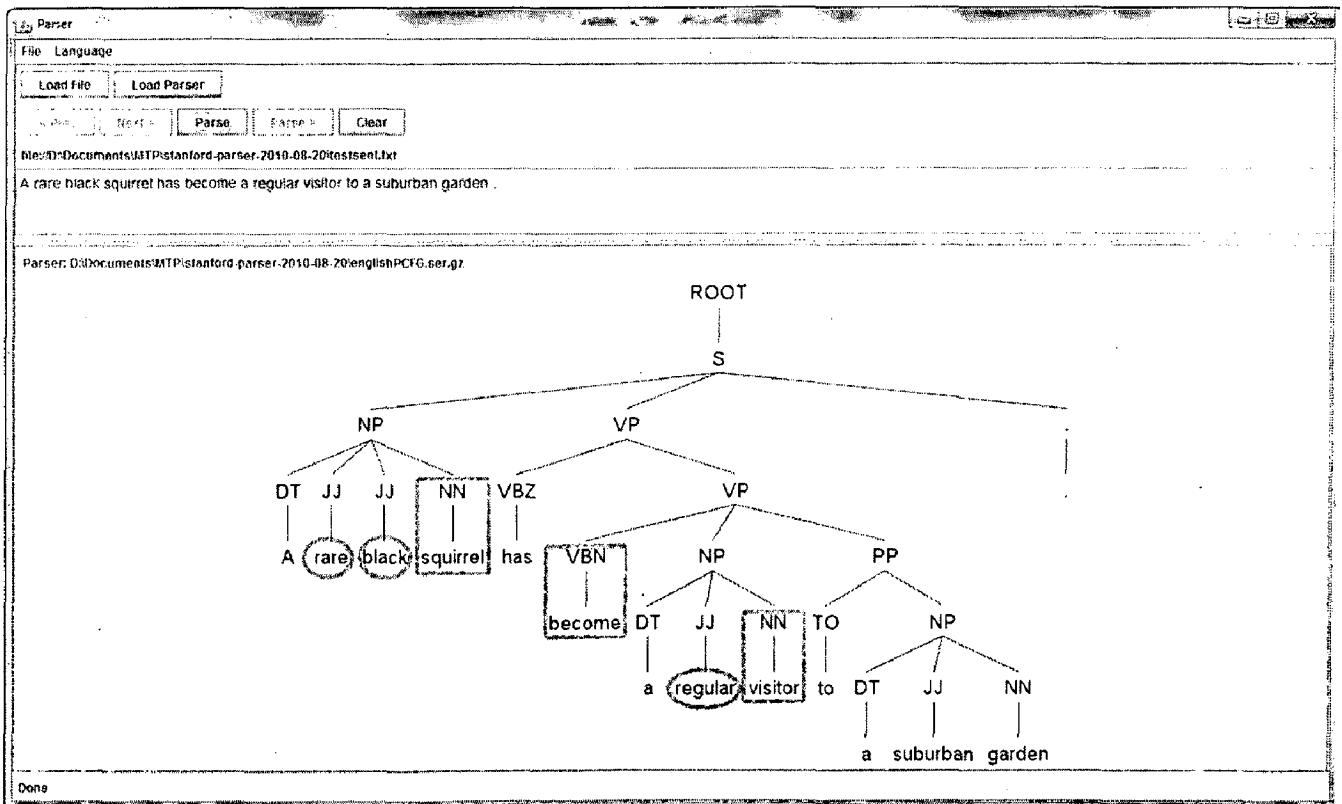


Figure 4.1: The parse tree generated by Stanford Parser

4.2 Linguistic Resource Used - WordNet

WordNet is a large lexical database of English, developed under the direction of George A. Miller. Nouns, verbs, adjectives and adverbs are grouped into sets of cognitive synonyms (synsets), each expressing a distinct concept. Synsets are interlinked by means of conceptual-semantic and lexical relations. The resulting network of meaningfully related words and concepts can be navigated with the browser. WordNet's structure makes it a useful tool for computational linguistics and natural language processing.

The purpose is twofold: to produce a combination of dictionary and thesaurus that is more intuitively usable, and to support automatic text analysis and artificial intelligence applications.

How WordNet is used?

A Java/Processing library is used that provides simple, string-based access to the Wordnet ontology. We have used to WordNet to find the synonyms of word.

4.3 Design and Development

The system is developed in a modular fashion. System consists of five modules and each stage's design is discussed individually.

The implementation of the proposed algorithms and all its pre-requisites are coded with using Java programming language. The expanse of the language and its seamless integration with fields such as databases, networks, data structure etc are one of the main reasons for choosing java as the language. Its modularity and object oriented nature gives the freedom to individually build and test the modules.

4.3.1 Text Preprocessing Module

We have implemented all the functions required to pre-process the text in the class Preprocess. Some pre-defined functions are also used to pre-process the text. Some main functions of the Preprocess are:

1. public spell_chk()

This function checks for each word in the text if it is present in the autocorrect list. If so, it replaces it with the correct spelling from the list.

2. lemmatize()

This function is used for the purpose of lemmatization. A single word is passed to the function and it returns lemma for the word.

