博士学位請求論文

An Analytical Approach for Affect Sensing from Text

テキストからの感情センシングのための解析的アプローチ

By

Mostafa Al Masum Shaikh
(48-57405)

A THESIS SUBMITTED TO

THE GRADUATE SCHOOL OF INFORMATION SCIENCE AND TECHNOLOGY

IN PARTIAL FULFILMENT OF THE REQUIREMENTS FOR THE

DEGREE OF Doctor of Philosophy

At the

University of Tokyo

Thesis Supervisor
Professor Mitsuru Ishizuka
(石塚　満　教授)

東京大学
THE UNIVERSITY OF TOKYO

Abstract

Studying the relationship between natural language and affective information as well as assessing the underpinned affective meaning of natural language are becoming crucial for improving human computer interaction. The area of such interactive applications is numerous and varied, ranging from categorizing newsgroup flame and augmenting search engine responses to analysis of public opinion trends towards a particular fact or entity and customer feedback. Text is not only an important medium to describe facts and events, but also to effectively communicate information about the writer's positive or negative sentiment underlying an opinion, or to express an affective or emotional state, such as happy, fearful, surprised, and so on. We consider sentiment assessment and emotion sensing from text as two different problems. Classifying the tone of the communication as generally positive or negative is considered as the task of sentiment assessment and recognition of particular emotion(s) being expressed is the task of emotion sensing. Therefore, the thesis first presents an analytical approach to sentiment assessment, i.e., the recognition of negative or positive valence of a sentence and then explains how a well-founded emotion model has been implemented for recognition of emotions. For the purpose of sentiment assessment from text, we perform semantic dependency analysis on the semantic verb frame(s) of each sentence, and then apply a set of rules to each dependency relation to calculate the contextual valence of the words used in the sentence. By employing a domain-independent, rule-based approach our system is able to automatically identify sentence-level sentiment. A linguistic tool called 'SenseNet' has been developed to recognize sentiments in text, and to visualize the detected sentiments. We conducted several experiments with a variety of datasets containing data from different domains. The obtained results indicate significant performance gains over existing state-of-the-art approaches. Emotions expressed in natural language are very often expressed in subtle and complex ways, presenting challenges which may not be easily addressed by simple text categorization approaches such as 'n-gram' or 'keyword identification' approaches. Numerous approaches have already been employed to “sense” affective information from text; but none of those ever employed the OCC emotion model – an influential theory of the cognitive and appraisal structure of emotion. The OCC model derives twenty-two emotion types and two cognitive states as consequences of several cognitive variables. This thesis therefore describes how to relate cognitive variables of the emotion model to linguistic components in text, in order to achieve emotion recognition for a much larger set of emotions than handled in comparable approaches. In particular, we provide tailored rules for textural emotion recognition, which are inspired by the rules of the OCC emotion model. Hereby, we clarify how text components can be mapped to specific values of the cognitive variables of the emotion model. The resulting linguistics-based rule set for the OCC emotion types and cognitive states allow us to determine a broad class of emotions conveyed by text.

Acknowledgements

I would like to express my sincere gratitude to Professor Mitsuru Ishizuka and Dr. Helmut Prendinger who were the persons that helped me the most by their guidance and prudent advice while completing this thesis. Prof. Ishizuka as my supervisor took care of me and guided me during all the years of my doctoral studies. Dr. Prendinger provided me with necessary information and advice contributing to my success.

The laboratory of Prof. Ishizuka is a very intellectually stimulating environment with many people coming from different countries and cultures, having different experiences and opinions. During my stay in the laboratory, I had occasions for collaboration on the scientific and cultural level. I wish to thank Hiroshi Dohi, Meiko Fujita, Adam Jatowt, Naoaki Okazaki, Arturo Nakasone, Jie Yang, Eiko Kin, Tadanobu Furukawa, Danushka Bollegala, Werner Breitfuss, Tomofumi Takayama, Yulan Yan, Alena Neviarouskaya, Manuel M. Martinez, A. S. M. Mahbub Morshed, Francisco Tacoa, Bahlul Haider, Fahim Ferdous Khan, Fahim Kawser, Ashequl Qadir, Md. Tawhidul Islam, Russell Rehman, Shahidul Islam, Zeenatul Bashar, Sarkar Abul Bashar, Nayeem Mahmud, Shakil Ahmed and many others for their help and assistance during my research.

Finally I would like to express my thanks to Japanese Ministry of Education, Culture, Sports, Science and Technology for enabling me to come and study in Japan by providing financial assistance in terms of “Monbukagakusho Scholarship”.

To the persons whom I love most – my parents and my wife

Table of Contents

Abstract i
Acknowledgements iii
List of Tables vii
List of Figures and Illustrations viii

CHAPTER ONE: INTRODUCTION 1
1.1 Structure of Thesis 2
1.2 Sentiment and Emotion in Text 3
1.3 Research Theme 4
1.4 Domain Knowledge 5
1.5 Core Feature of this Research 7

CHAPTER TWO: SENTIMENT ANALYSIS OF TEXT 10
2.1 Reviews of Existing Approaches 10
2.2 Topics Ignored and Our Focus 16
2.3 Summary of Our Approach 18

CHAPTER THREE: LINGUISTIC RESOURCES AND SENSENET 20
3.1 SenseNet Architecture 20
3.2 Semantic Parser 22
3.3 Developing Affective Lexica 23
3.3.1 The Knowledgebase: 25
3.3.2 Scoring a list of Verbs, Adjectives and Adverbs 26
3.3.3 Scoring of Nouns 28
3.3.4 Scored-list of Named Entity 30
3.4 Contextual Valence Assessment 32
3.5 Sentiment Assessment 37
3.6 SenseNet GUI 40

CHAPTER FOUR: SENSENET EVALUATION 44
4.1 The Datasets 44
4.2 Sentence Level Comparisons 47
4.3 Paragraph Level Comparisons 49
4.4 Evaluating Individual Components of SenseNet 53
4.5 Comparison to a State-of-the System 57
4.6 Discussion 58
4.7 Conclusion 63

CHAPTER FIVE: EMOTION ANALYSIS OF TEXT 64
5.1 Necessity of a New Approach 64
5.2 The OCC Emotion Model 65
5.2.1 Why OCC Model? 66
5.2.2 Characterization of the OCC Emotions 68
5.3 Implementation of the OCC Model in Linguistic Realm 71

5.3.1 Linguistic Resources 71
5.3.2 Assigning Values to the Variables 73
5.3.3 The Rules of the OCC Emotion Types 81
5.4 Walk-Through Examples for Emotion Recognition 84
5.5 Evaluation and Discussion 88
5.6 Summary 90

CHAPTER SIX: DEVELOPED APPLICATIONS 92
6.1 SenseNet 92
6.2 ASNA: Affect Sensitive News Agent 95
6.3 ESNA: Emotion Sensitive News Agent 97
6.4 Online System for Textual Affect Sensing 105

CHAPTER SEVEN: SUMMARY AND CONCLUSION 108

REFERENCES 112

APPENDIX A: PSEUDO CODE 119

APPENDIX B: EXPERIMENTAL RESULT FOR DATASET A 121

PUBLICATION LIST OF THE AUTHOR 123

List of Tables

Table 3.1 Triplet output of Semantic Parsing for the sentence given above. 23

Table 3.2 Sample list of verbs with associated Prior Valence. 28

Table 3.3 Sample list of scored named entities.. 31

Table 3.4 Symbols used in SenseNet browser. 43

Table 4.1 Input datasets 46

Table 4.2 Accuracy results obtained for Dataset B using different approaches. 48

Table 4.3 Summary of several systems that experimented with Dataset C 50

Table 4.4 Experimental result using the Dataset C 52

Table 4.5 Experimenting with different models of the system using all the datasets....... 55

Table 4.6 Accuracy Comparison Metrics between *EmpathyBuddy* and *SenseNet* 58

Table 5.1 the variables (i.e., cognitive variables) of the OCC Emotion Model............... 68

Table 5.2 the definitions of the rules for the OCC emotion types 69

Table 5.3 Semantic verb-frames output by the semantic parser for *"My mother presented me a nice wrist watch on my birthday and made delicious pancakes."* ... 72

Table 5.4 Recognizing the OCC emotions from the sentence *"I didn't see John for the last few hours; I thought he might miss the flight but I suddenly found him on the plane."* 85

Table 5.5 Preliminary experimental result of the two systems 90

Table 6.1 Excerpts from database of ranked words 93

Table 7.1 the summary of experimental results using different datasets 108

List of Figures and Illustrations

Figure 3.1 Architecture of SenseNet.. 21

Figure 3.2 ConceptNet output for the concept ‘rocket’ 25

Figure 3.3 Interface and Sample Output of SenseNet Browser 42

Figure 4.1 Relationship between the ‘Neutral Range’ of the system to signal neutrality of a sentence and other system performance measures namely, Accuracy, Average Precision, Recall, and F-Score for three classes. 48

Figure 5.1 The OCC Emotion Model 67

Figure 6.1 SenseNet GUI 94

Figure 6.2 Architecture of ASNA 96

Figure 6.3 ASNA News Browser 96

Figure 6.4 Architecture of Emotion Sensitive News Agent (ESNA) 98

Figure 6.5 User Interface to Customize Sentiment towards Named-Entities 100

Figure 6.6 the summary of user study 104

Figure 6.7 user-interface of the online system 105

Figure 6.8 the output interface of the online system 106

Figure 6.9 Survey form 107

Chapter One: Introduction

There is now plenty of evidence in neuroscience and psychology about the importance of emotional intelligence for the overall human performance in tasks such as rational decision-making, communicating, negotiating, and adapting to unpredictable environments. As a result, people can no longer be modeled as pure goal-driven, task-solving agents: they also have emotive reasons for their choices and behavior which (more often than not) drive rational decision-making [Mandler, 1975]. In holistic view the research is aimed at giving computer programs skills of emotional intelligence, including the ability to recognize, model, and understand human emotion, to appropriately communicate emotion, and to respond to it effectively. The new discipline coined as “Affective Computing” [Picard 1997] investigates the basics of human emotion and emphasizes both the physiological and cognitive aspects of emotion. The Affective Computing community developed several mechanisms for emotion sensing, including the processing of various physiological signals obtained from wearable sensors. Early work in Affective Computing emphasized the physiological and behavioral aspects of emotion, for instance, by analyzing biometric sensor data, prosody, posture, and so on. More recently, the ‘sensing’ of emotion from text gained increased popularity, since textual information provides a rich source of the expression of human affective state. The words we use reflect who we are and hence the word choice of one’s writing serves as a key to one’s personality, social situation and affective or attitudinal information conveyed through texts. Furthermore, people most naturally interact with their computers in a social and affectively meaningful way, just like with other people [Reeves and Nass, 1998]. These observations have created an expectation that the future human computer interaction (HCI) is in themes such as emotions, entertainment, attention, motivation, e-learning etc. So studying the relationship between natural language and affective information as well as assessing the underpinned affective qualities of natural language are becoming crucial for improving interaction with users. Specifically, this research is devoted to exploring different techniques to recognize positive and negative opinion, or favourable and unfavourable sentiments towards specific subjects occurring in natural language texts.

1.1 Structure of Thesis

This thesis is composed of seven chapters and two appendices, which provide background to this research, describe the core methodologies, demonstrate results of this work, describe the developed applications, and enlist pseudo codes of the approach discussed. The contents of each chapter are outlined below.

- Chapter one: This part is a general introduction to the topic. Since the research topic is multi-disciplinary, first the contribution and background knowledge obtained from different knowledge domains are discussed. Then the core features of this research are pointed out.
- Chapter two: In this chapter, the current state of the art approaches for sentiment analysis from texts have been discussed by pointing to the limitations of those. Finally, our approach is explained from the viewpoint of considering the previously ignored topics for the task of sentiment analysis of text.
- Chapter three: This chapter explains the core approach of this research. How different lexical resources have been developed and then employing several rules how an input text can be considered as an analytical model have been explained with examples. Our developed application, SenseNet, assesses an input text numerically in order to know whether the input text carries a negative or positive sense. The implementation detail of SenseNet is discussed in this chapter.
- Chapter four: This chapter contains experimental results for different standard datasets for the task of sentiment analysis. Different types of system evaluation are done and the chapter concluded with a discussion on obtained results and failure analysis.

- Chapter five: Though all emotions can be seen as positive or negative, this chapter extends the idea of recognizing more fine-grained named emotions (e.g., happy, sad, anger etc.). Towards this point how a well-founded emotion model (i.e., OCC emotion model taken from Cognitive Psychology) can be implemented in linguistic realm has been discussed. This is completely a new contribution that came out of this research.
- Chapter six: Grounding the developed theories and methodologies several applications are developed. In this chapter the developed applications are discussed in terms of their architectures, functional steps and graphical user interfaces.
- Chapter seven: This chapter contains summary and conclusions of the studies in sentiment and affect sensing from text.
- Appendix A: It contains the pseudo code of the algorithm for sentiment sensing from text.
- Appendix B: It contains the detail experimental result of one of the datasets.

1.2 Sentiment and Emotion in Text

We consider sentiment assessment and affect sensing from text as two different problems. We refer the phrases like ‘sentiment analysis’, ‘opinion mining’, ‘mood analysis’, ‘trend analysis’, to the problem of sentiment assessment aiming to determine whether a positive, negative or neutral attitude of a speaker or a writer has been communicated with respect to some topic. But, by the phrases like ‘emotion recognition’, ‘affect sensing’, ‘emotion analysis’, we refer to the problem of affect sensing aiming to detect the underlying emotions being communicated grounding on the theory of emotions from cognitive psychology. Hence, the research on affect sensing from text addresses certain aspects of subjective opinion, including the identification of different emotive dimensions and the classification of text primarily by the opinions (e.g., negative or positive) expressed therein, and then the emotion affinity (e.g., happy, sad, anger, etc) of textual information. Emotions are very often expressed in subtle and complex ways in natural language, and hence there are challenges which may not be easily addressed by simple text categorization approaches such as n-gram, lexical affinity or keyword identification based methods. It can be argued that analyzing affect in text is an "NLP"-complete problem [Shanahan et al., 2006] and interpretation of text varies depending on audience, context, and world knowledge.

1.3 Research Theme

The philosophy of developing emotionally intelligent computer program has been discussed by many researchers like Turing [Turing 1950], Weizenbaum [Weizenbaum 1966], Wallace [Wallace 2004], Minsky [Minsky, 2006], and so on. With the similar notion, this research also aims to incorporate emotional intelligence to the computer program to identify or classify the sentiment or affect communicated by a sentence (or paragraph). In particular, the research attempts to answer the following questions:

1. What can be understood from a textual description: For example, for the sentences like, *(i) I have an exam tomorrow, but I am not confident enough. (ii)The concert was really wonderful. (iii) The employee, suspecting he was no longer needed, he might be fired.,* what we understand in terms of topic (or theme) of the given sentences; how a processing model of our so-called understanding can be developed so that a computer program can process it for various purposes like, question answering, topic detection, text summarization, text mining, etc.

2. What meaning can be assessed from a textual description: For example, the sentences like, *"I have several assignments pending. Last week I visited Kyoto; it was an excellent trip."*, give positive meaning and the sentences like, *"I am little annoyed, I'm fed up. I can hardly complete the experiment on time."*, give negative meaning. Therefore, this research aims to find how such senses of positive and negative meaning can be assessed by a computer program.

3. How sentiment can be assessed from a given text: The interest in sentiment based automated text categorization has increased with the availability of large amount of text on the Internet. The applications range from document organization (e.g., positive or negative review classification),

automatic document indexing for information retrieval, text or email filtering, word sense disambiguation, categorization of web pages, news-article classification, review and, most recently, spam filters. This research targets to develop an analytical model of any given text to assess the negative or positive sentiment of the text by a method of numerical analysis.

4. What emotions are usually expressed in written text and how to detect: In our opinion, emotions can be considered as more fine grained sentiment expressing either a negative or positive feeling. For example, one can say that the sentences, *"The attack killed three innocent civilians. The woman was proud of saving the life of a drowning child. Mary was filled with affection as she gazed at her newborn infant."*, refer to emotions like 'sad', 'pride', and 'love', respectively. How do we know this? It is true that there are certain linguistic clues and tokens for expressing emotions and moreover we have commonsense knowledge for affective concepts to deduce emotion from a linguistic interpretation. Hence, different approaches have already been employed to "sense" emotion or affect, especially from the text, but none of those ever considered the valenced based cognitive and appraisal structure of emotions that this research is focusing on.

1.4 Domain Knowledge

This research, grounded in findings from cognitive science, psycholinguistics, computational linguistic, knowledge representation, and natural language processing, is to develop computer programs for recognizing, measuring, modeling, reasoning about, and responding to affect. Thus, we are particularly interested to develop new algorithms, computer programs, and theories that might enable to recognize the emotion/affect from textual information providing new forms of machine intelligence. In order to explore the answers to the aforementioned questions the research encircles knowledge from the following domains.

a) Cognitive Science: In order to know about the theory of emotions, the cognitive and appraisal structure of emotions the research reviewed the literatures from cognitive science domains. The key ideas obtained from Cognitive Science domain are,

- Valenced based reaction towards an event: the idea is about assigning a valenced reaction towards an event. Based on the value and polarity of the valence value one can decide whether an event is positively or negatively emotive or non-emotive.
- Rules of different emotions: Cognitive science also suggests some cognitive variables which are responsible to characterize different emotion types. In this research we consider those variables to develop the rules of different emotions.

b) Psycholinguistic: We concede that there are certain linguistic clues and token words to express sentiment and emotion by text. But according to a linguistic survey [Pennebaker et al. 2003], only 4% of the words used in written texts carry affective content. This finding shows that using affective lexicons is not sufficient to recognize affective information from text. It also indicates the difficulty of employing methods like machine learning, keyword spotting, or lexical affinity. Hence the remarks from this domain suggest us to investigate the problem with deeper outlook considering the semantics of language, context and affective meaning of the event(s) described by text.

c) Computational linguistics: Since we realized that the problem of sentiment analysis or affect sensing from text cannot be solved by simply finding the negative and positive keywords or phrases from the given sentence, we investigated the semantic structure of natural language. The key ideas obtained from this domain are,

- Semantic Verb Frame: We consider each semantic verb frame found in a sentence as an event which eventually underpins or causes to a sentiment or emotion.
- Semantic Orientation of Words: It means whether a word or phrase is semantically oriented with a positive or negative connotation. Employing several linguistic resources we assigned

either a positive or negative value to a bag of words which servers as the knowledgebase of the system.

d) Knowledge Representation: While developing the knowledgebase of the system we realized that it would be practical and robust if we could represent the knowledgebase in a flexible, and easy to extend manner. Therefore, the key ideas obtained from the domain are,

- Use of Commonsense Knowledgebase: We have incorporated a mechanism to utilize commonsense knowledgebase to assign a score to a previously non-scored word or phrase and adopted a technique to extend the system's knowledgebase automatically thereby.
- Real-world knowledge representation: We admit that emotions are often related with certain real-world entities likes some real person or something in the real-world. Applying commonsense knowledge we cannot make any significant assessment for those named-entities. Hence in this research we have devised a mechanism to extract and represent the knowledge of real-world named-entities.

e) Natural Language Processing: The use of natural language processing is non-trivial in this research. Since this research aims to deal with texts, the basic concept taken from this domain are,

- Semantic Parser: We have developed a semantic parser that produces the computational model of the input sentence.
- Contextual Valence: We have developed the rules to assign contextual valence values to the linguistic tokens and cognitive variables by using the knowledgebase. This is the main difference between our approach and other known approaches like keyword spotting, and traditional machine learning algorithms.