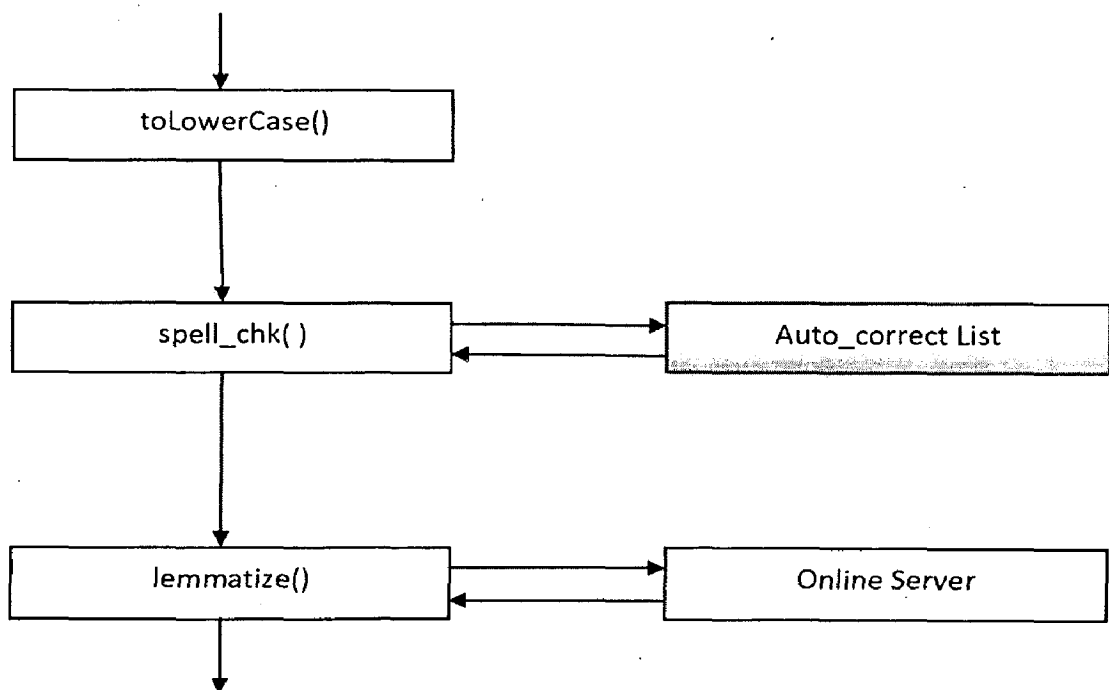


Figure 4.2: Function Calls in Text Pre-processing Module

4.3.2 Triplet Extraction Module

The implementation of all functions required to extract linguistic information from the text are defined in the class TripletFinder. There are two other classes which defines the triplet data structure and its various operations. The sequence of function calls is shown with the help of a diagram below.

The output of the Stanford parser is worked upon by these functions to construct a triplet. The output of the parser is in a tree structure. So these functions use tree traversal algorithms to spot all the entities of a triplet in the tree. Negation is also checked in this module by the Find_triplet() function, and if present a bool variable is set true and is returned to the FindATriplets() function along with the triplet.

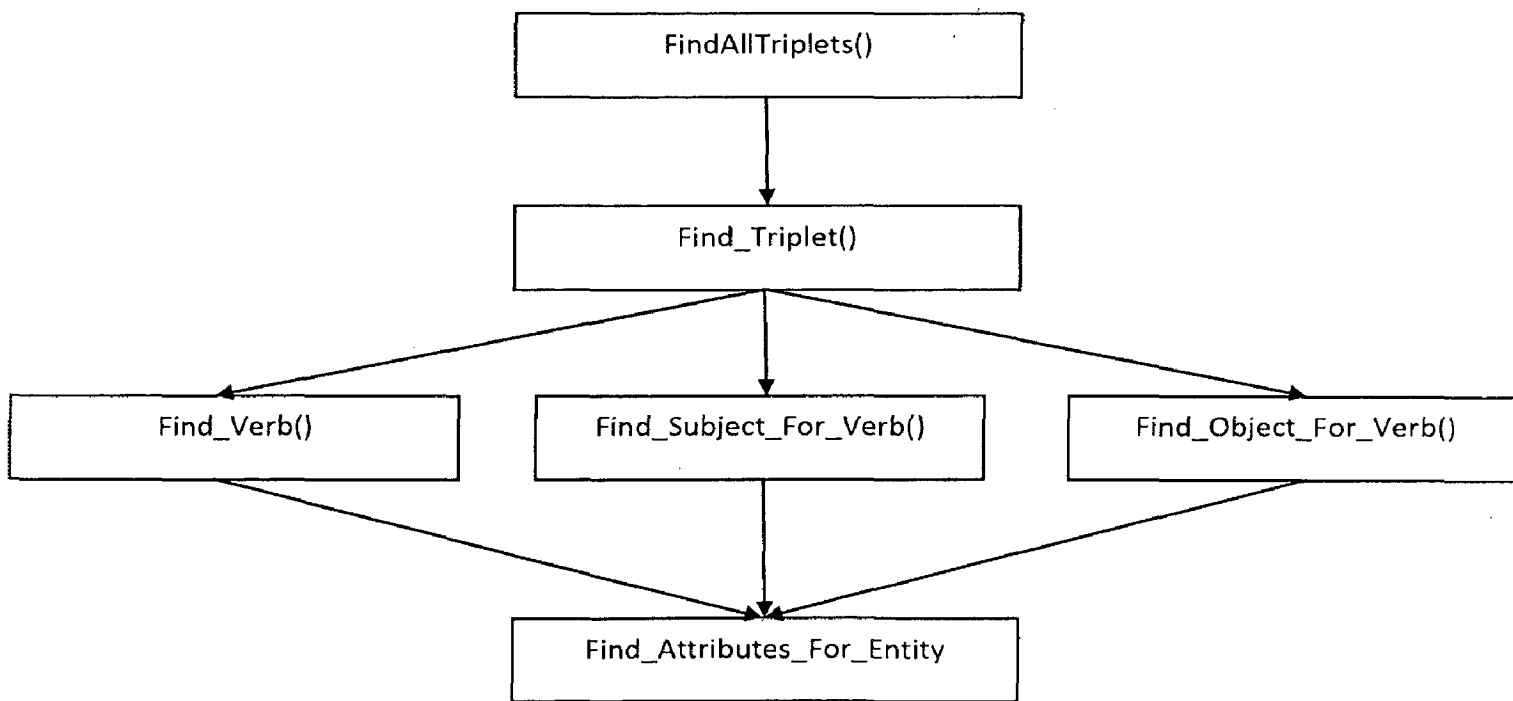


Figure 4.3: Sequence of Functions Calls in triplet extraction Module

4.3.3 Knowledgebase Extension Module

Knowledgebase extension is an important feature to increase the efficiency of the system. We have used WordNet for this purpose. All functions required for this purpose are defined in the class Knowledgebase. To make the search faster, we have loaded all our lexicon on to a map. A Java/Processing library is used that provides simple, string-based access to the Wordnet ontology.

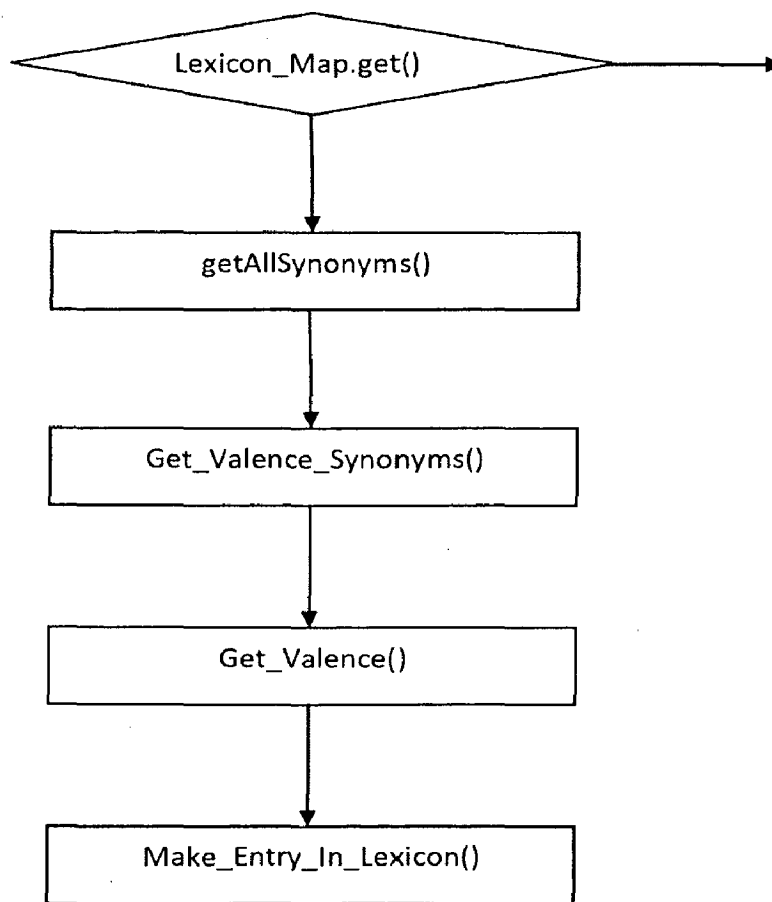


Figure 4.4: Sequence of functions calls in knowledgebase extension module

4.3.4 Valence Assessment Module

The valence assessment module is responsible for the triplet level numerical valence calculation. At this step, triplet is extracted. So the functions of this module are to find the valence of each word in the triplet and follow the rules to calculate triplet's valence. All functions required for this purpose are defined in the class ValanceProvider. According to algorithm first we have to calculate the object-verb pair valence, then the triplet valence.

The downward arrows in figure indicated the function call. The upward arrow are for return. The valence is returned with each function call return.