1.5 Core Feature of this Research

Recognizing or "sensing" affective information would benefit the development of text-based user interfaces since the words people use to express their feelings can be important clues to their mental, social, and physical state [Pennebaker et al. 2003]. Examples of such applications include the affective text analyzer ([Hu and Liu 2004][Mihalcea and Liu 2006][Shaikh et al. 2006b, 2007a, 2007b][Knobloch et al. 2004][Sentiment!][Pennebaker et al. 2001]), the affective email-client ([Liu et al 2003]); empathic chat ([Zhe and Boucouvalas 2002]); information and tutoring tools ([Rosis and Grasso 2000]); computational humor ([Stock and Strapparava 2003], [Strapparava et al. 2007]); affective lexicon ([Valitutti et al. 2004]); affective information recognizer ([Mihalcea and Liu 2006][Kim and Hovy 2006][Koppel and Shtrimberg 2004]); and psycholinguistic analysis ([Pennebaker et al. 2003][Kamps and Marx 2002]). We expect that more are likely to appear with the increase of textual resources on the internet (e.g. blogs, reviews, etc.).

We are interested in identifying positive and negative sentiment as well as emotions (e.g. happiness, sadness etc.) conveyed through text. Our approach is based on the semantic relationship between textual components in a sentence, and the computation of contextual valence of the used words. The core feature of our approach is to provide a more detailed analysis of text, so that named, individual emotions can be recognized, rather than dichotomies like positive-negative. From a technical viewpoint, there are four main factors that distinguish our work from other methods of textual emotion sensing. First, we have integrated semantic processing on the input text by functional dependency analysis based on semantic verb frames [Fellbaum 1999][Johnson et al. 2006]. Second, we utilize cognitive and commonsense knowledge resources to assign prior valence or semantic orientation (SO) [Hatzivassiloglou and McKeown 1997] to a set of words that leverages scoring for any new words. Third, instead of using any machine learning algorithm or corpus support, a rich set of rules for calculating the contextual valence of the words have been implemented to perform word-level sentiment (i.e., positive, negative or neutral) disambiguation and assign an overall valence to the whole sentence(s) by applying dedicated rules. Finally, we apply a cognitive theory of emotions known as the OCC model

[Ortony et al. 1988] which is implemented to distinguish several emotion types identified by assessing valanced reactions to events, agents or objects, as described in text. This paradigm of content analysis allows assessing sentiments from texts of any genre (e.g. movie or product review, news articles, blogs, etc.) at the sentence level.

Chapter Two: Sentiment Analysis of Text

Sentiment has been studied at three different levels: word, sentence, and document level. There are methods to estimate positive and negative sentiment of words (e.g., [Turney 2002][Esuli and Sebastiani 2005]), phrases and sentences (e.g., [Kim and Hovy 2006][Wilson et al. 2005]), and documents (e.g., [Hu and Liu 2004][Turney and Littman 2003]). Previous approaches for assessing sentiment from text are based on one or a combination of the following techniques: keyword spotting (e.g.,[Zhe and Boucouvalas 2002]), lexical affinity (e.g., [Valitutti et al. 2004][Kim and Hovy 2005]), statistical methods (e.g., [Pennebaker et al. 2001, 2003]), a dictionary of affective concepts and lexicon (e.g., [Rosis and Grasso 2000]), commonsense knowledgebase (e.g., [Mihalchea and Liu 2006][Liu et al. 2003]), fuzzy logic (e.g., [Subasic and Huettner 2001]), knowledgebase from facial expression (e.g., [Fitrianie and Rothkrantz 2006]), machine learning (e.g., [Kim and Hovy 2006], [Wiebe et al. 2005], [Sebastiani 2002], [Riloff et al. 2003], [Turney and Littman 2003], [Pang and Lee 2005]), domain specific classification (e.g., [Nasukawa and Yi 2003], [Koppel and Shtrimberg 2004]), and valence assignment ([Shaikh et al. 2007a, 2007b], [Wilson et al. 2005], [Polanyi and Zaenen 2004]). Some researchers proposed machine learning methods to identify words and phrases that signal subjectivity. For example, [Wiebe and Mihalcea 2006] stated that subjectivity is a property that can be associated with word senses, and hence word sense disambiguation can directly benefit subjectivity annotations. [Turney 2002], [Hatzivassiloglou and Wiebe 2000], and [Wiebe 2000] concentrated on learning adjectives and adjectival phrases, whereas [Wiebe et al. 2005] focused on nouns. [Riloff et al. 2003] extracted patterns for subjective expressions as well.

2.1 Reviews of Existing Approaches

While various conceptual models, computational methods, techniques, and tools are reported in [Shanahan et al. 2004], we argue that the current work for sensing affect communicated by text is incomplete and available methods need improvement [Shaikh et al. 2006a]. The assessment of affective content is inevitably subjective and subject to considerable disagreement. Yet the interest in sentiment or affect based text categorization is increasing with the large amount of text becoming available on the Internet. Different techniques applied to sense sentiment and emotion from the text are briefly described in the following paragraphs. According to a linguistic survey [Pennebaker et al. 2003]; only 4% of the words used in written texts carry affective content. This finding shows that using affective lexicons is not sufficient to recognize affective information from text. It also indicates the difficulty of employing methods like machine learning, keyword spotting, or lexical affinity.

Keyword Spotting: It is the most naïve approach and probably the most popular because of its convenience and faster computation. Text is classified into affect categories based on the presence of fairly unambiguous affect words like “distressed”, “furious”, “surprised”, “happy”, etc. Elliott’s Affective Reasoner [Elliott 1992], for example, watches for 198 affect key-words (e.g. distressed, enraged), plus affect intensity modifiers (e.g. extremely, somewhat, mildly), plus a handful of cue phrases (e.g., “did that”, “wanted to”). Ortony’s Affective Lexicon [Ortony et al. 1988] provides an often-used source of affect words grouped into affective categories. The weaknesses of this approach lie in two areas: poor recognition of affect when negation is involved, and reliance on surface features. About its first weakness: while the approach will correctly classify the sentence, *“John is happy for his performance,”* as being happy, it will likely fail on a sentence like *“John isn’t happy for his performance.”* About its second weakness: the approach relies on the presence of obvious affect words which are only surface features of the prose. In practice, a lot of sentences convey affect through underlying meaning rather than affect adjectives. For example, the text: *“The employee, suspecting he was no longer needed, he might* *be asked to find another job."* certainly evokes strong emotions, but use no affect keywords, and therefore, cannot be classified using a keyword spotting approach.

Lexical Affinity: This method is relatively more sophisticated than keyword spotting. Detecting more than just obvious affect words, the approach assigns arbitrary words a probabilistic "affinity" for a particular emotion. For example, "accident" might be assigned a 75% probability of being indicating a negative affect, as in "car accident," "hurt by accident." These probabilities are usually trained from linguistic corpora. Though often outperforming pure keyword spotting, we see two problems with the approach. First, lexical affinity, operating solely on the word-level, can easily be tricked by sentences like *"I avoided an accident,"* (negation) and *"I met my girlfriend by accident"* (other word senses). Second, lexical affinity probabilities are often biased toward text of a particular genre, dictated by the source of the linguistic corpora. This makes it difficult to develop a reusable, domain-independent model.

Statistical Natural Language Processing: This approach has been applied to the problem of textual affect sensing. By feeding a machine learning algorithm a large training corpus of affectively annotated texts, it is possible for the system to not only learn the affective valence of affect keywords as in the keyword spotting approach, but such a system can also take into account the valence of other arbitrary keywords (like lexical affinity), punctuation, and word co-occurrence frequencies. Statistical methods such as latent semantic analysis (LSA) have been popular for affect classification of texts, and have been used by researchers on projects such as Goertzel's Webmind [Goertzel et al. 2000]. However, statistical methods are generally semantically weak, meaning that, with the exception of obvious affect keywords, other lexical or co-occurrence elements in a statistical model have little predictive value individually. As a result, statistical text classifiers only work with acceptable accuracy when given a sufficiently large text input. So while these methods may be able to affectively classify the user's text on the page or paragraph-level, they will not work well on smaller text units such as sentences. Statistical methods are suited for a psycholinguistic analysis (e.g., [Pennebaker et al. 2003]) of persons' attitudes, social-class, standards etc. from documents rather than individual sentences.

Hand-Crafted Models: In the tradition of Schank and Dyer, among others, affect sensing is seen as a deep story understanding problem. Dyer's DAYDREAMER models [Dyer 1987] affective states through hand-crafted models of affect based on psychological theories about human needs, goals, and desires. Because of the thorough nature of the approach, its application requires a deep understanding and analysis of the text. The generalizability of this approach to arbitrary text is limited because the symbolic modeling of scripts, plans, goals, and plot units must be hand-crafted, and a deeper understanding of text is required than what the state-of-the-art in semantic parsing can provide.

Commonsense based approach: The latest attempt [Liu et al 2003], can categorize texts into a number of emotion groups such as the six so-called "basic" emotions (i.e., happy, sad, anger, fear, disgust, and surprise) based on "facial expression variables" proposed by Paul Ekman. In our view, this emotion set is not optimal for classifying emotions expressed by textual information. Most importantly, those 'expression' based emotion types do not consider the cognitive antecedents of human emotions, or their relation to humans' believes, desires, and intentions (by reference to the well-known BDI model for autonomous agents). In [Liu et al. 2003] the authors utilize a knowledge-base of commonsense that represents a semantic network of real-world concepts associated with the basic emotion categories. Hence, for the input *"My husband just filed for divorce and he wants to take custody of my children away from me."*, the system outputs it as a "sad" sentence, but it fails to sense the emotion correctly from the input like *"It is very* *difficult to take bad picture with this camera."*, and classifies it as a "sad" sentence as well. The limitation of this approach is that it does not consider the semantic relationship between the linguistic components of the input text and the context in which the words occur.

Fuzzy logic: Fuzzy logic assesses an input text by spotting regular verbs and adjectives, without processing their semantic relationships. Here, the verbs and adjectives have pre-assigned affective categories, centrality and intensity (for details see [Subasic and Huettner 2001]). Like lexical affinity based approaches, this method cannot adequately analyze smaller text units such as sentences, for instance, *"The girl realized that she won't be called for the next interview,"* where no affective word occurs.

Knowledge-base approach: The knowledge-base approach in [Fitrianie and Rothkrantz 2006] investigates how humans express emotions in face-to-face communication. Based on this study, a two-dimensional (pleasant/unpleasant, active/passive) affective lexicon database and a set of rules that describes dependencies between linguistic contents and emotions is developed. In our opinion, this approach is very similar to keyword-spotting and therefore not suitable for sentence-level emotion recognition.

Machine learning: Sentences typically convey affect through underlying meaning rather than affect words, and thus evaluating the affective clues is not sufficient to recognize affective information from texts. However, machine learning approaches (e.g., [Kim and Hovy 2006], [Wiebe et al. 2005], [Sebastiani 2002], [Strapparava et al. 2007], [Koppel and Shtrimberg 2004]) typically rely on affective clues in analyzing a corpus of texts. This approach works well when a large amount of training data of a specific domain of interest (e.g., movie review) is given. It requires, however, special tuning on data-sets in order to optimize domain specific classifiers.

Although some researchers (e.g., [Wiebe 2000], [Turney and Littman 2003], [Hu and Liu 2004], [Polanyi and Zaenen 2004]) proposed machine learning methods to identify words and phrases that signal subjectivity, these methods are not suitable for sentence-level emotion classification for not using of an emotion annotated corpus of every domain. Moreover, machine learning based approaches fail to incorporate semantic structure of sentences for rule-driven semantic processing of the words (e.g., contextual valence) used in a sentence.

Valence assignment: A number of researchers have explored the possibility of assigning prior valence (i.e., positive or negative value) to a set of words (e.g., [Polanyi and Zaenen 2004], [Wilson et al. 2005], [Kamps and Marx 2002], [Hatzivassiloglou and McKeown 2002]). By contrast, in the system [Shaikh et al. 2006b, 2007a, 2007b] we begin with a lexicon of words with prior valence values using WordNet [Fellbaum 1999] and ConceptNet [Liu and Singh, 2004], and assigns the contextual valence of each semantic verb frame (described in [Fellbaum 1999], [Johnson et al. 2006]) by applying a set of rules. [Kim and Hovy 2006] and [Wilson et al. 2005] count the prior polarities of clue instances of the sentence. They also consider local negation to reverse valence; yet they do not use other types of features (e.g., semantic dependency) contained in the approach mentioned by us. [Nasukawa and Yi 2003] compute the contextual valence of sentiment expressions and classify expressions based on manually developed patterns and domain specific corpora. The use of domain specific corpora for sentiment classification of text has shown very promising results regarding sentiment analysis of product reviews and blogs, but it requires special tuning of data in order to build category-specific classifiers for each text domain (e.g. product review or movie review). The system [Nasukawa and Yi 2003] used a sentiment analysis dictionary having 3,513 entries and instead of analyzing the favorability of the whole context each statement on favorability is extracted. But the system outputs -1 to indicate a negative sentiment (due to shallow understanding) for the sentence *"It's difficult to take a bad picture with this camera."*, whereas this is a positive statement for the object 'camera'.

2.2 Topics Ignored and Our Focus

Semantic Processing of Natural Language: We have noticed that most of the approaches for sentiment analysis or affect sensing applied machine learning algorithms where syntactic patterns of the input sentences have been trained to the machine to recognize the patterns and machine ranked scores have been applied to the words in the recognized pattern to get the overall score of the input sentence. This approach fails to process the sentence like, *"the test was too difficult to fail,"* for not considering the semantic relationship between the words 'difficult' and 'fail'. So our approach is different in the following manners.

1) It is a rule based system. The rules to assign the score to the words in a sentence have been developed according to the syntactic patterns of the sentence.

2) Semantic relationship among the words has been considered while assigning a score to a phrase. For example, for the aforementioned sentence the semantic relationship between the words is interpreted as, "a negative sense qualifying a negative verb with a 'to-dependency' usually refers a positive sense"; and hence such rule can refer a positive sentiment for the sentences like, *"It was hard to kill the monster,"* and *"It is difficult to make bad shots with this camera"*.

Recognizing only six emotions or less: It is also observed that previous attempts, e.g., ([Liu et al. 2003]), have categorized texts into a number of emotion groups such as the six so-called "basic" emotion based on "facial expression variables" proposed by Ekman, which we believe are not adequate for classifying emotions expressed by textual information. In this research we have employed the OCC emotion model – an influential theory of the cognitive and appraisal structure of emotion. The OCC model derives twenty-two emotion types and two cognitive states as consequences of several cognitive variables.

User Model, Real-world knowledge: We believe that sentiment or emotion is idiosyncratic and hence a user model should be incorporated. Moreover emotions are often related with certain real-world entities likes some real person or some entities in the real-world for which the system requires human assigned opinion to assess the context of the given text. In our approach we have incorporated user modeling in terms of asking a user to preset the opinion towards the named-entities in the text. We also made an automatic process of scoring the named-entities employing blogs text from where public opinion towards an input entity can be extracted.

Cognitive and Appraisal Structure of Emotion: It is been observed that all the previous approaches for analyzing texts for affect have commonly employed keyword spotting, lexical affinity, statistical methods, pre-processed models (for storytelling scenario), a dictionary of affective concept and lexicon, or commonsense knowledgebase, but none of those considered the cognitive structure of individual emotions or their appraisal structure. We also believe that sensing affect from linguistic descriptions or text should consider "phenomenal variables", "behavioral variables" and "cognitive variables" to characterize the structure of emotions usually underpinning text. Phenomenal variables are those variables which could be tracked during the compilation of the text to further associate affective/mental state of the user, for example, time spent to compile a line/paragraph, number of typos per minute etc. could signal about the person's attentiveness etc. In psychology, behavioral variables are some sets of specific behaviors that represent specific emotions. In this case we are particularly interested in "cognitive variables" within the linguistic data. Hence we have employed an emotion model which considers emotions as valenced reaction to consequences of events, actions of agents and different aspects of objects and these phenomena could be detected from linguistic data (e.g. email, chat log, customer feedback etc.) to assess affective information of the user. We believe that the particular emotion a person experiences or describes in text on some occasion is determined by the way he/she construes the world. Thus the attempt of using only commonsense knowledge without the Belief, Desire and Intention (BDI) model of the person/agent may not successfully sense the emotion type and intensity of emotion and the variables that influence the affective sense being conveyed. Particularly we would like to apply the cognitive structure of individual emotions and use the term "emotion type" which distinguishes a set of emotions that can be realized by finding and assessing the valenced reactions to events, agents or objects described in the texts as suggested by OCC Model. Our approach emphasizes cognitive principles underlying the experience of emotions and the characterization of each emotion type.

2.3 Summary of Our Approach

A number of researchers (e.g., [Kamps and Marx 2002], [Turney 2002], [Wilson et al. 2005]) have explored the automatic learning of words and phrases with prior positive or negative valence. By contrast, we begin with a lexicon of words by calculating prior valence using WordNet [Fellbaum 1999] and ConceptNet [Liu and Singh 2004], and assign the contextual valence [Polanyi and Zaenen 2004] of phrases by applying a set of dedicated rules. [Kim and Hovy 2006], [Hu and Liu 2004], and [Wilson et al. 2005] 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), as our approach does. [Nasukawa and Yi 2003] classifies the contextual valence of sentiment expressions (as we do) and also expressions that are about specific items based on manually developed patterns and domain specific corpora, whereas our approach is domain independent. Since a valence assignment approach focuses on the contextual aspects of linguistic expressions of attitude, it is suitable for sentence-level sentiment sensing (i.e., good or bad) from texts of any genre with higher accuracy. So we are motivated to apply rules based valence assignment approach to assess the input text. First, we integrated semantic processing of input text by performing dependency analysis of semantic verb-frame(s) of each sentence. Second, cognitive and commonsense knowledge resources have been utilized to assign a prior valence to a set of words, which leverage scoring for new words. Third, a set of rules to calculate contextual valence has been implemented to support word sense disambiguation. Fourth, instead of using machine learning or relying on text corpora, we followed a rule-based approach to assess the valence of each semantic verb frame in a sentence, and then assign an overall valence to the whole sentence(s) by applying dedicated rules. Finally, a cognitive theory of emotions known as the OCC model has been tailored to fit into the natural language processing domain to distinguish several emotion types identified by assessing valanced reactions to events, agents or objects, as described in text.

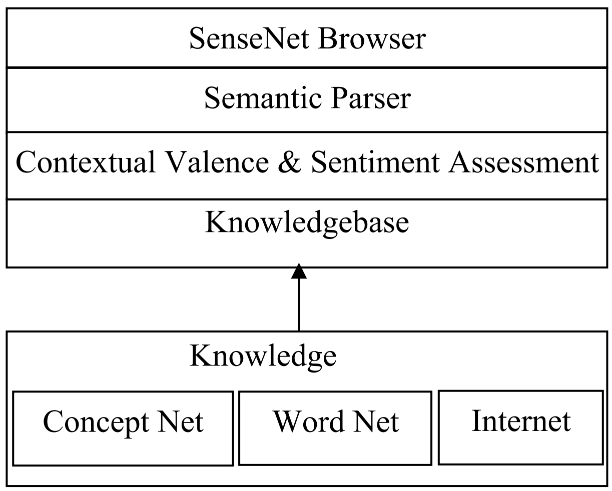
Chapter Three: Linguistic Resources and SenseNet

In general terms the research aims to a linguistic approach to affective computing. 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. This chapter is the core of this research where we frame our approach by describing the architecture of our system and implementation of the system. First we discuss our approach to semantic parsing of the input sentence(s). Then, we explain how different linguistic resources like WordNet [Fellbaum 1999], ConceptNet [Liu and Singh, 2004], Opinmind [Opinmind 2006], etc., have been integrated to build the system's knowledgebase. Next, we explain the formal underpinnings of the rules that compute the valence to indicate the positive, negative or neutral sentiment of the input sentence(s). Finally, based on an example input sentence, we provide a detailed explanation of our algorithm by "walking through" the steps of the algorithm.

3.1 SenseNet Architecture

We concede that the analysis of favourable or unfavourable opinions, or emotion-affinity, is a task requiring emotional intelligence and deep understanding of the textual context, involving commonsense, domain knowledge as well as linguistic knowledge. The interpretation of opinions is usually debatable, arguable, doubtful, subjective and idiosyncratic affair even for humans [Wiebe et al. 2001]. Nevertheless, by proposing SenseNet, we will attempt a computational approach to solve this task. The compositional architecture of SenseNet is indicated in Figure 3.1. SenseNet maintains a knowledgebase by employing three types of knowledge sources: WordNet 2.1, ConceptNet 2.1, and the Internet. A set of rules has been implemented to compute contextual valence and to perform sentiment assessment. The Semantic Parser has been developed employing a language parser [Machinese Syntax] and is utilized to perform semantic processing. The SenseNet browser shows the sentiment of each line of the input text by displaying numerical valence and icons. Subsequent sections explain the components in detail.

Figure 3.1 Architecture of SenseNet



In a linguistic context, as e.g., in WordNet, the sense of a word is a given meaning of that word within a certain context. Similar to WordNet, the term "sense" in SenseNet refers to the contextual sense of each semantic verb-frame(s) (discussed in [Fellbaum 1999], [Johnson et al. 2006]) of a sentence, whereby each sense is represented by a lexical triplet consisting of a subject or agent, a verb, and an object. According to this naming convention, the input sentence *"We have submitted a paper to the conference and we are very optimistic."* involves two senses. They are based on the two verbs "submit" and "be", and associated with two triplets [we, submit, paper] and [we, be, optimistic] respectively. The motivation for creating SenseNet is to utilize several linguistic resources (e.g., Language Parser, WordNet, ConceptNet, etc.) to construct "senses" based on the semantic verb-frames of the input sentence(s) as the computational elements; assess the contextual valence of the sense(s); and finally output a valence to indicate either positive or negative sentiment of the input sentence(s) by a graphical manner.

The system implements a pipelined design with the following phases: Parse, Process, Assess and Visualize. Briefly, the Parse phase implements semantic parsing, i.e., it performs dependency analysis on the words and outputs triplet(s) of subject, verb, and object according to each semantic verb frame of the input sentence(s). In the Process phase, rules are applied to assign contextual valence to the triplet(s). In the Assess phase an overall valence is assigned to each input sentence(s). Finally, the Visualize phase, the SenseNet browser displays the sentiments of the input text using icons and symbols.

3.2 Semantic Parser

For each input sentence, the Semantic Parser module outputs triplet(s) consisting of a subject or agent, a verb, and an object. Each member of the triplet may or may not have associated attribute(s) (e.g., adjective, adverb, etc.). Using the Machinese Syntax parser [Machinese Syntax], we first obtain XML-formatted syntactic and functional dependency information for each word of the input text, which constitutes the basis for generating the triplet(s). Since a new triplet is generated for each occurrence of a verb in the sentence, semantic parsing may extract more than one such triplet if multiple verbs are present in the sentence.

Basically, a triplet encodes information about “who is associated with what, where, and how” with a notion of semantic verb frame analysis. For example, the input sentence *“Eight members of a Canadian family vacationing in Lebanon were killed Sunday in an Israeli air raid that hit a Lebanese town on the border with Israel, Canadian and Lebanese officials said,”* produces three triplets as shown in Table 3.1.

Table 3.1 Triplet output of Semantic Parsing for the sentence given above.

Senses processed by SenseNet	
Triplet 1	[[[*'Subject-Name:'*, 'raid', *'Subject-Type:'*, 'concept', *'Subject-Attrib:'*, ['A ABS: Israeli', 'N NOM SG: air']], [*'Action-Name:'*, 'kill', *'Action Status:'*, 'Past Particle', *'Action-Attrib:',* ['passive', 'time: Sunday', 'place: Lebanon']], [*'Object-Name:'*, 'member', *'Object-Type:'*, 'person', *'Object-Attrib:'*, ['NUM: eight', 'A ABS: Canadian', 'N NOM: family', 'N NOM: vacationing']]]
Triplet 2	[[[*'Subject-Name:'*, 'raid', *'Subject-Type:'*, 'concept', *'Subject-Attrib:',* []], [*'Action-Name:',* 'hit', *'Action-Status:'*, 'Past ', *'Action-Attrib:'*, []], [*'Object-Name:'*, 'town', *'Object-Type:'*, 'N NOM', *'Object-Attrib:'*, ['A ABS: Lebanese', 'place: border', 'N NOM: Israel']]]
Triplet 3	[[[*'Subject-Name:'*, 'official', *'Subject-Type:'*, 'Object', *'Subject-Attrib:'*, ['A ABS: Canadian', 'A ABS: Lebanese']], [*'Action-Name:'*, 'say', *'Action-Status:'*, 'Past ', *'Action-Attrib:'*, []], [*'Object-Name:'*, '', *'Object-Type:'*, '', *'Object-Attrib:'*, []]]]

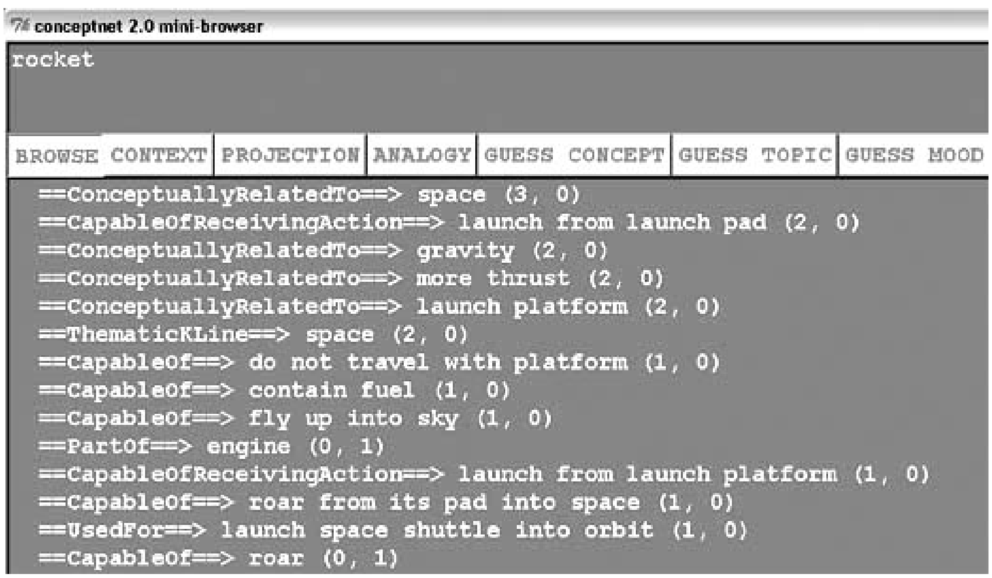
3.3 Developing Affective Lexica

WordNet: WordNet [Fellbaum 1999] is a database of English words organized into synonym sets, whereby each word is linked by a small set of semantic relations, such as the synonym relation or the 'is-a' hierarchical relation. The current version of WordNet (Version 2.1) contains 207,016 word-sense pairs and 78,695 polysemous senses. A sense, in the context of WordNet, is a distinct meaning that a word can assume. As a simple semantic network with words as the nodes, it can be readily applied to any textual input for query expansion, or determining semantic similarity. Thus, for an input word, we can obtain all the senses for a particular word which is usually not found in thesauri and dictionaries. The SenseNet is employing WordNet 2.1 for two purposes. The primary purpose is to assign a numerical value (i.e., prior valence), denoting either positive or negative valence, to each of our enlisted words (i.e., 'base list') obtained from English Vocabulary [English Club] based on manual investigation of senses of each word done by a group of judges (explained in sub-section Scoring a list of Verbs, Adjectives and Adverbs). The secondary purpose relates to situations where a word is not found in the 'base list'. Here, the system may automatically assign a valence for that word by first obtaining the synonyms of that word, and then screening the synonyms with respect to the 'base list' for which numerical values are already assigned. Then the new word and its valence are inserted into the 'base list'.

ConceptNet: ConceptNet [Liu and Singh 2004] is a semantic network of commonsense knowledge that currently contains about 1.6 million edges connecting more than 300,000 nodes. Nodes are semi-structured English fragments, interrelated by ontology of twenty semantic relations encompassing the spatial, physical, social, temporal, and psychological aspects of everyday life. ConceptNet is generated automatically from the 700,000 sentences of the Open Mind Common Sense (OMCS) Project, which were gathered from World Wide Web based collaboration with over 14,000 authors. A robust approach for weighting knowledge is implemented, which scores each binary assertion based on how many times it occurred in the OMCS corpus, and on how well it can be inferred indirectly from other facts in ConceptNet. One can consider ConceptNet as an extension of a model of purely lexical items with atomic meaning to higher-order compound concepts, which compose an action verb with one or two direct or indirect arguments. It also extends WordNet's list of semantic relations to a repertoire of twenty semantic relations including, for example, EffectOf (causality), SubeventOf (event hierarchy), CapableOf (agent's ability), PropertyOf, LocationOf, and MotivationOf (affect). Moreover, the knowledge in ConceptNet is of a more informal, defeasible, and practically valued nature. In the

SenseNet we have employed ConceptNet to retrieve all applicable semantic relationships of the input concept with other concepts. This is necessary to assign prior valence of a concept. (In the sub-section on Scoring of Nouns, we will explain how we process the output of ConceptNet.) By way of example, Figure 3.2 shows the semantic relationships obtained for the concept "rocket" with other concepts.

Figure 3.2 ConceptNet output for the concept 'rocket'



3.3.1 The Knowledgebase:

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 [Wilson et al. 2005]. For instance, the word 'destroy' usually bears a negative connotation, whereas 'develop' typically has a positive connotation. Cognitive and commonsense knowledge resources have been utilized to assign prior valence to the lexicon entries, and the resources also leverage scoring of new words, as will be explained in the following sub-sections.

The system maintains several lists of words having such prior valence. The “base list” is a list of verbs, adjectives, adverbs together with their prior valence, which is assigned based on WordNet. The “concept list” is a list of nouns, whose prior valence is calculated using ConceptNet. The “entity list” contains the named entities (e.g., kofi annan, ipod, etc.) for which ConceptNet fails to assign a prior valence. The prior valence of such named entities is assigned using an online resource named “Opinmind”. Since we are incorporating different resources to assign prior valence to the words, the question of ‘reliability’ of the assigned score might arise. We will briefly discuss this issue in the discussion section.

3.3.2 Scoring a list of Verbs, Adjectives and Adverbs

A group of eight judges has manually counted the number of positive and negative senses of each word of the initial “base list” of verbs, adjectives, and adverbs according to the contextual explanations of each sense found in WordNet 2.1. A judge’s score of a verb is stored as the following format:

verb-word: <positive-sense count, negative-sense count, prior valence, prospective value, praiseworthy value>

The prior valence, prospective and praiseworthy values indicate the lexical affinity of the verb with respect to “good” or “bad”, “desirable” or “undesirable”, and “praiseworthiness” or “blameworthiness”, respectively. Prospective and Praiseworthy values of the verb words are not used in SenseNet. We use those values in another system, where we aim to recognize more fine grained emotions like “happy”, “sad”, “relief”, etc.

We will explain the scoring procedure by an example. For the word ‘kill’, WordNet 2.1 outputs 15 senses as a verb and each of the senses is accompanied by at least an example sentence or explanation to clarify the contextual meaning of the verb. Each judge reads each meaning of the sense and decides whether it evokes positive or negative sentiment. E.g., for the word “kill”, one judge has considered 13 senses as negative and 2 senses as positive, which are stored in the scoring sheet. In this manner we initially collected the scores for 723 verbs, 205 phrasal verbs, 237 adjectives related to shape, time, sound, taste/touch, condition, appearance and 711 adjectives related to emotional affinity and 144 adverbs.

Eq. (1) assigns a prior valence (i.e., a value between -5 to 5) to each selected word.

$$pv(w) = \frac{\sum_{i=1}^{m}((\frac{p_i - n_i}{N}) * 5.0))}{m}. \tag{1}$$

Here, *pv(w)* = prior valence of word w, whereby -5 ≤ *pv(w)* ≤ +5
m = Number of judges (in this case, m=8)
p_i = The number of positive senses assigned by i-th judge, for word w
n_i = The number of negative senses assigned by i-th judge, for word w
N_i = Total number of senses counted by i-th judge for word w

A subset of verbs (e.g. like, love, hate, kiss etc.) of the "base list" is marked by a tag named <affect> to indicate that those verbs have affective connotation regarding preference or dislike. This tagging is done manually according to the semantic labels (i.e., a-labels) of WordNet-Affect [Valitutti et al. 2004]. To measure inter-agreement among judges, we used Fleiss' Kappa statistic [Fleiss 1971]. The Kappa value for the prior valence assignment task for the "base list" is reliable (κ=0.914). Moreover, our scoring resembles to the EVA function [Kamps and Marx 2002] score that assigns values to a word based on the minimal-path lengths from adjectives 'good' and 'bad'. A word not present in the annotated list is scored by calculating the average valence of its already scored synonyms obtained from WordNet. An excerpt from the verb database is given in Table 3.2.

Table 3.2 Sample list of verbs with associated Prior Valence.

verb word	prior valence
Amuse	3.750
Attack	-3.333
Battle	-5.000
Kill	-3.167
Thank	5.000
Wish	4.643
Yell	-1.250

3.3.3 Scoring of Nouns

Since manual scoring is a tedious job and the number of nouns is usually greater than the count of the words in “base list”, we employed ConceptNet to assign prior valence to nouns in an automatic manner. As described above, ConceptNet is a large semantic network of commonsense knowledge which encompasses the spatial, physical, social, temporal, and psychological aspects of everyday life. A value from [-5,+5] is assigned as the prior valence to an input noun or concept (we use “noun” and “concept” synonymously). If a concept is not present in our “concept list”, the system performs the following operations to assign prior valence to a concept. First, the system retrieves all other concepts which are semantically connected to the input concept using ConceptNet. For example, to assign valence to a concept *C*, the system collects all concepts Con_1, Con_2,…, Con_n, which are respectively connected to *C* with a specific semantic relationship like R_1,R_2, … R_m. ConceptNet defines twenty such relationship types between two concepts.

For the processing the returned concepts are separated into two lists depending on the type of semantic relationships. The entries in the first list correspond to relationships like ‘IsA', 'DefinedAs', 'MadeOf', 'PartOf', etc. and the entries in the second list correspond to relations like, 'CapableOf', 'UsedFor', 'CapableOfReceivingAction', ‘SubEventOf’, etc. Of the two groups, the first one indicates associated concepts which are basically nouns, and the second one indicates the actions (i.e., verb words) that the input concept can either perform or receive. The first list is matched against the "concept list", such that a maximum number of five concepts, which are found in the "concept list", are considered. The average of the prior valence values of the found concepts is assigned as the prior valence of the 'to be scored' concept. For faster processing we limit the number to five. If this procedure cannot assign a non-zero value, a similar procedure is performed considering the second list and the scored verbs of the "base list". The system considers the input concept as a named entity if the second procedure fails to assign a non-zero value as the prior valence of the 'to be scored' concept. If a non-zero valence is obtained, the input concept and its prior valence are inserted into the "concept list" for future use.

Let us look at an example. Initially, for the concept 'doctor', the system failed to find a prior valence in the existing scored list of nouns. Here, the following two lists are obtained by applying the explained procedures from the commonsense knowledgebase of ConceptNet.

related_concept_list = ['person', 'smart person', 'human', 'conscious being', 'man', 'wiley bandicoot', 'clever person', 'dentist', 'pediatrician', 'surgeon', 'physician', 'veterinarian', 'messy handwriting', 'study medicine', 'job']
related_action_list = ['examine', 'help', 'look', 'examine patient', 'help sick person', 'wear', 'prescribe medicine', 'treat', 'prescribe', 'wear white coat', 'look at chart', 'save life', 'heal person', 'take care'] (the list is truncated due to space limitations)

In this case the system first processed the 'related_concept_list', and failed to assign a non-zero value because initially the "concept list" did not have the score for those concepts in "related_concept_list". Therefore, the second list, 'related_action_list', is processed and from the second list the system returned the value 4.21 by averaging the scores of the verbs, 'examine (4.50)'; 'help (5.00)'; 'wear (2.57)'; 'prescribe (4.27)' and 'treat (4.69)'. Hence the value 4.21 is assigned as the prior valence for the concept 'doctor' and stored for future use. Instead of performing manual scoring of verbs, adjectives and adverbs we initially scored about 4500 concepts using the procedure explained above. The “concept list” is maintained to speed up the processing time since otherwise the system would have to invoke ConceptNet and perform scoring every time for a concept (i.e., noun word).