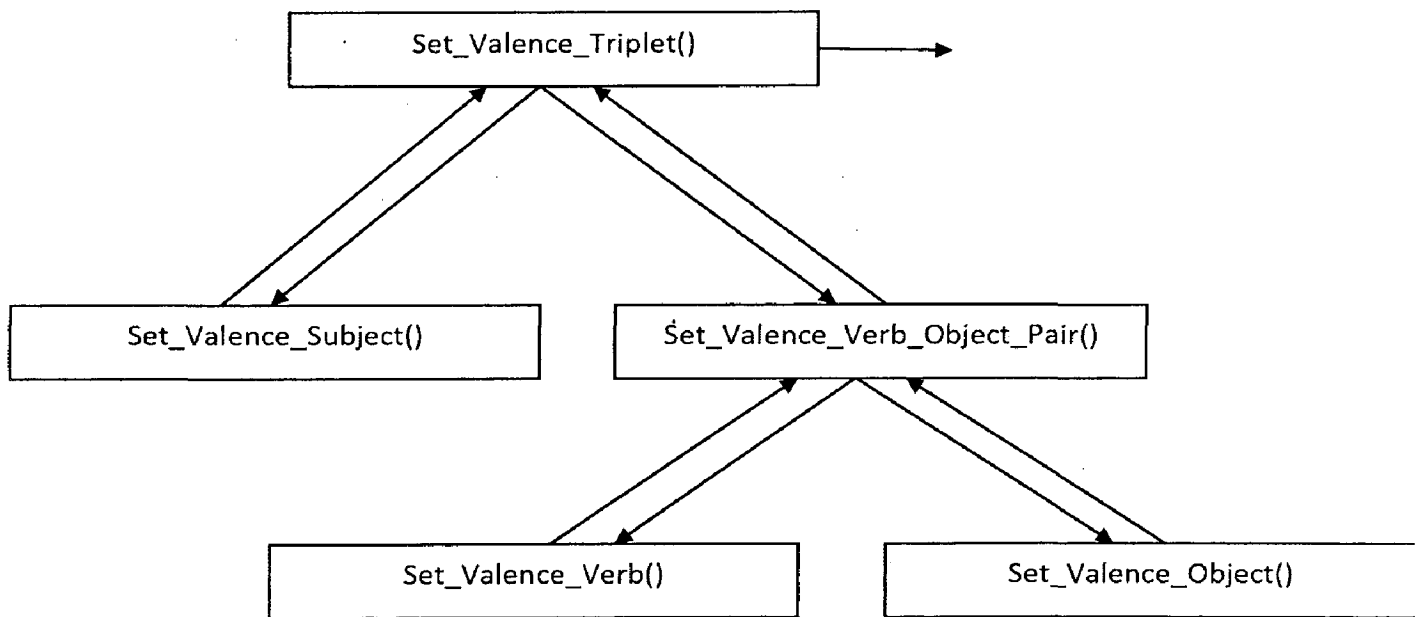


Figure 4.5: Sequence of Calls and Returns in Valence Assessment Module

4.3.5 Opinion Detection Module

The main class used to assess the sentiment of the text is IdentifyOpinion. All the functions used in this module are defined in this class. The functions of this class call the functions defined in the previous classes to complete the task. Get_Triplet_Valence() function returns all the triplets with their associated numerical valences. Next job is to find the conjunctions used in the text. Get_Conjunctions function works upon the typed dependency output of the Stanford parser with the help of string manipulation functions to get the list of conjunctions.

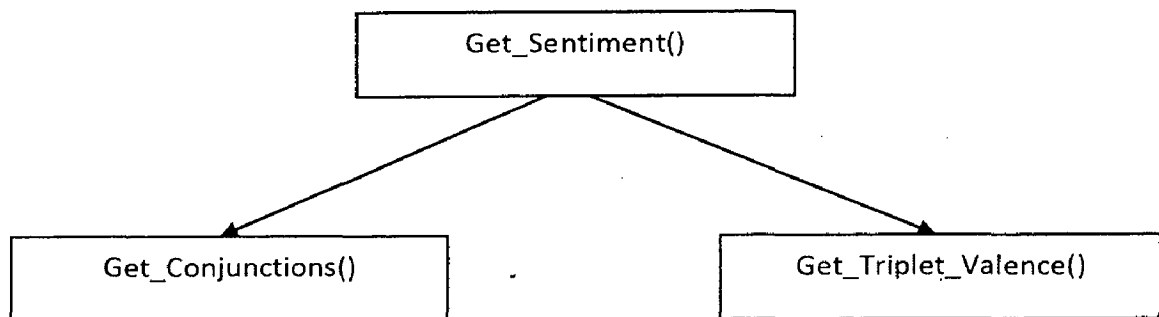


Figure 4.6: Function Calls in Opinion Detection Module

Chapter 5

Results and Discussion

5.1 Dataset used

5.1.1 Dataset for triplet extraction module

The triplet extraction module was tested with a dataset of randomly chosen five news articles from newspaper. The dataset of news articles was chosen for several reasons:

- One, the heavily fact based nature of the articles they represent an excellent choice for our method of information extraction.
- Second, the articles are written for human consumption making the job of the human tester easier.
- Also, this type of articles reflects a sizable portion of the documents placed on the Internet within a given day.

5.1.2 Dataset for opinion detection module

Cornell movie review dataset

URL: <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

These corpora, first introduced in Pang and Lee [32], consist of the following datasets, which include automatically derived labels.

- Sentiment polarity datasets:
 - Document-level: polarity dataset v2.0: 1000 positive and 1000 negative processed reviews.
 - Sentence-level: sentence polarity dataset v1.0: 5331 positive and 5331 negative processed sentences/snippets.

The primary motivation of using this dataset is that they contain individual sentences classified as positive or negative which is accord with the purpose of our first experiment, namely, to answer how efficiently the system can assess sentiments at sentence level.

Since movie reviews are known to be difficult to classify [Turney 2002], we are motivated to test the performance of our system with such data. A summary of used datasets is given in table 5.1.

Dataset	Data Type	Data Attributes	Data Source
Dataset A	Paragraph	Data collected from various domains, data used in triplet extraction module	Managed to collect the data from newspapers and triplets are extracted by human user and computer system
Dataset B	Sentence	Collected from Movie Review (Rotten Tomatoes). There are two files. One contains 5331 positive snippets and other has 5331 negative snippets, all snippets are down-cased	Sentence polarity dataset v1.0., can be found in at this source - http://www.cs.cornell.edu/people/pabo/movie-review-data/

Table 5.1: Input Datasets

5.2 Performance of triplet extraction module

Testing results for triplet extraction module were obtained by comparing the output of our system with the human generated triplets. We chose human testing rather than testing with some pre-existing corpus because of the novelty of our approach. While several testing corpuses address a problem that is related to those we wish to address none provide a dataset that provides for a clear translation from sentence to triplet.

The first part of the testing phase was accomplished by presenting three Computer Science graduate students who are not related to this research project with a worksheet that included instructions for triplet generation in addition to five news articles. They then generated all the triplets that they believed were possible from these articles. The triplets the students generated were then gathered and reviewed to form a set of correct student generated triplets. This set was created by first inspecting all triplets created by each subject and eliminating all incorrect triplets. The set was then further reduced by comparing each subject's triplets against the triplets generated by all if any incorrect triplets were present in this human created set. By doing this, the final set of human created triplets includes only correct unique triplets as given by the nine human subjects. Thus, through initial production followed by review a sort of "gold standard" of triplets was generated for the news articles.

This set is then compared against the triplets generated by our system. Comparison is done according to a similarity measure $\epsilon [0, 1]$ between two triplets.