3.3.4 Scored-list of Named Entity

The system maintains a list named “entity list” that contains prior valence of named entities. We did not use any named entity recognizer to identify a named entity, and hence make the simplifying assumption that anything for which ConceptNet fails to assign a non-zero value is a named entity. The information of an entity is stored in the following format:

named entity: <concept, concept valence, general-sentiment, prior valence>

The attribute ‘concept’ indicates a noun which describes the named entity in terms of “is a kind of” or “conceptually related to” type relationships. Attribute ‘concept valence’ indicates the prior valence of the concept. E.g., for the sentence, “*President George Bush spoke about the ‘Global War on Terror.’”*, the system signals “George Bush” as a named entity because it failed to assign a non-zero valence using ConceptNet. However, based on the output of the Semantic Parser, the system finds the noun “president” as an attribute associated with this named entity. Hence for this named entity the system considers “president” as the ‘concept’ attribute and from the ConceptNet system gets the prior valence for “president” as +2.75. If the system fails to receive any such noun attribute associated with the named entity, the system assumes an abstract concept named “person” assuming to have a “conceptually related to” type relationship. In this manner we attempt to extend the scope of ConceptNet by incorporating real-world knowledge.

The attribute ‘general-sentiment’ contains either a negative (-1) or a positive (+1) value based on the value of the prior valence towards the named entity. To assign ‘general-sentiment’ as well as prior valence we have developed a tool that can extract sentiment from Opinmind [Opinmind

2006]. Opinmind is a web-search engine which has a sentiment scale named 'Sentimeter' that displays the relative number of positive and negative opinions expressed by people on anything regarding one's views on politics and current events. It also finds what people think about products, brands, and services by mining the opinion bearing texts of people's blogs. Opinmind exercises no editorial judgment when computing 'Sentimeter' values. For example, ConceptNet fails to assign a valence to "George Bush" or "Tokyo University". From Opinmind we obtain 37% positive, and 63% negative opinion regarding the named entity "George Bush". Similarly for the input "Tokyo University" we obtain 100% positive, 0% negative opinion. From the obtained values we set the 'general-sentiment' and 'prior valence' as -1 and -3.15 (considering the maximum of the absolute value of the votes in the scale of 5) for "George Bush" and similarly for "Tokyo University" the values are set to +1 and +4.1. Hence these are stored as: George Bush:<President, +2.752, -1, -3.15>; Tokyo University:<school, 4.583, +1, +5.0>. Initially a list of 2300 entries is manually created and scored using Opinmind. This list grows automatically whenever the system detects a new named entity. Usually the value of 'general-sentiment' is idiosyncratic and arguable. If the valence sign of the 'concept valence' and 'general-sentiment' (e.g., President [+2.752], George Bush [-1]) differs from each other, the system considers this as an ambiguity and assigns neutral valence to the sentence referring that named entity. An excerpt from database is given in Table 3.3 to illustrate the idea.

Table 3.3 Sample list of scored named entities

Named Entity	Concept	Concept Valence	General-Sentiment	Prior Valence
Bin Laden	War	-4.625	-1	-3.10
George Bush	President	2.752	-1	-3.15
Discovery	Shuttle	3.984	+1	+4.25
Kofi Annan	Person	2.562	-1	-4.50
Microsoft	Software	4.583	-1	-2.65
NASA	Space	3.784	+1	+3.80

3.4 Contextual Valence Assessment

Before explaining the contextual valence assignment algorithm, we first discuss its underlying data structure.

Input. The minimal input to the system is a sentence *S*. A paragraph *P*, containing one or more sentences can also be processed by the system.

Processing elements. We assume the input is a paragraph *P*, containing *n* sentences, such that $P = <S_1, S_2, \ldots, S_i, \ldots, S_n>$ and $1 \leq i \leq n$. As a sentence S_i may have one or more verbs, the semantic parser may output one or more triplet(s) for S_i. We represent S_i as a set of *m* triplets *T*, i.e., $S_i = <T_1, T_2, \ldots T_j, \ldots, T_m>$, whereby $1 \leq j \leq m$. A triplet T_j has the following form: ⟨actor, action, concept⟩. The triplet elements "actor" and "concept" have the following form, ⟨name, type, attribute⟩. The action has the form ⟨name, status, attribute⟩. An attribute is either an empty set or non-empty set of words. For example, the input *S*, *'The President called the space shuttle Discovery on Tuesday to wish the astronauts well, congratulate them on their space walks and invite them to the White House.'*, the following four triplets are obtained for the four verbs.

T_1 = ⟨⟨President, Concept, {the}⟩, ⟨call, past, {time: Tuesday, dependency: to}⟩, ⟨discovery, Named Entity, {the, space, shuttle}⟩⟩

T_2 = ⟨⟨President, Concept, {the}⟩, ⟨wish, infinitive, {dependency: and}⟩, ⟨astronaut, Concept, {the, adv: well}⟩⟩

T_3 = ⟨⟨President, Concept, {the}⟩, ⟨congratulate, infinitive, {dependency: and}⟩, ⟨astronaut, Concept, {goal: space walk}⟩⟩

T_4 = ⟨⟨President, Concept, {the}⟩, ⟨invite, infinitive⟩, ⟨astronaut, Concept, {place: white house}⟩⟩

Knowledgebase. The knowledgebase of the system has been discussed above. Using that data source, the system builds the following computational data structure that is consulted to process the input text. 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 (AV_{pos}, V_{pos}) and negative (AV_{neg}, V_{neg}) groups on the basis of their prior valence. Similarly, adjectives (ADJ), adverbs (ADV), concepts (CON) also have positive and negative groups indicated by ADJ_{pos}, ADJ_{neg}, ADV_{pos}, ADV_{neg}, CON_{pos}, and CON_{neg}, respectively. For a named entity (NE) the system creates three kinds of lists, namely ambiguous named entity (NE_{ambi}), positive named entity (NE_{pos}) and negative named entity (NE_{neg}). The named entity that has a different sign for the valence of 'genre' and 'general sentiment' fields is a member of NE_{ambi}.

Algorithm. The core algorithm underlying our system can be summarized as follows. Input, *P,* is a paragraph which is a sequence of sentences. Output of the system is *V* that indicates valence values for each corresponding sentence. For each sentence the following steps are performed. The pseudo-code of the algorithm for contextual valence assignment (i.e., function *getValence()*) is given in Appendix A.

First, the triplet representation (i.e., a set of triplets) of the sentence is obtained from the Semantic Parser. A triplet is basically consisting of a subject, verb and object where each of them might have associated attributes like adverb, adjective or nominative noun. To indicate dependency relationship between two adjacent triplets the parser outputs a dependency tag like, "dependency: to", "dependency: and", "dependency: but", "dependency: nonetheless", "dependency: as" etc., associated with a triplet depending on the presence of connectives or conjunctions in the input sentence. At present, for simplicity the dependency relationships are grouped into two types namely, "to_dependency" (i.e., "dependency: to") and "not_to_dependency" (i.e., all others except "dependency: to").

Second, all the triplets obtained from the input sentence are processed to assign a valence value to the sentence. This procedure involves the following steps. (1) Rules are applied to assign contextual valence to the subject, verb and object of the triplet considering their attributes (i.e., adverb, adjective). (2) Conditionality, negation, and previously assigned contextual valence values are considered to assign a contextual valence to the triplets. Thus each triplet is assigned a contextual valence. (3) The dependency relationships (if any) among the adjacent triplets are considered and resultant valence values are assigned according to the "dependency processing" algorithm mentioned in the sub-section on Sentiment Assessment. For the two types of dependencies different sets of rules are applied to calculate resultant valence for two interdependent triplets. Finally a valence is calculated for the input sentence from those resultant valence values. In this procedure valence values are assigned to all the sentences of the input paragraph.

Here are some example rules to compute contextual valence using attributes (e.g., adjectives and adverbs).

- ADJ_{pos}+(CON_{neg} or NE_{neg})→neg. Valence (e.g., strong cyclone; nuclear weapon)
- ADJ_{pos}+(CON_{pos} or NE_{pos})→pos. Valence (e.g., brand new car; final exam)
- ADJ_{neg}+(CON_{pos} or NE_{pos})→neg. Valence (e.g., broken computer; terrorist group)
- ADJ_{neg}+(CON_{neg} or NE_{neg})→neg. Valence (e.g., ugly witch; scary night)

Note that the sign of the valence switches because of the adjectives when there is a negative scored adjective qualifying a CON_{pos} or NE_{pos}. In other cases the sign of respective CON or NE is unchanged. The resultant valence (i.e., actor valence or object valence) is also intensified than the input CON or NE due to ADJ.

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.

- ADV_{pos} + (AV_{pos} or V_{pos})→ pos. Valence (e.g., write nicely; sleep well)
- ADV_{pos} + (AV_{neg} or V_{neg})→ neg. Valence (e.g., often miss; always fail)
- ADV_{neg} + (AV_{pos} or V_{pos})→ neg. Valence (e.g., rarely complete; hardly make)
- ADV_{neg} + AV_{pos} → pos. Valence (e.g., badly like; love blindly)
- ADV_{neg} + (AV_{neg} or V_{neg})→ ambiguous (e.g., hardly miss; kill brutally)

Hence, the rules to resolve the ambiguity are:

- ADV_{neg}-except+(AV_{neg} or V_{neg})→pos. Valence (e.g., rarely forget; hardly hate)
- ADV_{neg}-not except+(AV_{neg} or V_{neg})→neg. Valence (e.g., suffer badly; be painful)

The contextual valence of Action-Object pairs is computed based on the following rules taking the contextual valence of action and object into consideration.

- Neg. Action Valence + Pos. Object Valence → Neg. Action-Object Pair Valence (e.g., kill 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., buy 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., buy a gun, patronize a famous terrorist gang, make nuclear weapons, etc.)

We are aware that 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 (AV_{pos}, AV_{neg}) that uses the following rules to assign contextual valence for an affective verb.

- AV_{pos} + (pos. or neg. Object Valence) = pos. Action-Object Pair Valence (e.g., I like romantic movies. She likes horror movies.)
- AV_{neg} + (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. 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. The rules are:

- (CON_{pos} or NE_{pos})+ Pos. Action-Object Pair Valence→ Pos. Triplet Valence (e.g., the professor explained the idea to his students.)
- (CON_{pos} or NE_{pos}) + Neg. Action-Object Pair Valence→ Neg. Triplet Valence (e.g., John rarely attends the morning lectures.)
- (CON_{neg} or NE_{neg}) + Pos. Action-Object Pair Valence→ Tagged Negative Triplet Valence (e.g., the robber appeared in the broad day light.) to process further.
- (CON_{neg} or NE_{neg}) + Neg. Action-Object Pair Valence→ Neg. Triplet Valence (e.g., the strong cyclone toppled the whole city.)

For example, the input sentence *"The robber arrived with a car and mugged the store-keeper."* outputs two triplets with a 'dependency: and' attribute in the first triplet indicating that the first triplet has an '*and* relationship' with the second one. Of the two triplets the first one is assigned to 'tagged negative triplet valence' for the negative valence actor *'robber'* with a positive 'action-object pair valence' for *'arrive, car'*. The other triplet is assigned with a 'negative triplet valence' for having actor (*'robber'*) and 'action-object pair valence' for *'mug, store-keeper'* as negative. So in this case we notice that a negative valence actor is associated with a positive and negative 'action-object pair'. For such cases our simplified heuristic is that if a negative valenced actor is associated with at least one 'negative action-object pair', the tagged output is considered as negative and the resultant valence is made negative. But if a negative valenced actor is associated with all positively scored 'action-object pair' the 'tagged negative triplet valence' is set to positive and the resultant valence is made positive. For example, *"The kidnapper freed the hostages and retuned the money."* gives two tagged negatives scores (i.e.; -8.583 and -9.469) for two positive "action-object pair valence" (i.e., 'free, hostage' and 'return, money'). Hence, the system finally assigns a positive valence because the negative valenced actor is not associated with any negative 'action-object pair'. This implies that an action done by a negative-role actor is not necessarily always negative. We also consider the cases of negation and conditionality like [Hu and Liu 2004][Wilson et al. 2005].

3.5 Sentiment Assessment

In the previous sub-section we described how valence is assigned to triplets. Now we explain how sentiment (i.e., assessing contextual valence of the triplets) is assessed for a sentence. It is previously mentioned that from the Semantic Parser two types of dependencies are tagged to indicate the dependency between two triplets. The system invokes a function (*processTripletLevelContextualValence()*) to process the dependencies among the triplets and set the contextual valence of those triplets. The algorithm of this function is described below:

For the two triplets, T_1 and T_2 where T_1 has a "to_dependency" relationship with T_2, the contextual valence of the triplets are calculated according to the following rules,

- Contextual Valence Value = (abs(valence of T_1) + abs(valence of T_2) / 2
- Pos. valence of T_1 + Pos. valence of T_2 → Pos. Contextual Valence (e.g., I am interested to go for a movie.)
- Neg. valence of T_1 + Pos. valence of T_2 → Neg. Contextual Valence (e.g., It was really hard to swim across this lake.)
- Pos. valence of T_1 + Neg. valence of T_2 → Neg. Contextual Valence (e.g., It is easy to catch a cold at this weather.)
- Neg. valence of T_1 + Neg. valence of T_2 → Pos. Contextual Valence (e.g., It is difficult to take bad photo with this camera.)

Similarly, the rules to deal with "not_to_dependency" relationship are:

- Contextual Valence Value = (abs(valence of T_1)+(valence of T_2)))/2
- Pos. valence of T_1 + Pos. valence of T_2 → Pos. Contextual Valence (e.g., they got married and lived happily.)
- Neg. valence of T_1 + Pos. valence of T_2 → Pos. Contextual Valence (e.g., John was not a regular student but he finally scored good grades.)
- Pos. valence of T_1 + Neg. valence of T_2 → Neg. Contextual Valence (e.g., the movie was very interesting but at the end it became monotonous.)
- Neg. valence of T_1 + Neg. valence of T_2 → Neg. Contextual Valence (e.g., I feel very sad when my paper gets rejected.)

The pseudo-code of the function *processTripletLevelContextualValence()* is given in Appendix A. This function returns a list namely 'contextualValence' which contains valence values of the triplets after processing their dependencies. The average of the absolute values of the list 'contextualValence' is assigned as the 'sentimentScore' for the sentence, S. The 'valenceSign' is set +1 if the count of positive values in the list is greater than the number of negative ones and vice versa. If both negative and positive counts are equal then +1 is set if the sign of the maximum value considering the absolute values of the list is positive, otherwise -1 is set. The value of 'sentimentScore' is multiplied with 'valenceSign' to get 'sentenceValence' and this is the valence the system finally for the input sentence. According to the scoring system the range of 'sentenceValence' is ±15 since the maximum and minimum valence of a triplet can be 15 and -15 respectively.

The above idea is further explained by an example of how contextual valence values are assigned to the triplets of the input sentence, *"Tropical storm Bilis killed at least 48 people and injured hundreds as it churned across China's south-east, toppling houses and forcing authorities to evacuate a prison and thousands of villagers."*

SenseNet detected the following seven triplets for the input sentence
Triplet 1: ['Bilis {tropical, storm}' , 'kill {dependency: and}', 'people {at least, 48}'],
Triplet 2: ['Bilis', 'injure {dependency: as}', 'people, {hundreds}'],
Triplet 3: ['Bilis', 'churn across {dependency: and}', 'china {south-east}'],
Triplet 4: ['Bilis', 'topple {dependency: and}', 'house'],
Triplet 5: ['Bilis', 'force {dependency: to}', 'authority'],
Triplet 6: ['authority', 'evacuate {dependency: and}', 'prison'],
Triplet 7: ['authority', 'evacuate', 'villagers']

All the attributes of the triplets are not shown due to space limitations. In the first triplet the subject "Bilis" is a named entity which will be evaluated as the concept "storm" because it appears as a noun attribute of the subject (i.e., "Bilis"). In the subsequent triplets the pronoun "it" as the subject has been replaced by the previously found subject 'Bilis'. Due to the presence of a noun (i.e., 'authority') as the object in the fifth triplet and the presence of a verb (i.e., 'evacuate') with a "dependency: to" relationship without having a direct subject, Semantic Parser considers 'authority' as the subject for the sixth and seventh triplet. The sixth and seventh triplet have the same verb connecting two objects with an 'and' relationship.

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

“storm”: -3.394; “tropical”: 2.861; “kill”: -3.937; “people”: 2.5; “injure”: -3.634; “churn across”: -3.696; “china”: 3.450; “south-east”:0.0; “topple”:-3.324; “house”:5.0; “force”: 2.985; “authority”:3.196; “evacuate”: -2.694; “prison”:0.588; “villager”:3.812.