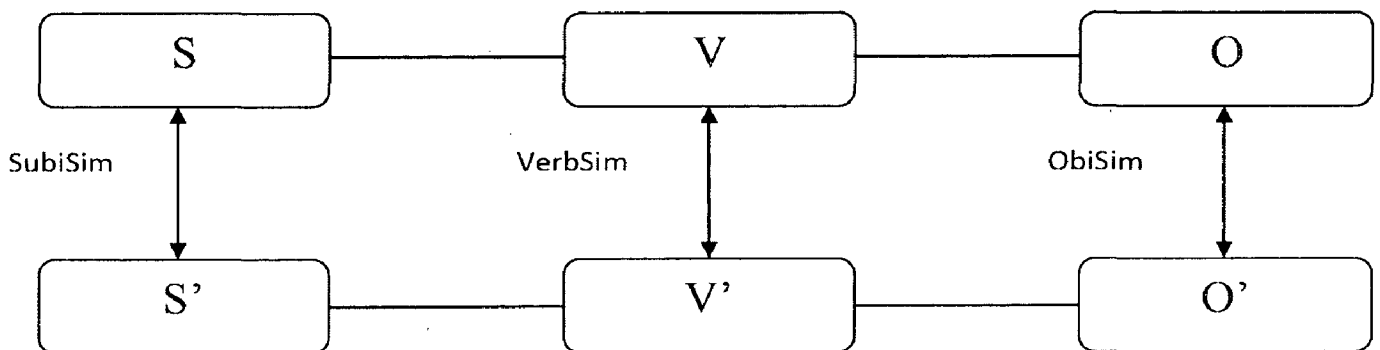


Figure 5.1 : Triplet Similarity Measure

$$\text{TripSim} = (\text{SubjSim} + \text{ObjSim} + \text{VerbSim}) / 3$$

$$\text{TripSim}, \text{SubjSim}, \text{ObjSim}, \text{VerbSim} \in [0, 1]$$

SubjSim is assigned a score of 1 if subject of system generated triplet and subject of human generated triplet are same; otherwise a score of 0 is assigned. Similarly scores of ObjSim and VerbSim are calculated.

Efficiency Calculation of the Module

Number of triplets in human generated set - 118

Number of triplets in system generated set – 131

Number of triplets with complete overlap – ~~79~~

Number of triplets with two matches – ~~34~~

Number of triplets with one match – 5

Number of triplets with no match – 0

Number of incorrect triplets – 13

Degree of Overlap	Number of Triplets	Score
3	79	79
2	34	22.66
1	5	1.66
0	0	0
Incorrect	13	0

Table 5.2: Triplet Extraction Module Testing Statistics

Total Score = 103.33

System Efficiency = 87.56%

The above statistics are more clearly explained with the help of bar graphs below.