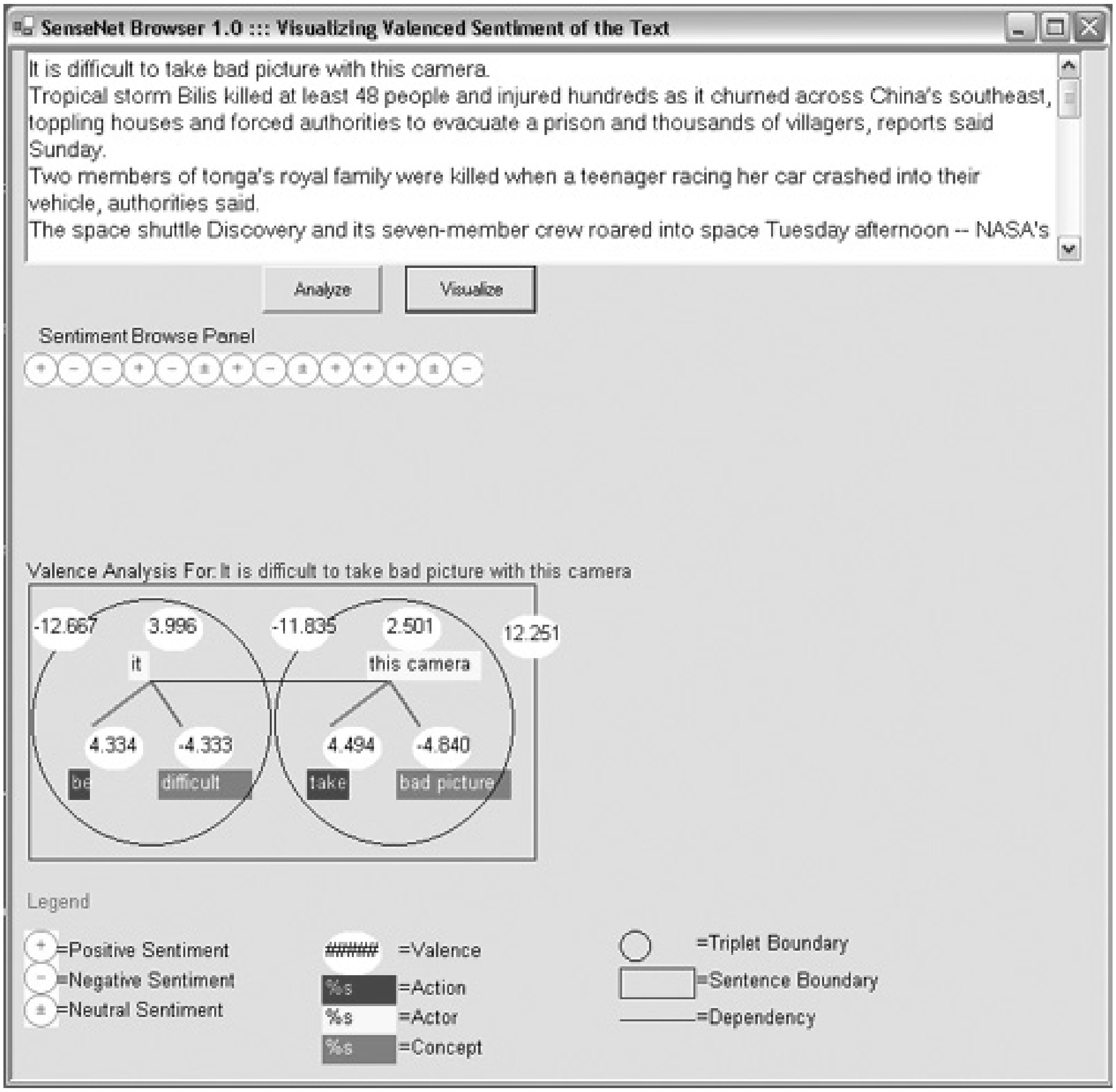
According to the algorithm (*getValence()* in Appendix A) the system prepares the list of triplets along with the dependency relationships (i.e., “tripletResult”) as following: {(-10.650, true, “dependency: and”), (-10.343, true, “dependency: as”), (-11.359, true, “dependency: and”), (-12.537, true, “dependency: and”), (-10.394, true, “dependency: to”), (-6.478, true, “dependency: and”), (-9.702, false, *null*)}. The numerical values shown in the list indicates the valence of the corresponding triplets (i.e., “tripletValence”). The dependencies among the triplets and the valence of the triplets (i.e., “tripletResult”) are processed (by the function *processTripletLevelContextualValence()*) to set the contextual valence of those triplets. According to the aforementioned algorithm of this function the following list (i.e., “ContextualValence”) of values is obtained: [-10.496, -10.851, -11.948, -11.465, +9.242]. The fifth value of the list is positive because of the rule of having two negative triplets connected with “to_dependency” relationship. On processing this list of values the ‘valenceSign’ is set negative because most of the values are negative and the ‘sentimentScore’ is obtained as 10.80 for this sentence. Finally the ‘sentenceValence’ is outputted as -10.80 indicating that the sentence bears a negative sentiment. Similarly for the sentence *“It is difficult to take bad photo with this camera”*, the “sentenceValence” is obtained as +12.251 indicating the sentence expressing a positive sentiment.

3.6 SenseNet GUI

The SenseNet browser graphically visualizes each sentence in terms of the triplets and their associated valence values. SenseNet Browser is the front-end user interface for SenseNet and it is written in C#. It takes the input from the users and sends it to the backend python implemented program for analysis through TCP/IP socket connection. As shown in Fig. 3.3, the browser has two panels for user interaction, namely "Input panel" and "Sentiment Browse panel". In "Input panel" a chunk of text can be inputted and clicking on "Analyze" button sends the text to the backend python application to process it and finally receives the output. By clicking on the "Visualize" button an ordered iconic representation of underlying sentiment of each input sentence(s) is displayed on the "Sentiment Browse panel" corresponding to the order of appearance of the sentences. The browser also has two other panels, namely "Valence Analysis panel" and "Legend panel". A click on any of the icons of "Sentiment Browse panel" is considered as the user's request to show the analysis of the sentiment for that particular sentence represented by that icon. "Valence Analysis" panel then shows the triplets and the valences associated with those. The "Legend panel" explains the different icons and symbols used by the browser. SenseNet classifies sentences into three classes namely, Negative, Positive, and Neutral. According to performed experiment (mentioned in SenseNet Evaluation) it is decided that for a sentence which valence is between the ranges of ±3.5 is decided as a neutral sentence.

Figure 3.3 shows an example of output obtained by SenseNet. The interface indicates that it has processed 14 sentences of which there are six positive, five negative, and three neutral sentiment carrying sentences. This is represented by the line of the circles with embedded polarity signs. Clicking on the first circle the valence analysis for that sentence is shown. This type of browser will be helpful to readily identify and visualize the positive, negative or neutral sentiment bearing sentences from the textual data like product reviews or users' comments, blogs posts, email contents etc. Moreover the iconic representation of underlying sentiment of the input text will help a user to grasp the sentimental perception (i.e., negative, positive, or neutral) of the input text in an easy manner. The idea of this browser might be extended to multi-document level (e.g., a set of emails etc.) where the iconic representation of "Sentiment Browse panel" would be produced based on the overall sentiments of the input documents. Such information visualization will help to filter contents quickly and easily.

Figure 3.3 Interface and Sample Output of SenseNet Browser



SenseNet browser uses several symbols to represent the visualization and analysis of sentiment of texts. Table 3.4 explains the symbols used in SenseNet.

Table 3.4 Symbols used in SenseNet browser.

Symbol	Explanation
+	Indicates positive sentiment of a sentence.
−	Indicates negative sentiment of a sentence.
±	Indicates neutral sentiment of a sentence.
######	White circle with a numerical value inside indicates the contextual valence of a linguists components (e.g., noun, verb, sense, sentence)
%s	Text in black color filled rectangle indicates an action of a triplet.
%s	Text in white color filled rectangle indicates an actor of a triplet.
%s	Text in gray color filled rectangle indicates an object of a triplet.
	Black bordered circle represents a triplet.
	Black bordered rectangle represents the boundary of a sentence containing all the trip
	Black line connecting two actors indicates the interdependency between two triplets. Black lines connection action and concept/object with actor indicates the connectivity within a triplet.

Chapter Four: SenseNet Evaluation

We intend to evaluate our system both at the sentence level and paragraph (or document) level. To this end, we performed system evaluation in two ways, first, by comparison with a "gold standard", and second, by comparison to another state-of-the-art system [Liu et al. 2003].

4.1 The Datasets

We use four datasets to test our method of sentiment assessment for both sentence and paragraph (or document) level. The evaluation to assess the accuracy of sentence level sentiment recognition is performed by comparing system results to human-ranked scores (as "gold standard") for two datasets.

The first one, Dataset A, is created by collecting 200 sentences from internet based sources for reviews of products, movies, and news [My Yahoo!], and email correspondences. It was scored by 20 human judges according to positive, negative, and neutral sentiment affinity by an online survey[1]. The judges were instructed to login to the online survey system to read the sentences and score each sentence in terms of 'Sentiment' (i.e., negative, positive or neutral) and "Intensity" (i.e., low, mid, high, extreme) of sentiment by selecting radio buttons. After the survey the number of positive, negative, and neutral sentences has been decided according to the scores for which maximum number of judges are found unanimous for each sentence. For example, the input sentence *"She is extremely generous, but not very tolerant with people who don't agree with her."*, was rated as negative by 14 judges (out of 20), as neutral by five judges, and as positive by one judge. Since the majority of the judges voted this sentence as a negative sentence, the sentence is considered as a negative sentence in our "gold standard" dataset. The inter-rater agreement was calculated using Fleiss' Kappa statistics. The Kappa coefficient (κ) for sentence

[1] http://ita.co.jp/research/survey/ (one can login using a guest username)

scoring is 0.782, showing good reliability of inter-rater agreement. This dataset contains 90 positive, 87 negative, and 23 neutral sentences. Detail about the dataset is given in Table 4.1.

The second dataset, Dataset B, is the sentence polarity dataset v1.0[2] introduced in [Pang and Lee 2005). The dataset contains 5331 positive and 5331 negative classified sentences or snippets (i.e., only the subjective opinion sentences of movie reviews). The primary motivation of using these two datasets is that they contain individual sentences classified as positive, negative or neutral (for Dataset A), or positive or negative (Dataset B), which is accord with the purpose of our first experiment, namely, to answer how efficiently the system can assess sentiments at sentence level.

The evaluation to assess the accuracy of paragraph (or document) level sentiment recognition is performed using Dataset C and Dataset D. We consider a paragraph (or document) as a set of sentences and the sentiment for a paragraph (or document) is currently assessed by considering the average score obtained from the scores of the sentences of the pertaining paragraph (or document). Dataset C is the polarity dataset V2.0 introduced in Pang and Lee (2004), which consists of 1000 positive and 1000 negative review documents. This dataset has become the *de facto* standard dataset for sentiment-classification and has been used in over 15 research papers. Since movie reviews are known to be difficult to classify [Turney 2002][Turney and Littman 2003], we are motivated to test the performance of our system with such data.

Dataset D is a set of 100 reviews taken from www.epinions.com. This dataset contains 50 positive and 50 negative reviews. The reviews were collected from a variety of product reviews, including reviews on computer, mp3 player, mobile phone, car, vacuum cleaners, TVs, and washing machines. Reviews at epinions.com are rated with a 5 star system where 1 is the lowest and 5 is the highest score. Reviews where the product gets 1 or 2 stars are considered to be negative, reviews with 4 or 5 stars are considered to be positive. The purpose of using Dataset D

---

[2]Introduced in Pang and Lee at ACL 2005. at http://www.cs.cornell.edu/People/pabo/movie-review-data/

is to measure the accuracy of the system in assessing the sentiment from product reviews. Both of the datasets (i.e., Datasets C and D) contain more than 4 sentences in each review. A summary of our "gold standard" datasets is given in Table 4.1.

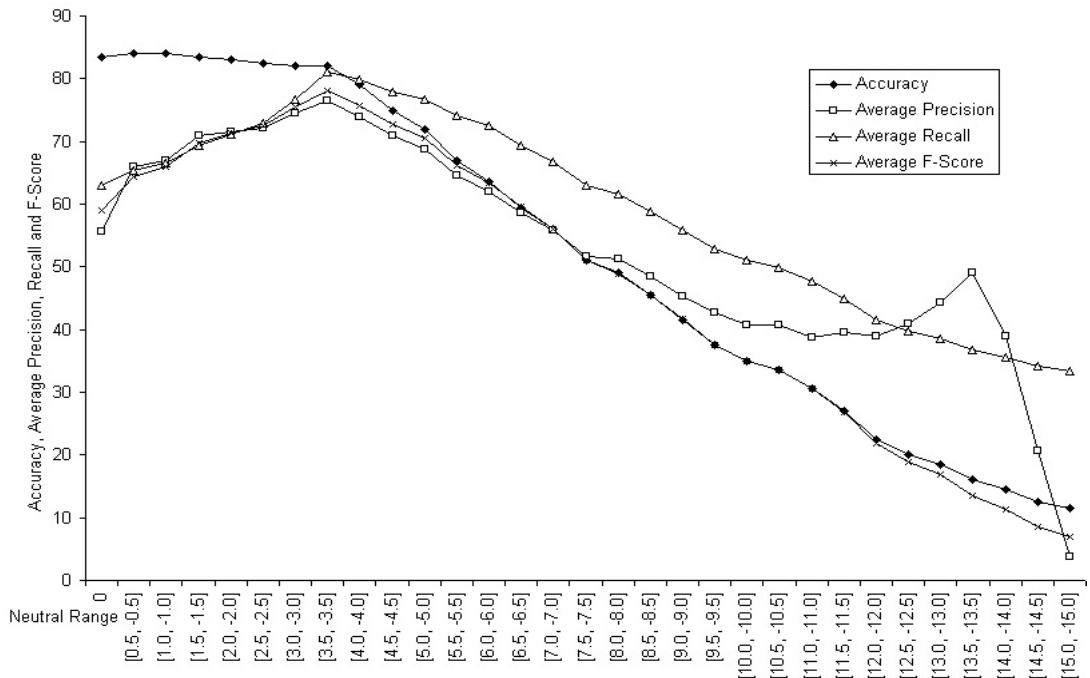
Table 4.1 Input datasets

Dataset	Data Type	Data Attributes	Data Source
Dataset A	Sentence	Data collected from various domains. 90 Positive, 87 Negative and 23 Neutral sentences. More specifically the contexts and sentences are: Email: 6 pos, 5 neg, & 2 neu Product Review: 21 pos, 21 neg, 6 neu Movie Review: 15 pos, 16 neg, 5 neu News: 48 pos, 45 neg, 10 neu	Authors managed to collect the data and scoring is done by an online survey.
Dataset B	Sentence	Collected from Movie Review (Rotten Tomatoes pages). There are two files. One contains 5331 positive snippets and other has 5331 negative snippets. Each line in these two files corresponds to a single snippet (usually containing roughly one single sentence); all snippets are down-cased.	Sentence polarity dataset v1.0. Introduced in Pang and Lee at ACL 2005. Can be found in at this source(http://www.cs.cornell.edu/People pabo/movie-review-data/)
Dataset C	Paragraph	Movie Review 1000 positive and 1000 negative processed reviews.	polarity dataset v2.0 Introduced in Pang and Lee at ACL 2004. can be found at this source(http://www.cs.cornell.edu/People pabo/movie-review-data/)
Dataset D	Paragraph	Product Review 50 positive, 50 negative reviews about different products, including computer, mp3 player, mobile phone, car, vacuum cleaners, TVs, and washing machines taken from epinions.com.	The authors collected this data from the website www.epinions.com

4.2 Sentence Level Comparisons

Comparing to Gold Standard: In our first experiment, since Dataset A has neutral sentences, the system performs as a three-class (i.e., positive, negative and neutral) classifier. Hence, we set different valence ranges to signal the neutrality of sentiment. The motivation is to identify the valence range for which the system shows the highest F-score in terms of classifying negative, positive or neutral sentiment bearing sentences with respect to the gold standard. The experimental result details are given in Appendix B. According to the result, increasing the neutral range increases the recall of neutral sentences, but decreases recall for positive and negative sentence classes. We noticed that after a certain range (here, -6 to 6), the recall for the neutral sentence class is maximized (100%), and the recall for other two classes becomes lower than 80% for the range -4.5 to 4.5. We also calculated the average of recall, precision, and F-score of the three classes for each neutral range and plotted it in line graphs, as shown in Figure 4.1. According to Figure 4.1, the system achieves the highest accuracy (84%) for the ranges ±0.5 and ±1.0, but it shows the highest average recall (81.04%), precision (76.49%) and F-score (78%) for the neutral range ±3.5. Since the highest F-score is achieved at this point, we decided this valence range to classify a sentence as 'neutral', i.e. the 'sentenceValence' score resides within this range.

Figure 4.1 Relationship between the 'Neutral Range' of the system to signal neutrality of a sentence and other system performance measures namely, Accuracy, Average Precision, Recall, and F-Score for three classes.



Comparing to Gold Standard and SVM based approaches: Dataset B has only two types of sentences, either positive or negative. Hence, in this experiment our system acts as a two-class (i.e., positive, negative) classifier. We compared the performance of our system with several other methods. Table 4.2 summarizes the accuracy of different approaches including ours for this dataset.

Table 4.2 Accuracy results obtained for Dataset B using different approaches.

Approaches	Accuracy
Unigram SVM	75.11%
Bi-gram SVM	71.04%
Linguistic Tree Transform SVM	84.09%
Our Approach	91.53%

From this database, the first 4000 sentences were used to form a training set, and the remaining 1331 sentences were used to test accuracy performance using SVM approaches according to the experiments regarding SVM described in [Pang and Lee 2005] and [Eriksson 2006]. The lists output from the Linguistic Tree Transformation Algorithm were arranged into frequency SVM model form (with the SVM-light software package). Performance was tested against a frequency unigram SVM model and a frequency bi-gram SVM mode. In our experiment, 10,662 sentences were input to the system and obtained a recall of 90.62% and 92.44%, with precision of 92.07% and 91.01%, for classifying positive and negative sentences, respectively.

4.3 Paragraph Level Comparisons

Comparing to Machine Learning Approaches: Dataset C has been tested by comparing various approaches, including approaches based on machine learning algorithms. While we built our system mainly to assess sentence level sentiment, we carried out this experiment in order to investigate the performance of the system when processing chunks of sentences (i.e., a paragraph or document). The method to obtain a score for text chunks is straightforward. We obtain its score by averaging over the scores of individual (positively and negatively scored) sentences. For example, following except is taken from one of the positive movie reviews found in Dataset C (only 3 subjective sentences are given for space limitation).

"if you want some hearty laughs , then rat race is the movie for you . this unpretentious little comedy , which sneaks into theaters today with very little hype , will have you bouncing in your theater seat. and while the film fits neatly into the low-brow , slapstick school of comedy , one refreshing aspect is its lack of mean-spiritedness."

The system obtained +4.90 as the score for the above paragraph whereby the three sentences received the scores 11.11, -7.72, and 11.32 respectively. Performance results for both machine learning and non-machine learning based approaches are reported in Table 4.3.