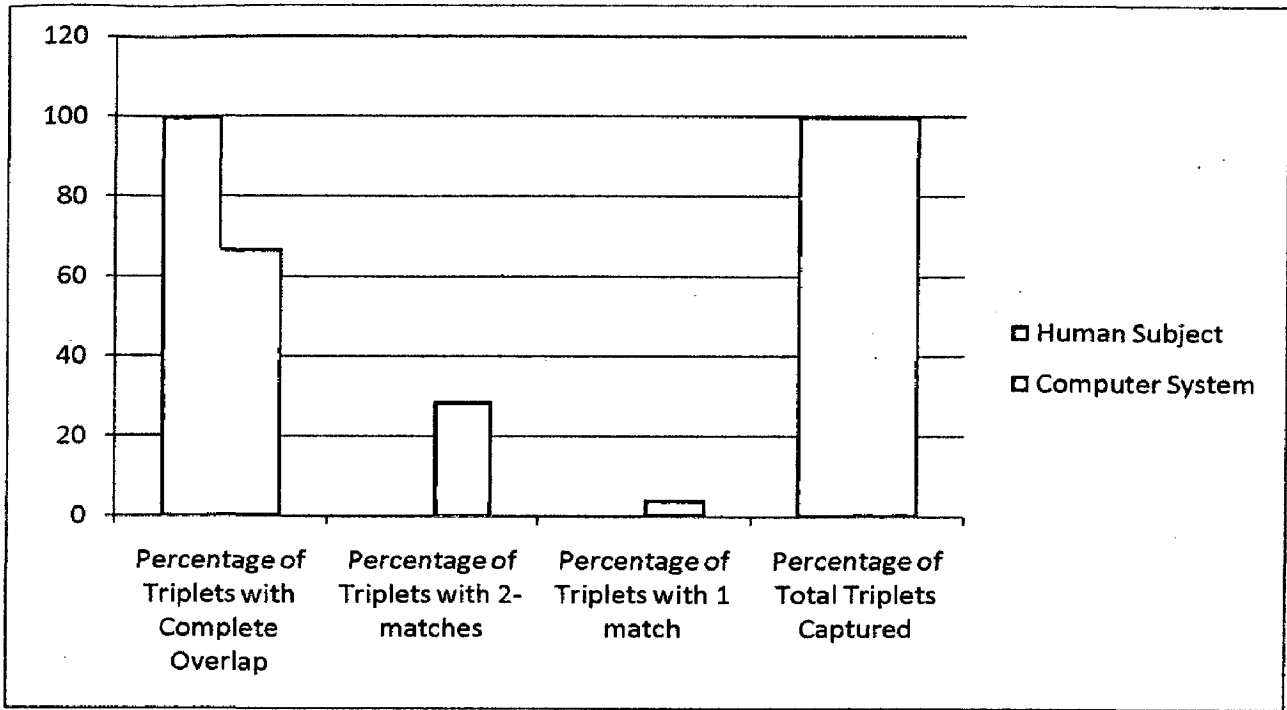


Figure 5.2: Percentage of Correct Triplets Generated : Expert Human vs Computer

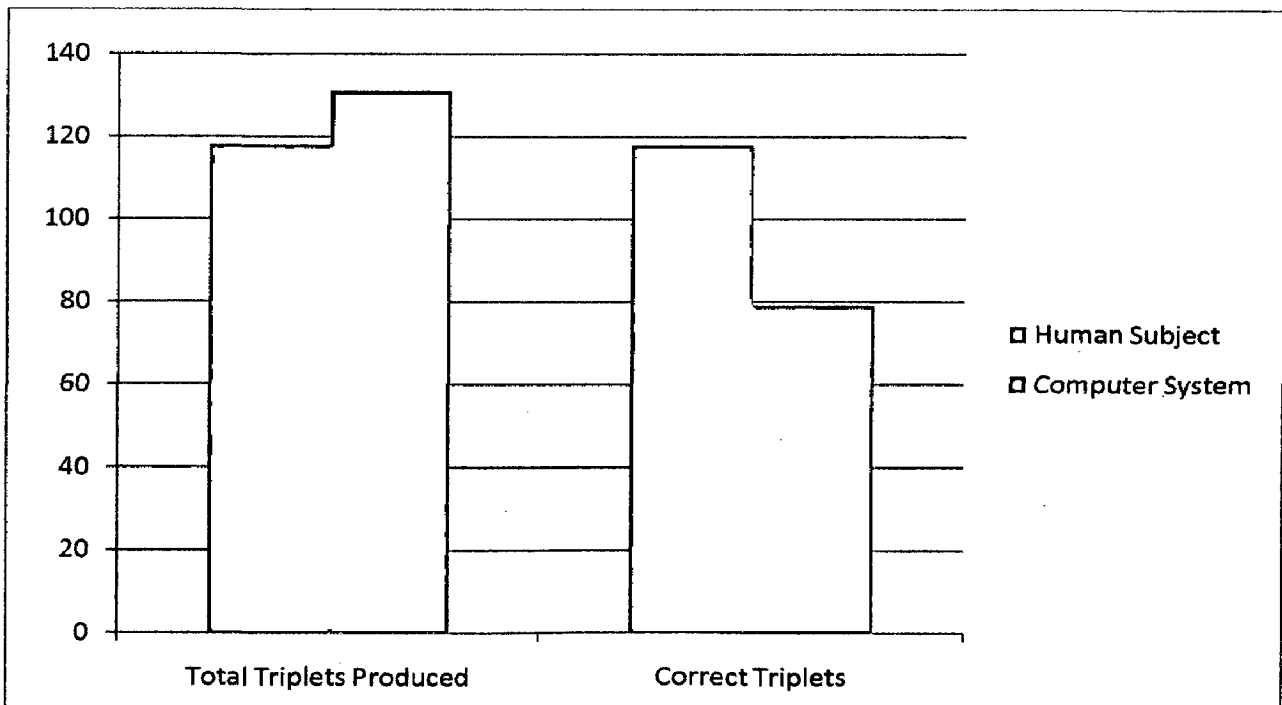


Figure 5.3: Comparison of Triplet Production: Expert Human vs Computer

5.3 Performance of Opinion Detection Module

The system tagged a sentence with positive, negative or neutral sentiment. Hence we set valence ranges to signal the neutrality of the sentence. The system finally gives a numerical valence in between -15 to 15 to a sentence. More the valence is negative, the sentence carry a strong negative opinion and vice versa.

Efficiency Calculation of the System

We are testing the system on sentence polarity dataset v1.0. The dataset contains 5331 positive and 5331 negative processed snippets. This dataset has become the de facto standard dataset for sentiment-classification and has been used in over 15 research papers. Table 5.3 summarizes the accuracy of different approaches for this dataset.

Clearly the valence assessment approach suggested by Mostafa Al Masum(2009) performs better than other word sentiment classification and machine learning approaches. We have adopted the same approach for sentiment analysis.

Author	Technique Used	Accuracy (%)
Pang & Lee(2002)	NB Classifier	71.5
Pang & Lee(2002)	ME	71.0
Pang & Lee(2002)	SVM	72.9
Salvetti(2004)	NB Classifier	79.5
Kamps et al.(2004)	WordNet	78.7
Mullen and Collier(2004)	SVM	81.0
Beineke(2005)	NB Classifier	65.2
Moastafa Al Masum(2009)	Valence Assessment	85.2

Table 5.3: Accuracy of different approaches of Sentiment Analysis

1000 snippets from both files of sentiment polarity dataset were chosen and the system was evaluated on them.

Sentiment Extracted	Positive Snippets	Negative Snippets
Positive	873	58
Neutral	86	62
Negative	41	880

Table 5.4: Opinion Detection Module Testing Statistics

System Accuracy = 86.65%

Our system gives the result of sentiment analysis with an accuracy of 86.65 %. The first module of our system, triplet extraction module extracted the linguistic information from text with an accuracy of 87.56%. So if we can enhance the results of triplet extraction module to 100%, the system can reach up to an accuracy of 98.96%.

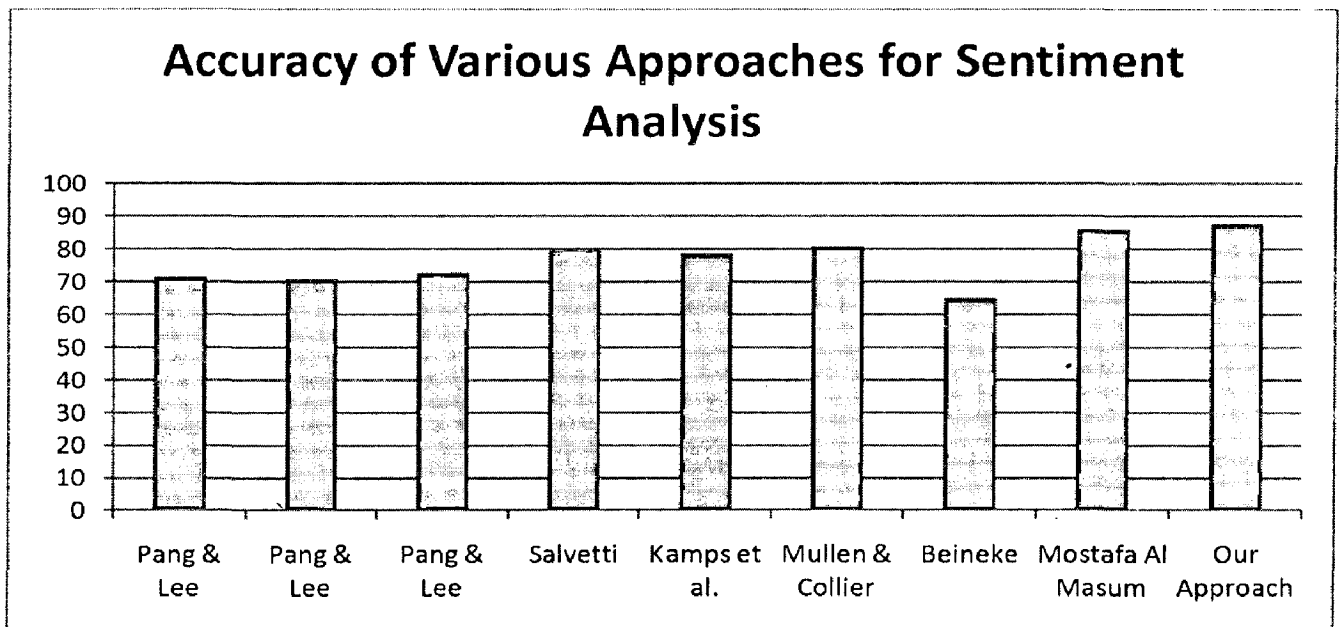


Figure 5.4: Accuracy of Various Approaches for Sentiment Analysis

5.4 Impact of various conditions and parameters on results

The accuracy of the system is hindered by the following factors:

- Knowledgebase Size – The knowledgebase we employed has approximately 10,000 words and their valences. Words which are not found in the knowledgebase are often assigned incorrect valences, resulting in wrong results.
- Foreign Words – Foreign words are treated as named entities by the parser and are tagged as noun. Named entities are always assigned a valence of 1.
- Syntax – Reviewers often do not follow the correct syntax of language while writing their opinions. Incorrect syntax lead to incorrect triplets.
- Number of Triplets - Triplets are formed for each verb encountered in the text. So number of triplets formed by the system is often more than the human generated set.

Chapter 6

Conclusion and Future work

“Now this is not the end. It is not even the beginning of the end. But it is, perhaps, the end of the beginning” - Sir Winston Churchill

6.1 Conclusion

Sentiment detection has a wide variety of applications in information systems, including classifying reviews, distinguishing synonyms and antonyms, extending the capabilities of search engines, summarizing reviews, tracking opinions in online discussions, and analyzing survey responses. There are likely to be many other applications that we have not anticipated. The technique described here proposes a novel method to recognize sentiment at sentence level in presence of conjunctions. The main features of the work are:

- We concentrate on the effects of the conjunctions and sentence construction which have not been researched. The system first performs semantic processing, then applies rules to assign contextual valence to the linguistic components in and then applies conjunction rules in order to obtain sentence- level sentiment valence.
- The system is well-founded because we have used the knowledgebase which employed both cognitive and commonsense knowledge to assign prior valence to the words and the rules are developed following the heuristics to exploit linguistic features.
- We have conducted several studies using various types of data that demonstrate the accuracy of our system. Moreover, it outperforms a state-of-the-art system.
- Our method does not need a training set since it depends on linguistic analysis.

Automatic sentiment detection can never be perfect. The opinion of the writer depends on the context in which he is writing the text. Also there are elements of sarcasm and irony which are still far to assess. We can observe that use of conjunctions have substantially increased the efficiency of sentiment analysis. On movie review data our approach when

incorporated with valence assessment approach outperformed other non-linguistic approaches. We also achieved better performance or almost similar performance with machine learning approaches with the same datasets.

In general terms the research aims at giving computer programs a skill known as artificial intelligence with the ability to understand human sentiment and to respond to it appropriately. We believe that this linguistic approach to assessing sentiment from texts would strengthen human-computer interaction with fun.

6.2 Scope for Future Work

There is a lot of scope to extend the current study in order to be more efficient in opinion mining. In future more work is needed to deal with these outstanding problems:

- Since sentiments can be expressed with various expressions including indirect expressions that require common sense reasoning to be recognized as a sentiment, it's been a challenge to analyze the complex structures of sentences in the input context that negates the local sentiment for the whole.
- Some reviewers use terms that have negative connotations, but then write an equivocating final phrase explaining that over all they were satisfied. Mixed reviews introduce considerable noise to the problem of scoring words.
- An accurate identification of semantic orientation requires analysis of units larger than individual words; it requires understanding of the context in which those words appear. To this end, we intend to use Rhetorical Structure Theory to impose on the text a structure that indicates the relationships among its rhetorical units.
- There are elements of sarcasm and irony which are still far to assess.
- Pronoun resolution can be worked upon to form more correct triplets.

References

- [1] M. Koppel and I. Shtrimberg, “Good news or bad news? Let the market decide”, *In Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text*, Palo Alto, CA, pages 86–88, 2004.
- [2] Ahmed Abbasi, “Affect intensity analysis of dark web forums”, *In Proceedings of Intelligence and Security Informatics (ISI)*, pages 282–288, 2007.
- [3] Everett Rogers, “Diffusion of Innovations”, *Free Press, New York, ISBN 0743222091, Fifth edition*, 2003.
- [4] Claire Cardie, Cynthia Farina, Thomas Bruce and Erica Wagner, “Using natural language processing to improve eRulemaking”, *In Proceedings of Digital Government Research*, 2006.
- [5] Andrew B. Goldberg, Jerry Zhu, and Stephen Wright, “Dissimilarity in graph-based semi-supervised classification”, *In Artificial Intelligence and Statistics (AISTATS)*, 2007.
- [6] S. Argamon, M. Koppel & G. Avneri, “Routing documents according to style”, *In First international workshop on innovative information systems*, 1998.
- [7] B. Kessler, G. Nunberg, & H. SchÄutze, “Automatic detection of text genre”, *In Proceedings of the 35th ACL/8th EACL*, pages 32–38, 1997.
- [8] P. Turney, “Thumbs up or thumbs down? semantic orientation applied to unsupervised classification of reviews”, *In Proceedings of the 40th Annual Meeting on Association for Computational Linguistics (ACL-02)*, Pennsylvania, Philadelphia, pages 417–424, 2002.
- [9] J. W. Pennebaker, M. E. Francis and R. J. Booth, “Linguistic Inquiry and Word Count”, *LIWC (2nd ed.)*, 2004.
- [10] Pang, Bo, Lee, & S. Vaithyanathan, “Thumbs up? Sentiment classification using machine learning techniques”, *In Proceedings of the conference on empirical methods in natural language processing (EMNLP)*, pages 79–86, 2002.