Table 4.3 Summary of several systems that experimented with Dataset C

Machine Learning Systems		
Experimental Result reported in [Vincent et al. 2006]	**Accuracy**	
Adding bi-grams and tri-grams	89.2%	
Adding dependency relations	89.0%	
Adding polarity info of adjectives	90.4%	
Discarding objective materials	90.5%	
Experimental results reported in [Mullen and Collier 2004]	**Accuracy**	
	3 folds	10 folds
[Pang et al. 2002]	82.9%	NA
Turney Values only	68.4%	68.3%
Osgood only	56.2%	56.4%
Turney Values and Osgood	69.0%	68.7%
Unigrams	82.8%	83.5%
Unigrams and Osgood	82.8%	83.5%
Unigrams and Turney	83.2%	85.1%
Unigrams, Turney, Osgood	82.8%	85.1%
Lemmas	84.1%	85.7%
Lemmas and Osgood	83.1 %	84.7%
Lemmas and Turney	84.2%	84.9%
Lemmas, Turney, Osgood	83.8%	84.5%
Hybrid SVM (Turney and Lemmas)	84.4%	86.0%

Hybrid SVM (Turney/Osgood and Lemmas)	84.6%		86.0%		
Non-Machine Learning Systems					
Experimental result reported in [Kennedy and Inkpen 2006]	Accuracy (A),Precision (P),Recall (R) for Positive and Negative Classes				
Basic: GI	A=59.5%;P= 57.8%,69.8%;R=82.8%,36.1%;				
Basic: GI & CTRW & Adj	A=65%; P=64.5%,69.6%; R=73.3%, 56.6%;				
Basic: GI & SO-PMI 1	A=57.7%; P=87.9%,54.6%; R=18.8%, 96.6%				
Basic: GI & SO-PMI 2	A=63.2%; P=61.1%,73.5%; R=82.5%,43.8%				
Improved: GI	A=62.7%; P=59.8%,71.1%; R=81.7%, 43.6%				
Improved: GI & CTRW & Adj	A=66.7%; P=65.8%,70%; R=73.4%,60.1%				
Improved: GI & SO-PMI 1	A=58.4%; P=87.3%,55.1%; R=20%,96.8%				
Improved: GI & SO-PMI 2	A=65.1%; P=61.9%,73.9%; R=81.6%,48.6%				
Our Approach [Non-Machine Learning System]					
	Class	Accuracy	Precision	Recall	F-Score
Our System (i.e., SenseNet)	Positive	85.5%	87.78%	79.7%	83.54%
	Negative		83.60%	91.3%	87.28%

As already pointed out by [Turney 2002], we notice that the movie review data contains a large portion of ‘objective’ data (i.e., the text that describes about the plot of the movie), which cause noise in the analysis. As a review can contain both subjective and objective phrases, review identification can be viewed as an instance of the broader task of identifying which sentences in a document are factual or objective, and which are opinionated or subjective. There have been attempts on tackling this so-called document-level subjectivity classification task, with very encouraging results (see [Yu and Hatzivassiloglou 2003] and [Wiebe et al. 2004] for details). Our system outperformed the non-machine learning approaches, and achieved almost the same result as Hybrid SVM (Turney/Osgood and Lemmas) approach. The approach “Discarding objective materials” achieved the best performance using this dataset. However, in that experiment, first, the objective sentences are detected from the input review, and then classification is done based on the auto-detected subjective sentences of the review. In our opinion, if the objective sentences could be omitted, the performance of our system would increase but at present we have not considered pre-processing in order to filter the objective sentences.

Comparing to Online Rating as Gold Standard: Dataset D is a set of 100 reviews taken from the web-site: www.epinions.com. Table 4.4 summarizes the experimental result using this dataset.

Table 4.4 Experimental result using the Dataset C

Review Data Genre	Class/Sample Size	Accuracy	Precision	Recall	F-Score
Computer	Positive/8	78.57%	85.71%	75.00%	80.00%
	Negative/6		71.43%	83.33%	76.92%
mp3 Player	Positive/6	72.73%	66.67%	66.67%	66.67%
	Negative/5		80.00%	80.00%	80.00%
Mobile phone	Positive/10	85.00%	80.00%	80.00%	80.00%
	Negative/10		90.00%	90.00%	90.00%
Automobile	Positive/8	83.33%	77.78%	87.50%	82.35%
	Negative/10		88.89%	80.00%	84.21%
vacuum cleaner	Positive/4	87.50%	100.00%	75.00%	85.71%
	Negative/4		80.00%	100.00%	88.89%
TV	Positive/9	83.33%	80.00%	88.89%	84.21%
	Negative/9		87.50%	77.78%	82.35%
Washing Machine	Positive/5	81.82%	80.00%	80.00%	80.00%
	Negative/6		83.33%	83.33%	83.33%
Average	Positive/50	81.75%	81.45%	79.01%	79.85%
	Negative/50		83.0%	84.92%	83.67%

The result shows that the system's performance for product reviews (i.e., 81.75% accuracy) and movie reviews (i.e., 85.5% accuracy) does not vary significantly. In the approach [Turney 2002], on the other hand, the movie review data achieved lower accuracy than product review data.

4.4 Evaluating Individual Components of SenseNet

In order to evaluate individual components of our system, we prepared different versions (or models) such that some rules are either present or absent. Since our system implements several rules to deal with adjectives, adverbs, negations, conditions and dependencies to get the contextual valence of the semantic verb frame(s) triplets (discussed in Contextual Valence Assessment), different versions of our system are realized by either considering or not considering the respective rules, as follows.

a. The 'no ADJ' version of the system does not consider the rules that handle the adjectives in contextual valence assessment. Thus for the sentence, *"in a time when so many movies are timid and weak, american history x manages to make a compelling argument for racism without advocating it any way*.", the 'no ADJ' version does not consider the adjectives 'timid', 'weak', 'compelling' while scoring this sentence. It hence outputs a lower score (i.e., 7.04) than the complete system (i.e., 10.81). In some cases (e.g., *"I would scale down the movie for its very poor visual effect."*), this version outputs complete different sentiment than that of the original system.

b. The 'no ADV' version of the system does not consider the rules dealing with adverbs. Thus, for an example positive review sentence, *"Animated film 'Monster House' rarely receive critical raves."* this model outputs a negative sentiment (-12.47) as it does not consider the adverb 'rarely'.

c. The 'no ADJ & no ADV' model is the combination of the two models above. Hence we expect to receive lower recall and F-scores for this system based on the hypothesis that both adjective and adverb are important linguistic components to assess sentiment from the text. Hence the hypothesis is supported by the obtained result given in Table 9. We notice that this model of the system received lower accuracy and average F-scores for all the datasets than that of the two models above.

d. The ‘no NEG & no CND’ version of the system does not consider negation and conditionality while calculating contextual valence. So, for a sentence present in Dataset B, *“it's a shame that his full talents were not used to full effect here.”*, the system assesses the first triplet as a negative one but the second one is assessed as positive for not considering the negation. Thus finally a positive valence is set for the sentence according to the rule for “not_to_dependency” triplets where a negative triplet precedes a positive triplet. Thus this model signals this sentence as a positive sentence although the sentence indicates a negative sentiment.

e. The ‘no Dependency’ version of the system does not consider the rules (as discussed in Sentiment Assessment) processing the dependency relationships between the triplets. Instead, it considers the average score of the triplets obtained from an input sentence. For the sentence present in Dataset B, *“the producers of this crow were either too dim to realize their story was doomed to be a hollow rehash, or too cynical to figure their audience would know the difference.*”, this model did not apply the rules that process dependency and thus miss-classified it as a positive sentence, whereas it is a part of a negative review and the original system scored it -9.61 to classify as a negative one.

The outcomes of the experimental results employing all the datasets by the models of the system discussed above are summarized in the Table 4.5. In the table Precision, Recall and F-Score are given for each individual class.

Table 4.5 Experimenting with different models of the system using all the datasets

Model	System Performance Measures					
	Datasets	Class	Accuracy (%)	Precision (%)	Recall (%)	F-Score (%)
no ADJ						
	Dataset A	Positive	61.5	68.18	66.67	67.42
		Negative		61.45	58.62	60.00
		Neutral		41.38	52.17	46.15
	Dataset B	Positive	75.26	77.58	73.21	75.33
		Negative		73.18	77.30	75.19
	Dataset C	Positive	67.05	68.42	61.30	64.66
		Negative		65.94	72.80	69.20
	Dataset D	Positive	70.00	69.81	74.00	71.84
		Negative		70.21	66.00	68.04
no ADV	Dataset A	Positive	76	83.53	78.89	81.14
		Negative		71.26	71.26	71.26
		Neutral		67.86	82.61	74.51
	Dataset B	Positive	80.10	78.74	79.40	79.07
		Negative		81.48	80.79	81.13
	Dataset C	Positive	76.70	78.59	70.10	74.10
		Negative		75.18	83.30	79.03
	Dataset D	Positive	60.00	53.45	62.00	57.41
		Negative		69.05	58.00	63.04
no ADJ & no ADV	Dataset A	Positive	55.5	59.34	60.00	59.67
		Negative		52.38	50.57	51.46
		Neutral		52.00	56.52	54.17
	Dataset B	Positive	63.82	65.39	62.00	63.65
		Negative		62.41	65.65	63.99
	Dataset C	Positive	49.05	36.00	33.30	34.60
		Negative		60.28	64.80	62.46
	Dataset D	Positive	48.00	43.75	56.00	49.12
		Negative		55.56	40.00	46.51

no NEG & no CND	Dataset A	Positive	73.5	75.79	80.00	77.84
		Negative		75.00	68.97	71.86
		Neutral		60.00	65.22	62.50
	Dataset B	Positive	81.98	84.33	84.47	84.40
		Negative		79.63	79.50	79.56
	Dataset C	Positive	78.30	79.64	75.50	77.52
		Negative		77.09	81.10	79.04
	Dataset D	Positive	69.00	68.75	66.00	67.35
		Negative		69.23	72.00	70.59
no Dependency	Dataset A	Positive	55	61.45	56.67	58.96
		Negative		54.76	52.87	53.80
		Neutral		39.39	56.52	46.43
	Dataset B	Positive	60.17	62.70	58.99	60.79
		Negative		57.92	61.34	59.58
	Dataset C	Positive	53.05	40.77	35.80	38.13
		Negative		62.66	70.30	66.26
	Dataset D	Positive	56.00	51.79	58.00	54.72
		Negative		61.36	54.00	57.45

We observe that the “no NEG & no CND” and “no Dependency” model shows the worst performance over all the datasets. This reinforces our belief that adjectives, adverbs, as well as the relationships among the semantic verb frames of a sentence are very important linguistic clues to assess the sentiment of text. Moreover in [Benamara et al. 2006], the results of experiments on an annotated set of 200 news articles (annotated by 10 students) lead to higher accuracy by aggregating scores of both adverbs and adjectives using three specific adverb-adjective-combinations (AACs) scoring methods. This finding is also supported by our experiment.

4.5 Comparison to a State-of-the System

Although the system *EmpathyBuddy* [Liu et al. 2003] does not directly assess sentiment of text (as our system does), it is known for its outstanding performance in analyzing emotion from text of smaller input size (e.g. a sentence). Like our system, Liu's system is a rule based system. It is said to be the best performing system for sentence-level emotion sensing. On the practical side, it is freely available on the internet, and thus easily available for comparison.

In order to compare the output of Liu's system to our scoring model, we considered 'fearful', 'sad', 'angry', and 'disgust' emotions as belonging to the negative sentiments, and 'happy' and 'surprise' as belonging to the positive sentiments. These are the emotions that *EmpathyBuddy* can recognize. The system considers 'surprise' as a positive emotion, and hence it resolves one of the example sentences mentioned in Liu et al. [2003], *"it's a gorgeous new sports car!"* as a positive one, which as the "surprise" emotion associated to it. For each sentence a vector containing the percentage value afferent to each emotion is returned by this system. For example, for the two sentences *"It is difficult to take bad photo with this camera."*, and *"Of all my relatives, I like my aunt Martha the best."*, *EmpathyBuddy* outputs the following sets of emotions along with their level of percentage: {surprised (67%), angry (38%), sad (31%), happy (0%), fearful (0%), disgusted (0%)} and { fearful (20%), happy (0%), sad (0%), angry (0%), disgusted (0%), surprised (0%)}.

In our analysis, we consider the highest percentage value from the positive or negative emotion group for each input sentence of our datasets obtained from their system. Thus for those two sentences, the first one is considered as positive and the other one as negative according to the output given by *EmpathyBuddy*. Table 4.6 summarizes the accuracy obtained for Dataset A and Dataset B from the experimental runs of the system where the valence range to signal neutrality is ±3.5 for Dataset A. This resulting average performance gain of our system is 11.17% and 12.86% with regard to accuracy for these two datasets respectively, when compared to [Liu et al. 2003].

While our system outperforms Liu's system in this setting, we want to emphasize that Liu's system was not designed for sentiment recognition. Hence, a direct (fair) comparison was not possible.

Table 4.6 Accuracy Comparison Metrics between *EmpathyBuddy* and *SenseNet*

Dataset A		Dataset B	
Our System	Liu's System	Our System	Liu's System
82%	70.83%	91.53%	78.67%

4.6 Discussion

The goal of the previous section was to compare our rule-based approach to other methods for sensing sentiment of text. For this purpose we performed experiments with four datasets. The results of the experiments indicate that our approach has an improving effect with regard to the classification of reviews. We could show that using our approach, accuracy and recall for Dataset B are improved over other methods (i.e., gain of 7.44% on accuracy). Table 4.3 shows that our approach attains a gain of 18.8 percentage points (from 66.7% to 85.5%) over non-machine learning approaches, when applied on movie review data (i.e., Dataset C). In general, our approach also shows better performance than machine learning approaches, with the exception of Hybrid SVM, which is 0.50 percentage points over our approach (see Table 4.3).

Since Dataset A and D are our original datasets, we could not compare the results to other methods. The specialty of Dataset A is that it has three types of sentences including neutral sentences. The experiment with this dataset revealed that if the valence range is ±3.5 to signal neutrality, the average recall and F-score are maximized to 81.04% and 76.49%, respectively. For Dataset D we achieved an accuracy of 81.75% with a recall of 79.01% and 84.92% for positive and negative sentence classes, respectively.

Movie reviews usually contain many sentences with ‘objective’ information about the characters or the plot of the movie. Although these sentences are ‘objective’ (in the sense of not being subjective) they may contain positive and negative terms. This is also true of movie titles, for example, “Ghost”, “Pirates of the Caribbean: Dead Man's Chest”, “Star Wars”, “Mission: Impossible”, “Die Another Day” etc. These are very positively reviewed movies, however their titles contain some negative terms. Repeating of such titles of the film in the review would make the review seem more negative (or positive for negative reviews for the titles with positive terms). Similar problems might exist for product reviews, maybe to a lesser extent (as pointed out by [Turney 2002]). In order to validate this claim we experimented with both types of data: movie review and product review. Datasets B and C are movie review data, and Dataset D contains product reviews. The percentage differences between the accuracy and average recall obtained for Dataset C and Dataset D are 3.75% and 3.54%, respectively, which indicates that the system shows better accuracy for movie review data than product review data. Hence, in our opinion, although there are objective sentences in the input text, and the system treats those objective sentences as if they were subjective, the average score of all the sentences of the whole input text is similar to the “gold standard” ranking.

Our system is robust in the sense that it can tackle the case where a negative term containing movie title for a positive review or vice versa may produce wrong outputs by keyword spotting or machine learning approaches. Since our system works on the basis of semantic structure of the sentence it considers the name as the subject or object of the sentence having attributes and emphasizes on the scoring of verb to which it is associated. Thus for the input sentence, *“at the end of the film pirates of the caribbean: dead man's chest, i was involved in the characters, and i was satisfied with the outcome.”*, the system found two positive verbs namely “involve” and “satisfy with” associated with the object “characters” and “outcome” where the object “characters” is having the attributes “film”, and “pirates of the caribbean: dead man's chest” which finally assign a positive contextual valence to the object "characters" according to our algorithm. Thus, our system output for this sentence is +8.453, that is, a positive sentence. On the other hand, keyword spotting based machine-learning and non-machine based approaches, such as ([Polanyi and Zaenen 2004], [Kennedy and Inkpen 2006]) will produce the wrong output for such cases.

Like the work of [Liu et al. 2003], our approach to sensing affective information from text relies on commonsense knowledge, which contributes to their robustness. Textual information (e.g. nouns) is mapped to concepts, which are derived from a large-scale real-world knowledge base of commonsense knowledge. The concepts usually have inherent affective connotation, such as 'positive' or 'negative', 'happy' or 'sad' etc. Hence for the input *"Mary was invited to Jack's party. She wondered if he would like a kite. She went and shook her piggy bank. It made no sound."*, humans apply commonsense to draw the following inferences: Gift is related with a party. Kite may be a gift item. Money is essential to buy a gift. If there is no coin in a piggy bank, no clattering sound is produced. No money and no gift make someone discouraged for the party. In this case, the commonsense model should relate "party" (i.e., positive event) to a "gift" (i.e., positive concept) concept and finally obtained a scenario mapped to negative concept "no money". Relating real-world scenarios to concepts and concepts to emotional affinity works well when the sentences are semantically simple and descriptive. But commonsense based approaches may fail for the sentences, *"You will hardly get a bad shot with this camera."*, and *"the three simple words you need to know in order to make your choice about owning your own ipod nano are: It's Sexy. It's Sleek. It's Small."*. They may fail because, first, they do not consider the semantic structure of the sentence and second, they may not have knowledge about concepts such as 'iPod' to assess emotional affinity.

Our approach overcomes such problems because we consider the semantic structure of the sentence and then assign the contextual valence based on the assessment of the semantic verb-frame(s). In fact we also have incorporated the commonsense knowledge in terms of assigning prior-valence values to words and implementing the rules to process the linguistic components for valence assignment. Moreover, we employ online resources to assess positive/negative opinions about new concepts (e.g., iPod), which might not (yet) or never be part of the commonsense knowledge base. In our opinion, our approach is robust and can be thought as an improvement over the commonsense based approach because commonsense approach maps a description to a collection of concepts and then concepts to their affective nature of everyday situations to classify sentences into "basic" emotion categories (i.e., either negative or positive), whereas our approach employs commonsense knowledgebase to assign words either a negative or positive score, considers the semantic structure of the sentence, and apply rules to assign the contextual valence of the so-called concepts (i.e., semantic verb-frame) and their associated relationships obtained from the sentence.

We are using different linguistic resources in order to assign prior valence to words. Our notion of 'prior valence' is sometimes called 'semantic orientation' (SO) in the literature ([Hatzivassiloglou and McKeown 2002]). We are aware of the procedures mentioned in [Turney and Littman 2003][Grefenstette et al. 2004], which employed a hit-result (of search engines) method to assign different semantic axes (i.e., positive or negative, excellent or bad, etc.) to words. Due to some limitations of SO approach mentioned in [Turney 2002][Turney and Littman 2003], we are motivated to apply a new approach that incorporates (1) WorldNet based manual scoring for verbs and adjectives, (2) commonsense knowledge to score nouns, and (3) Opinmind to score named-entities. In our future work we plan evaluate the scores obtained by our approach with respect to other approaches (e.g., SO).

Our system builds the computational model of the input sentence after the output of the language parser. We have noticed several problems with the language parser. For example, for the input, *"pretty cool movie though."*, the sentence/expression does not contain a verb, and hence the computation model (i.e., triplet) cannot be formed. In such cases we scored it by calculating the context valence considering the adjectives and nouns (i.e., similar to keyword spotting based approach). We observed that for the malformed or incomplete or too fragmented sentences the Semantic Parser sometime outputs erroneous triplets in terms of identifying the interdependencies between the triplets. For example, a sample review sentence, *"there's more, i suppose, but it's not worth it; the acting is bland, neither arsenic nor gravy; the music disposable; the camera work turgid."* , formed erroneous triplets because of possible missing verb in this part (i.e, *the music disposable;*) and the parser considered those linguistic components as the attributes of the last well formed triplet and the contextual valence is calculated thereby. Thus malformation of triplet might be one the sources of our errors.