- [11] Yorick Wilks and Mark Stevenson, “The grammar of sense: Using part-of-speech tags as a first step in semantic disambiguation”, *Journal of Natural Language Engineering*, 4th edition, pages 135–144, 1998.
- [12] Tony Mullen and Nigel Collier, “Sentiment analysis using support vector machines with diverse information sources”, *In Proceedings of the Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 412–418, July 2004.
- [13] Ellen Riloff, Janyce Wiebe, and Theresa Wilson, “Learning subjective nouns using extraction pattern bootstrapping”, *In Proceedings of the Conference on Natural Language Learning (CoNLL)*, pages 25–32, 2003.
- [14] Jin-Cheon Na, Haiyang Sui, Christopher Khoo, Syin Chan, and Yunyun Zhou, “Effectiveness of simple linguistic processing in automatic sentiment classification of product reviews”, *In Conference of the International Society for Knowledge Organization (ISKO)*, pages 49–54, 2004.
- [15] Theresa Wilson, Janyce Wiebe, and Paul Hoffmann, “Recognizing contextual polarity in phrase-level sentiment analysis”, *In Proceedings of the Human Language Technology Conference and the Conference on Empirical Methods in Natural Language Processing (HLT/EMNLP)*, pages 347–354, 2005.
- [16] V. Hatzivassiloglou & K. R. McKeown, “Predicting the semantic orientation of adjectives”, *In Proceedings of the 35th annual meeting of ACL*, 1997.
- [17] V. Hatzivassiloglou, & J. Wiebe, “Effects of adjective orientation and gradability on sentence subjectivity”, *In Proceedings of the 18th international conference on computational linguistics*, 2000.
- [18] S. R. Das, & M. Y. Chen “Sentiment parsing from small talk on the web”, *In Proceedings of the 8th Asia Pacific finance association annual conference*, 2002.
- [19] J. Wiebe, T. Wilson, & M. Bell, “Identifying collocations for recognizing opinions”, *In Proceedings of the ACL/EACL workshop on collocation*, 2001.
- [20] C. Fellbaum (Ed.), “*WordNet: An electronic lexical database*”, Cambridge, MA: MIT Press, 1997.

- [21] M. Hu, & B. Liu, “Mining opinion features in customer reviews”, *In AAAI*, pages 755–760, 2004.
- [22] J. Kamps, & M. Marx, “Words with attitude”, *In Proceedings of the first international conference on global WordNet*, CIIL, Mysore, India, pages 332–341, 2002.
- [23] S. Kim, E. Hovy, “Identifying and analyzing judgment opinions”, *In Proceedings of HLT/NAACL*, New York City, 2006.
- [24] P. Turney, & M. L. Littman, “Measuring praise and criticism: Inference of semantic orientation from association”, *ACM Transactions on Information Systems*, pages 315–346, 2004.
- [25] A. Esuli, & F. Sebastiani, “Determining the semantic orientation of terms through gloss classification”, *In Proceedings of CIKM-05, the ACM SIGIR conference on information and knowledge management*, Bremen, DE, 2005.
- [26] A. Andreevskaia & S. Bergler, “Mining WordNet for a fuzzy sentiment: Sentiment tag extraction from WordNet glosses” *In Proceedings EACL-06*, Trento, Italy, 2006.
- [27] M. Gamon & A. Aue, “Automatic identification of sentiment vocabulary exploiting low association with known sentiment terms” *In Proceedings of the ACL workshop on feature engineering for machine learning in NLP*, Ann Arbor, pages 57–64, 2004.
- [28] A. Kennedy, & D. Inkpen, “Sentiment classification of movie reviews using contextual valence shifters”, *Computational Intelligence*, 22(2), pages 110–125, 2006.
- [29] S. Ait-Mokhtar, J. P. Chanod, & C. Roux, “Robustness beyond shallowness: Incremental deep parsing”, *Natural Language Engineering*, 8(2–3), pages 121–144, 2002.
- [30] C. Whitelaw, S. Argamon, & N. Garg, “Using appraisal taxonomies for sentiment analysis”, *In Proceedings of the first computational systemic functional grammar conference*, University of Sydney, Sydney, Australia, 2005.
- [31] P. Beineke, T. Hastie, S. Vaithyanathan, “The sentimental factor: Improving review classification via human-provided information”, *In Proceedings of the 42nd ACL conference*, 2004.

[32] Sentence polarity dataset v1.0, Online available at - <http://www.cs.cornell.edu/people/pabo/movie-review-data/>

[33] Mostafa Al Masum Shaikh, Helmut Prendinger and Mitsuru Ishizuka, "Assessing Sentiment of Text by Semantic Dependency and Contextual Valence Analysis", *In Proceedings of the International Conference of Affective Computing and Intelligent Interface (ACII-2007)* Lisbon, Portugal, Springer LNCS 4738, pages 191-202, 2009.

[34] Dan Klein and Christopher D. Manning, "Accurate Unlexicalized Parsing" *In Proceedings of the 41st Meeting of the Association for Computational Linguistics*, pages 423-430, 2003.

[35] Marie-Catherine de Marneffe, Bill MacCartney and Christopher D. Manning, "Generating Typed Dependency Parses from Phrase Structure Parses", *In LREC 2006*.