We also observed that sometimes our approach of automatically assigning a new valence value to a non-scored new word outputs erroneous valence, which causes wrong classification of sentence. For example, for the input sentence, *"everything in the movie is so forced, so unauthentic that anyone with an i.q. over 80 will know they wasted their money on an unfulfilled desire."* , our automatic approach assigned a positive score (i.e., 1.363) for the adjective 'unauthentic' which made the evaluation of the first triplet (i.e., [['Subject Name:', 'sb/sth', 'Subject Type:', '', 'Subject Attrib:', []], ['Action Name:', 'force', 'Action Status:', 'Past Particle', 'Action Attrib:', ['ADV: so', 'passive', 'dependency: that']], ['Object Name:', 'everything', 'Object Type:', 'Object', 'Object Attrib:', ['Determiner: the', 'N NOM SG: movie', 'ADV: so', 'A ABS: unauthentic']]]) as a positive one, although it is negative. In our experience, the major reason for generating wrong outputs by the system is caused by this process of automatic scoring of new words. Hence we plan to revise our method of assigning prior valence values for new words by investigating other approach like affect control theory [Heise 2007] which assigns different scores (i.e., evaluation, potency and activity) for a word based on different social settings (e.g., culture, situations etc.).

4.7 Conclusion

This research is having a complementary research direction originating in Natural Language Processing to put emphasis on emotion sensing from text. We believe that text is an important modality for computer-human interaction, and sensing of textual affective information can significantly contribute to the success of affective user interfaces and intelligent machines. For the task of emotion sensing, text can also complement other modalities like speech or gesture (as reported in [Russell et al. 2003]), and thus increase the robustness of emotion recognition.

The system described in this chapter proposes a novel method to recognize sentiment at the sentence level. The system first performs semantic processing and then applies rules to assign contextual valence to the linguistic components in order to obtain sentence-level sentiment valence. The system is well-founded because we have 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 when compared to human performance as "gold standard". Moreover, it outperforms a state-of-the-art system (under simplifying assumptions). We also achieved better performance or almost similar performance while experimenting with machine learning approaches with the same datasets.

In general terms this research aims at giving computer programs a skill known as "emotional intelligence" with the ability to understand human emotion and to respond to it appropriately. We plan to extend the sentiment recognition system into a full-fledged emotion recognition system, which may classify named emotions rather than positive or negative sentiments. In the following chapter we implement the OCC emotion-model [Ortony et al. 1988] by applying different linguistic tools and heuristics to sense a rich set of affective information from the text.

Chapter Five: Emotion Analysis of Text

The focus of this chapter is to provide a set of rules for emotions as defined by the OCC emotion model, and to show how the rules can be implemented using natural language processing (NLP). We use the words ‘sentiment’ (e.g., good or bad) and ‘opinion’ (e.g., positive or negative) synonymously, and consider sentiment sensing as a task that precedes the task of “affect” or “emotion” (e.g., happy, sad, anger, hope etc.) sensing that has been performed by the developed system SenseNet. In other words, affect sensing or emotion recognition from text is the next step of sentiment recognition or opinion assessment from text. Emotion sensing requires a significantly more detailed analysis, since ultimately, it strives to classify twenty-two emotion types rather than two categories (such as ‘positive’ and ‘negative’).

5.1 Necessity of a New Approach

Although different types of emotions are expressed by text, research efforts so far were mostly confined to the simplifying case of recognizing positive and negative sentiment in text. (Observe that from a more general perspective, all emotions can be seen as positive or negative). A recent attempt described in [Liu et al. 2003] goes beyond the positive/negative dichotomy by aiming to sense six emotions. This is achieved by detecting associations between an event/concept and emotions, using commonsense knowledge of everyday life.

In our opinion, the emotion recognition capacity of this system is limited in the following aspects:

- It does not incorporate any semantic assessment of text (i.e., the contextual meaning of text),
- It does not consider the appraisal structure of emotions (i.e., the cognitive antecedent of a particular emotion, and
- It does not consider the variety of cognitive emotions that can be expressed by text (e.g., hope, love etc.)

To summarize, none of the models and techniques we have encountered so far has ever considered the cognitive structure of individual emotions. On the other hand, an emotion model which considers emotions as valenced reactions to the consequences of events, actions of agents and different aspects of objects as explained by the theory of emotion in [Ortony et al. 1988] has the potential to detect a large number of different from text. By way of example, it can detect the user's attitude towards events or objects as described in email, chat, blogs etc.

The approach described in this chapter can be considered as the extension of existing research for assessing sentiment of text by applying the valence assignment approach discussed above. We assume that a particular emotion a person experiences or describes in text on some occasion is determined by the way he/she construes the world. Thus the attempt of using only commonsense knowledge without considering the cognitive structure of emotions and a semantic interpretation of the words used in a sentence will fail to successfully recognize the emotion and the intensity of emotion. Therefore, the goal of this chapter is to describe a linguistic implementation of the OCC emotion model. This paradigm of content analysis will allow sensing emotions from texts of any genre (e.g., movie or product review; news articles; blogs-post etc.).

5.2 The OCC Emotion Model

In 1988, Ortony, Clore and Collins published the book titled *The Cognitive Structure of Emotions* [Ortony et al. 1988], which explores the extent to which cognitive psychology could provide a viable foundation for the analysis of emotions. Taking the first letters of the authors' names their emotion model is now commonly referred to as the OCC model. It is presumably the most widely accepted cognitive appraisal model for emotions. The authors propose three aspects of the environment to which humans react emotionally: (1) events of concern to oneself, (2) agents that one considers responsible for such events, and (3) objects of concern. These three classes of reactions or emotion eliciting situations lead to three classes of emotions, each based on the appraisal of different kinds of knowledge representations. They set forth the model to characterize a wide range of emotions along with the factors that influence both the emotion eliciting situations and intensity of each emotion, i.e. cognitive variables. According to the OCC model, all emotions can be divided into three classes, six groups and 22 types as shown in Figure 5.1. The model constitutes a systematic, comprehensive, and computationally tractable account of the types of cognition that underlie a broad spectrum of human emotions.

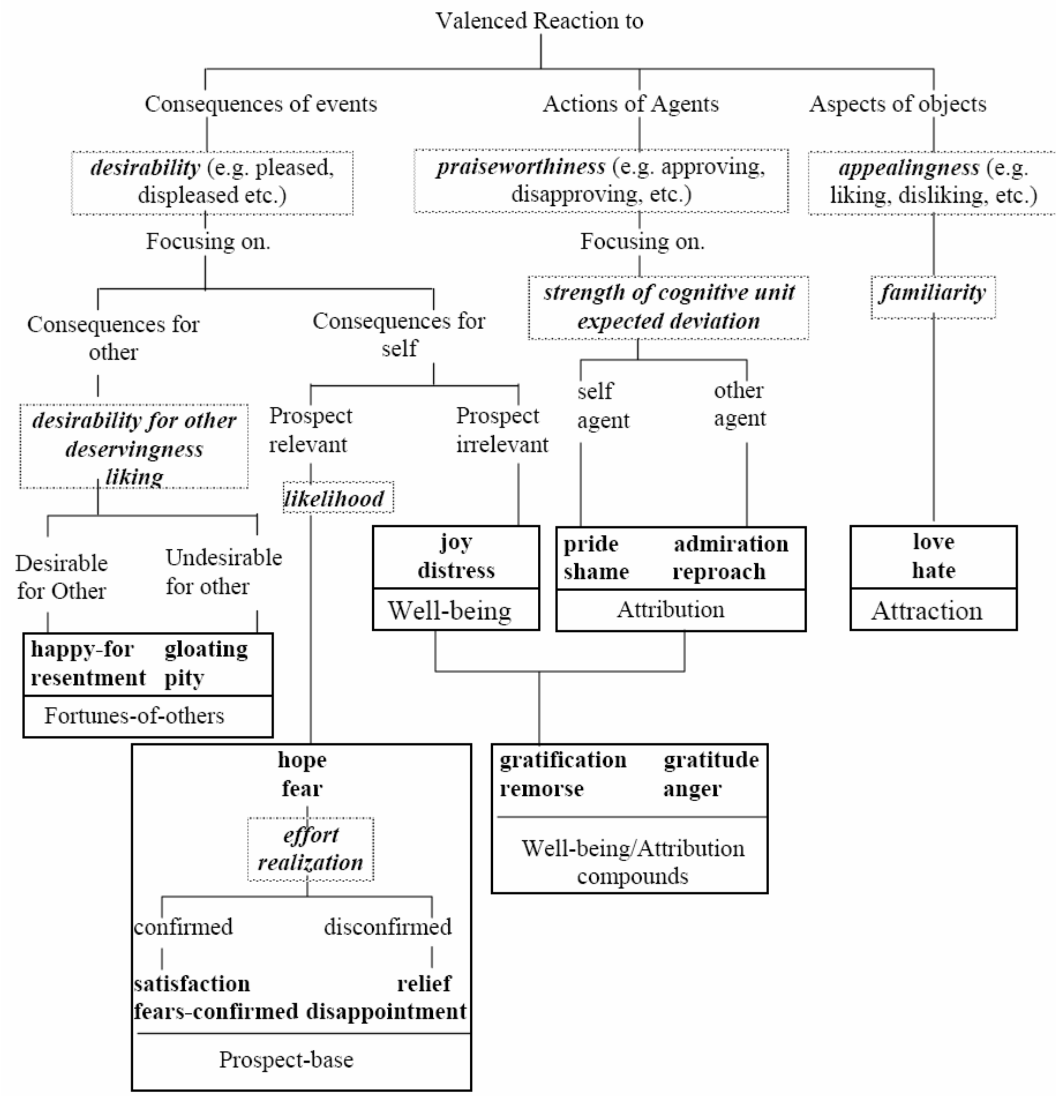
5.2.1 Why OCC Model?

The core motivation for choosing the OCC model is that it defines emotions as valanced reaction to events, agents, and objects, and considers valenced reactions as a means to differentiate between emotions and non-emotions. This approach is very suitable for affect sensing from text, also in view of the valence assignment approach mentioned above. Moreover, the OCC model constitutes a goal-, standard- and attitude-oriented emotion appraisal structure. As such, it provides an opportunity for applying natural language processing (NLP) to the identification of emotion inducing situations (e.g., event/action), the cognitive state of the user (usually expressed by adjectives and adverbs), and the variables causing emotion (e.g., real-world knowledge about something or somebody etc.).

In our search for relevant literature, we did not find any research that implements the OCC emotion model for the purpose of affect sensing from text. Yet, by incorporating intelligent text processing and semantic analysis, we can uncover the values that are needed as input to the antecedents of the rules for emotion recognition. The OCC model is widely used in Intelligent User Interfaces employing embodied life-like agents, in order to process feedback from the interaction partner (e.g. the user or another life-like agents), and to generate an appropriate emotional reaction as found in Bartneck 2003; Chengwei and Gencai 2001; Prendinger and Ishizuka 2005; Mr. Bubb in Space 2002.

However, we did not find any implementation of the OCC emotion model in the linguistic domain. In fact, the rule-based approach of the OCC emotion types and a rich set of linguistic tokens to represent those emotions, offer a sophisticated methodology and 'can do' approach for a computer program to sense the emotions expressed by textual descriptions. Hence this chapter describes how to apply an NLP method for emotion sensing based on the OCC emotion model.

Figure 5.1 The OCC Emotion Model



This figure has been re-produced by considering two original models [18, page 19 and 69], and are provided here for the easier reference. The ***bold-italic*** phrases indicate cognitive variables and **bold** phrases indicate emotion types.

5.2.2 Characterization of the OCC Emotions

The cognitive and appraisal structure of the OCC emotion types can be characterized by specific rules and their interplay with several variables. In Table 5.1 and 5.2 the variables and rules are listed, respectively. They directly relate to the OCC emotion model shown in Figure 5.1. The names of the variables are mostly self-explanatory. Some of them will be discussed in details.

The Variables. There are two kinds of variables, namely, *emotion inducing variables* (event, agent and object based) and *emotion intensity variables*. For our purpose, we characterize some of the variables slightly different from their definition in the OCC emotion model. The *event-based variables* are calculated with respect to the event, which is typically a verb-object pair found in the sentence. For example, the simple sentence, "*John bought Mary an ice-cream.*", describes an event of the form (buy, ice-cream). The abbreviations of variables are represented by bold italic letters in Table 5.1. In general we can call these variables "cognitive variables".

Table 5.1 the variables (i.e., cognitive variables) of the OCC Emotion Model

Variables for the OCC Emotion Types		
Type	Variable Name	Possible Enumerated Values
agent based	agent_fondness ***(af)***	liked, not liked
	direction_of_emotion ***(de)***	self, other
object based	object_fondness ***(of)***	liked, not liked
	object_appealing ***(oa)***	attractive, not attractive
event based	self_reaction ***(sr)***	pleased, displeased
	self_presumption ***(sp)***	desirable, undesirable
	other_presumption ***(op)***	desirable, undesirable
	prospect ***(pros)***	positive, negative
	status ***(stat)***	unconfirmed, confirmed, disconfirmed
	unexpectedness ***(unexp)***	true, false
	self_appraisal ***(sa)***	praiseworthy, blameworthy
	valenced_reaction ***(vr)***	true, false

intensity	event_deservingness ***(ed)***	high, low
	effort_of_action ***(eoa)***	obvious, not obvious
	expected_deviation ***(edev)***	high, low
	event_familiarity ***(ef)***	common, uncommon

The Rules for Emotion Types. The OCC emotion model specifies 22 emotion types and two cognitive states. Table 5.2 enlists the definitions of the 22 emotion types and the two cognitive states according to the OCC emotion model by employing the values of the variables mentioned in Table 5.1. The definitions are given in verbal form (rather than formalized form) for easier explanation and intuitive understanding of the emotion rules. In formalized form, these definitions are rules (Horn clauses), whereby the cognitive variables constitute the antecedent, and the emotion type is represented as the consequent (or head) of the rule. From a computational point of view, emotion recognition consists in inferring the set of emotions by rule application. Depending on whether states expressed by certain cognitive variables hold or do not hold, multiple emotions can be inferred from a given situation, i.e., the cognitive variables of one rule antecedent can be a proper subset of the antecedent of another rule (as e.g. for 'Joy' and 'Happy-for' in Table 5.2). This computational feature of the OCC rules is in accord with our intuition that text may express more than one type of emotion.

Table 5.2 the definitions of the rules for the OCC emotion types

Defining the OCC Emotion Types using the OCC Emotion Variables	
Emotion	Definition
Joy	Pleased about a Desirable event
Distress	Displeased about an Undesirable event
Happy-for	Pleased about an event Desirable for a Liked agent
Sorry-for	Displeased about an event Undesirable for a Liked agent
Resentment	Displeased about an event Desirable for another agent who is a Not Liked agent

Gloating	Pleased about an event Undesirable for another agent who is a Not Liked agent
Hope	Pleased about Positive Prospect of a Desirable Unconfirmed event
Fear	Displeased about Negative Prospect of an Undesirable Unconfirmed event
Satisfaction	Pleased about Confirmation of Positive Prospect of a Desirable event
Fears-Confirmed	Displeased about Confirmation of Negative Prospect of a Undesirable event
Relief	Pleased about Disconfirmation of Negative Prospect of an Undesirable event
Disappointment	Displeased about Disconfirmation of Positive Prospect of a Desirable event
Shock	Distress emotion with Unexpected Undesirable event
Surprise	Joy emotion with Unexpected Desirable event
Pride	Pleased for Praiseworthy action/event of Self
Shame	Displeased for Blameworthy action/event of Self
Admiration	Pleased for Praiseworthy action/event of Other
Reproach	Displeased for Blameworthy action/event of Other
Gratification	Higher Joy emotion with higher Pride emotion
Remorse	Higher Distress emotion with higher Shame emotion
Gratitude	Higher Joy with higher Admiration
Anger	Higher Distress with higher Reproach
Love	Liking an Attractive entity (e.g. agent or object)
Hate	Disliking an Unattractive entity

Now we briefly explain the idea for one emotion type (which will be explained in more detail below). "Happy-for" is characterized as: an agent *a* (actor in the sentence) senses *"Happy-for"* emotion towards someone/object *x*, for an event *e*, with respect to an input text *txt*, if 1) there found explicit affective lexicon(s) for "Joy" emotion type without any negation in the input text *txt* or 2) there is a valanced reaction (i.e., a certain degree of negative or positive sentiment, θ) to trigger emotion from *txt*, and the values of the associated cognitive variables (represented by bold-face) are as following: *a*'s **self-reaction** for *e* in *txt* is "pleased", **other-presumption** (i.e., *x*'s self-presumption) for *e* in *txt* is "desirable", **agent-fondness** is "Liked" (i.e., *a* Likes *x* in the context of *txt*), and **direction-of-emotion** is "other" (i.e., *a* and *x* are not the same entity).

5.3 Implementation of the OCC Model in Linguistic Realm

In order to implement the rules, first, we have to device how the values of the cognitive variables could be assigned by applying NLP techniques and tools. In this section, we explain how such values can be assigned to the cognitive variables of the emotion rules.

5.3.1 Linguistic Resources

In this sub-section, we explain the different linguistic resources being utilized. They partly rely on other reliable sources.

Semantic Parser. We have created a semantic parser using Machinese Syntax [Machinese Syntax] that produces XML-formatted syntactic output for the input text. For example, for the input sentence, *"My mother presented me a nice wrist watch on my birthday and made delicious pancakes."*, the output tuples of the semantic parser are shown in Table 5.3. It outputs each semantic verb-frame of a sentence as a triplet of "subject-verb-object". So the parser may output multiple triplets if it encounters multiple verbs in a sentence. The output given in Table 5.3 has two triplets, which are mutually dependent, as indicated by the 'dependency' keyword in the action attribute of Triplet 2. This Semantic Parser outputs the computational data model for each sentence. This semantic parser is the part of the SenseNet which is already discussed in chapter three.

Table 5.3 Semantic verb-frames output by the semantic parser for "My mother presented me a nice wrist watch on my birthday and made delicious pancakes."

Tuple Output of Semantic Parser	
Triplet 1	[[*'Subject Name:'*, 'mother', *'Subject Type:'*, 'Person', *'Subject Attrib:'*, ['PRON PERS GEN SG1:i']], [*'Action Name:'*, 'present', *'Action Status:'*, 'Past ', *'Action Attrib:'*, ['time: my birthday', 'Dependency: and']], [*'Object Name:'*, 'watch', *'Object Type:'*, 'N NOM SG', *'Object Attrib:'*, ['Determiner: a', 'A ABS: nice', 'N NOM SG: wrist', 'Goal: i']]]
Triplet 2	[[*'Subject Name:'*, 'mother', *'Subject Type:'*, 'Person', *'Subject Attrib:'*, []], [*'Action Name:'*, 'make', *'Action Status:'*, 'Past ', *'Action Attrib:'*, []], [*'Object Name:'*, 'pancake', *'Object Type:'*, 'N NOM PL', *'Object Attrib:'*, ['A ABS: delicious']]]

These two triplets indicate two events, *(present, watch)* and *(make, pancake)*. The actor for both the events is "mother". In this case, the triplets also contain additional attributes which give more information about the events.

Scored List of Action, Adjective, Adverb, and Nouns. Initially eight judges have manually counted the number of positive and negative senses of each word of a list of verbs, adjectives, and adverbs according to the contextual explanation of each sense found in WordNet 2.1. The results are maintained as lists scored words. In chapter three how a prior-valence value is assigned to a word is discussed. Using the Equations (2)-(3) the each verb is also assigned two more values namely, prospective value and praiseworthy value. Adjectives and adverbs have valence values only.

$$\text{prospect polarity} = (\text{Positive-Sense Count} > \text{Negative-Sense Count})\ ?\ 1 : -1 \tag{2}$$

$$\text{prospective value} = \text{Average}(\max(\text{Positive-Sense Count}, \text{Negative-Sense Count})/\text{Total Sense Count}) * 5.0 * \text{Prospect Polarity})$$

$$\text{praiseworthy value} = \text{Average}\ (\text{prior valence} + \text{prospective value}) \tag{3}$$

The prior valence, prospective and praiseworthy values indicate the lexical affinity of a word with respect to "good or bad", "desirable or undesirable", and "praiseworthiness or blameworthiness", respectively. The "prospective value" and "praiseworthy value" of a verb/action word are necessary to evaluate an event according to the OCC emotion model. A value between -5 to 5 is assigned as the valence for a noun or concept. SenseNet maintains a list of scored nouns and named-entities. The scoring procedure is already described in chapter three.

SenseNet. It calculates the contextual valence of the words using rules and prior valence values of the words. It outputs a numerical value ranging from -15 to +15 flagged as the 'sentence-valence' for each input sentence. As examples, SenseNet outputs -11.158 and +10.466 for the inputs, *"The attack killed three innocent civilians."* and *"It is difficult to take bad photo with this camera,"* respectively. These values indicate a numerical measure of negative and positive sentiments carried by the sentences. The accuracy of SenseNet to assess sentence-level negative and positive sentiment has been reported to vary from 92% to 82% in experimental studies on different datasets as mentioned in chapter four. In our approach to sensing affect the output of SenseNet for each sentence is considered as the "valenced reaction" (*vr*) which is the triggering variable to control the mechanism of emotion recognition to branch towards either positive or negative groups of emotion types or signal neutrality.

5.3.2 Assigning Values to the Variables

In this subsection we will describe the cognitive variables listed in Table 5.2 and explain how the enumerated values can be assigned to those variables using the aforementioned linguistic resources. The notion of "self" refers to the computer program or the system itself which senses the emotion from the text. We assume that the system usually has positive sentiment towards positive concept and vice versa. For example, in general the events "pass examination" and "break up friendship" give 'positive' and 'negative' sentiments, respectively, and the system also considers those as 'positive' and 'negative' events. Moreover we consider the system as a positive valenced actor or entity while assessing an event from the "self" perspective.

Self Presumption (*sp*) and Self Reaction (*sr*). According to the appraisal structure of an event after the OCC model, the values for the variables self_presumption (*sp*) and self_reaction (*sr*) are "desirable" or "undesirable", and "pleased" or "displeased", respectively. These variables are assessed with respect to the events. For example, for the events *"buy ice-cream"*, *"present wrist watch"*, *"kill innocent civilians"* referred in the example sentences, SenseNet returns contextual valence as +7.83, +8.82, and -8.46, respectively. According to the SenseNet scoring system, the valence range for an event (i.e., verb, object pair) lies between -10 to +10. Thereby we decide that for an event, if the valence is positive (e.g., "buy ice-cream"), *sp* and *sr* are set as "desirable" and "pleased", and in the case of negative valence (e.g., "kill innocent civilian") both *sp* and *sr* are set to "undesirable" and "displeased", respectively. But for the sentences like, *"I like romantic movies"* and *"She likes horror movies"* the positive action "like" is associated with a positive concept (i.e., romantic movie) and negative concept (i.e., horror movie) that eventually give "desirable & pleased" and "undesirable & displeased" assessment for the events, respectively. But both events are conveying positive sentiment because positive affect is being expressed by the word "like". In order to deal with such cases, SenseNet has a list of positive and negative affective verbs (e.g., love, hate, scare etc.) that deals with the heuristic that if a positive affect is expressed by an event, the event is considered as positive irrespective to the polarity of the object and vice versa. Thus for the events, *"loathe violence"* and *"loathe shopping"* are assessed as "undesirable" and "displeased" due to negative affective verb "loathe".

Other Presumption (*op*). As above, the values for other_presumption (*op*) could be set "desirable" or "undesirable" while assessing the event from the perspective of the agent pertaining to an event being assessed.

We explain the heuristic of assigning value to this variable by using several examples. For the sentence *"A terrorist escaped from the Jail."*, the value of *op* (for the event "escape from jail") is presumably "desirable" for the agent "terrorist" because the contextual valence of the event *"escape from jail"* is negative (i.e., -6.715) which is associated with a negative valenced actor *"terrorist"* (i.e., -3.620). For simplicity, we assume that a negative actor usually desires to do negative things. Thus the event *"escape from jail"* is usually "desirable" by the actor *"terrorist"* under this simplified assumption. But for the same event, it is "undesirable" and "displeased" for *sp* and *sr* because of negative valence (i.e., -6.715). Thus in this case, we set *op* as "desirable" because of having a negative valenced event associated with a negative valenced agent. Similarly we provide the following simple rules to assign the values to *op* for other cases.

- If a positive valenced event is associated with a positive valenced agent, *op* is set to "desirable"; e.g., *"The teacher was awarded the best-teacher award. John bought a new car."* The first sentence is a passive sentence where the actor is *"someone"* according to the output of Semantic Parser and object is *"teacher"* having attribute *"best-teacher award"*. The agent *"someone"* has the positive valence (i.e., +4.167) and the event *"award teacher best-teacher award"* also has positive valence (i.e., +8.741), hence the event's *op* value is "desirable" with respect to the agent. For the second sentence, *"John"* is the actor associated with a positive event (i.e., "buy new car"), and hence the *op* value for the event is "desirable" considering *"John"* as a positive named-entity.
- If a negative valenced event is associated with a positive valenced agent, *op* is set to "undesirable"; e.g., "*Boss sacked the employee from the job. Teacher punished the boy for breaking the school rule."* For, the first sentence the negative valenced event (i.e., -7.981) "sack employee from job" is associated with a positive valenced actor (i.e., +3.445) "Boss", hence the *op* value for the event is "undesirable" for *"Boss"*. Similarly, for the second

sentence the *op* value for the event *"punish boy for breaking the school rule"* is also "undesirable" for *"Teacher"*.

- If a positive valenced event is associated with a negative valenced agent, *op* is set "undesirable"; e.g., in the sentence "*the kidnapper freed the hostage."*, a negative valenced actor (i.e.,-4.095) *"kidnapper"* is associated with a positive valenced event (i.e, +5.03) *"free the hostage"* and hence the *op* value for this event is set "undesirable". This simplified rule may produce peculiarities, for example, *"the murderer lived in this town."* or *"the criminal loved a girl."* sentences have positive valenced events *"live in this town"* and *"love a girl"* associated with negative valenced actors *"murderer"* and *"criminal"*, respectively. According to the rule for both of these events, the *op* value will be set to "undesirable" for the agents. For simplicity we assume that any positive event is not expected for a negative role actor. But in the future, we would like to deal with such cases more deeply.

Direction of Emotion (*de*). Depending on whether the agent that experiences some emotion is reacting to consequences of events for itself or to consequences for others, the system sets the value of the variable 'Direction of Emotion' *(de)* as either 'self' or 'other'. If 'other' is set the emotion being recognized belongs to 'fortune-of-others' emotion group and the recognized emotion is anchored to the author or the subject of the event. This value for *de* is set as "other" if the object or the predicate of the event described in the text is a person (e.g., John) or a personal pronoun (e.g., I, he) according to the triplet output given by the Semantic Parser; otherwise it is set as "self". For the sentences, *"Mary congratulates John for having won a prize."*, and *"I heard Jim having a tough time in his new job."* the value of *de* is set "other" for having "John" and "Jim" as the "person" objects detected by Semantic Parser. Thus the value of *de* being set as 'other' makes our system recognize an emotion from the 'fortunes-for-other' emotion group and eventually emotions like "happy-for", "sorry-for" would be recognized from the given sentences anchored to the author for the objects of the events. Additionally the system will also recognize that the authors/agents of the events (e.g., 'Mary' and 'I') are with "joy" and "distress" while considering the "well-being" emotion group. But, for the sentence, *"Susan won the million dollar lottery."*, "*It is a very interesting idea.",* the value of *de* is set "self" which eventually indicates that the sensed emotion is anchored to the author himself and the system will not proceed to recognize any 'fortunes-of-others' emotion types.

Prospect (*pros*). According to the OCC model, prospect of an event involves a conscious expectation that it will occur in the future, and the value for the variable prospect (*pros*) can be either "positive" or "negative". According to the aforementioned Equation (2), SenseNet considers either the positive or negative sense-count (whichever is the maximum for a verb) to calculate "prospective value". The heuristic behind this score is to know the semantic orientation (SO) of a verb with respect to "optimism" and "pessimism". According to the equation, if a verb has more positive senses than negative senses, the verb is more close to "optimism" and has positive prospective value; otherwise, the verb gets a negative prospective value.

In order to assign *pros* value to an event, we consider the 'prospective value' of the verb instead of 'prior-valence' of that verb. Empirically we have set that if the valence of some event is higher than or equal to +3.5, the *pros* value of that event is set "positive". Similarly, if the valence is less than or equal to -3.5, the *pros* value for the event is "negative". Otherwise, *pros* value is set to "neutral". For example, for the events *"admit to university", "kill innocent people",* and *"do it",* SenseNet returns +9.375, -8.728, and +2.921 as the valence values for the events, respectively, and according to the values, *pros* values of the events are set to "positive", "negative" and "neutral", respectively.

Status (*stat*). The variable status (*stat*) has values like "unconfirmed", "confirmed" and "disconfirmed". If the tense of the verb associated with the event is present or future or modal, the value is set to "unconfirmed" for the event. For examples, *I am trying to solve it. He will come. The team may not play.*, the tense of the verbs are "present", "future" and "modal", respectively, and hence the *stat* value of the events is “unconfirmed”. If the verb of the event has positive valence and the tense of the verb (with or without a negation) is past, *stat* is set “confirmed” (e.g., *I succeeded. He didn’t come. The team played well.*). Again, if the verb of the event has negative valence and the tense of the verb is past without a negation, the value of *stat* is also set “confirmed” (e.g., *The hostage was killed. The team defeated its opponent.*). But if the verb of the event has negative valence and the tense of the verb is past with a negation, *stat* is set “disconfirmed” (e.g., *I didn’t fail. He didn’t hit the boy. The team couldn’t defeat its opponent*).

Agent Fondness (*af*). If the valence of the agent or object associated with the event is positive, “liked” is set to the variables agent_fondness (*af*) and object_fondness (*of*); otherwise “not-liked” is set. For example, for the sentences, *“The hero appeared to save the girl.”*, and *“A terrorist escaped from the jail”*, *af* for *“hero”* and *“terrorist”* is set to “liked” and “not-liked”, respectively, because of positive and negative valence returned by SenseNet. In the same manner the value of *of* is set “liked” and “not-liked” for the objects *“girl”* and *“Jail”*, respectively.

Self Appraisal (*sa*). According to the appraisal structure of an event mentioned in the OCC model, the value for self_appraisal (*sa*) can be either “praiseworthy” or “blameworthy”. In the aforementioned Equation (3), SenseNet takes the average of “prior valence” and “prospective value” of a verb to assign the praiseworthy value of that verb. The praiseworthy value is considered as the semantic orientation score of a verb with respect to “praise” and “blame”. To assign the *sa* value to an event, the “praiseworthy value” of the verb is considered instead of its ‘prior-valence’. Empirically we have set that if an event’s valence is more than or equal to +4.5, the *sa* value of that event is set “praiseworthy”. Similarly, if the valence is less than or equal to -4.5, the *sa* value for the event is “blameworthy”. Otherwise *sa* value is set to “neutral”. For example, let’s consider these events, *“pass final exam”*, *“forget friend’s birthday”*, and *“kick ball”*. For these events SenseNet returned +7.95, -9.31, and -3.87, respectively. Thereby for the events discussed above, the value for *sa* is set “praiseworthy”, “blameworthy”, and “neutral”, respectively.

Object Appealing (*oa*). The value of object_appealing (*oa*) indicates whether an object is “attractive” or “not attractive”. In order to assign a value to *oa* for an object, we consider two scores, namely, ‘object valence’ and ‘familiarity valence’ with the following heuristic. The value “attractive” is set if the object has a positive ‘object valence’ with a ‘familiarity valence’ less than a certain threshold. Reversely “not attractive” is set if the object has a negative ‘object valence’ with a ‘familiarity valence’ above than a certain threshold. The ‘familiarity valence’ is obtained from ConceptNet by calculating the percentage of nodes (out of 300,000 concept-nodes) linking to and from the given object/concept. For example, the ‘familiarity valence’ for the object/concept *“restaurant”*, *“thief”,* and *“diamond ring”* are 0.242%, 0.120%, and 0.013%, respectively. Empirically we kept the value 0.10% as the threshold value to signal familiarity and unfamiliarity of an object. Basically the ‘familiarity valence’ indicates how common or uncommon the input object is with respect to the commonsense knowledge-base corpus. According to our heuristic an object which is relatively uncommon and bears a positive sense is usually “attractive”. On the contrary, an object which is relatively common and shows a negative concept is usually “not attractive”. Thus *“diamond ring”* and *“thief”* appears to be “attractive” and “not attractive”*,* respectively, but “restaurant” receives the value ‘neutral’ due to being positive and too common.

Valenced Reaction (*vr*). The value for valenced_reaction (*vr*) is set either “true” or “false” in order to initiate further analysis to sense emotions or decide the sentence(s) as expressing a neutral emotion. We consider *vr* to be “true” if the ‘sentence-valence’ returned by SenseNet is either above than 3.5 or less than -3.5. For example, *“I go.”*, does not lead to further processing (i.e., sentence-valence is +3.250) but *“I go to gym everyday.”*, yields an emotion classification because of the higher ‘sentence-valence’ (i.e., +7.351). This value indicates the negative or positive sentiment of the input sentence numerically. Thus we call this variable as the trigger for a further emotion analysis process.

Unexpectedness (*unexp*). In English language, there are some words that express suddenness of an event. The variable *unexp* indicates whether an event (either positive or negative) is described in an abrupt or sudden manner. The value to the variable *unexp* is set "true" if there is a linguistic token to represent suddenness (e.g., abruptly, suddenly etc.) of the event in the input sentence; otherwise "false" is set. We have a list of such tokens to indicate suddenness. For example, the sentences *"I met my friend unanticipatedly at the bar."*, and *"The storm hit the city without warning."* indicate two events *"meet friend"* and *"hit city"* expressed with a kind of suddenness represented by two of our listed linguistic clue words namely, 'unanticipatedly' and 'without warning'. Hence the value for *unexp* is set true for both events.

Event Deservingness (*ed*). The OCC model has several variables to signal emotional intensity. Event deservingness is one of them. It indicates the degree to which 'self' desires the event for 'oneself' as well as for 'others'. Actually there are two types of variables: 'Event Deservingness for Self' and 'Event Deservingness for Other'. In this case we assign the same value obtained towards an event for 'Event Deservingness for Self' to the variable 'Event Deservingness for Other'. This implies that what someone deserves for oneself, to the same extent that event is deserved for other by that someone. For that reason our model is not able (i.e., designed not to be able) to infer 'resentment' and 'gloating' type ill-will emotions. In the model, the value for the intensity variable event_deservingness (*ed*) is set "high" for an event having a higher positive valence (i.e., above +7.0) or "low" for higher valence in negative scale (i.e., less than -7.0). The values are set empirically.

Effort of Action (*eoa*). According to the OCC model, when effort is a factor, the greater the effort invested, the more intense the emotion. It is difficult to answer a question like, *"has any effort been realized for the event?"*, from a single sentence. However we make the simplified assumption that, if an action is qualified with an adverb (except the exceptional adverbs like hardly, rarely listed in SenseNet), e.g., *He worked very hard*, or target object qualified with an adjective (e.g., *I am looking for a quiet place*) without a negation, the value for effort_of_action (*eoa*) is set "obvious"; otherwise "not obvious".

Expected Deviation (*edev*). The variable called expected_deviation (*edev*) indicates the difference between the event and its actor in terms of expectancy of the event being associated with the actor. For example, in the sentence *"The police caught the criminal finally."*, the actor *"police"* and the event *"catch criminal"* do not deviate because the action is presumably expected by the actor. We set the value for *edev* to "low" if ConceptNet can find any semantic relationship between the actor and event; otherwise "high" is set. For example, for sentence *"a student invented this theory."*, *edev* is set "high" because ConceptNet doesn't return any relationship between "student" and "invent". On the contrary, for the sentence *"The scientist invented the theory."*, the value of edev is "low" because ConceptNet finds a conceptual relationship between the action *"invent"* and actor *"scientist"*.

Event Familiarity (*ef*). The values "common" or "uncommon" are set for event_familiarity (*ef*) according to the average familiarity valence obtained from ConceptNet for the action and object of the event. For example, the events *"eat sushi"* and *"buy diamond ring"*, we obtain the familiarity score for 'eat' as 0.205, 'sushi' as 0.062, 'buy' as 0.198 and 'diamond ring' as 0.013. This gives the familiarity score for the events as, 0.134 and 0.106, respectively. Empirically we have set the value less than 0.15 to set "uncommon", else "common" as the value of *ef* for the event.

5.3.3 The Rules of the OCC Emotion Types

In the sub-section 5.2.2 we have illustrated how a rule for an emotion defined in the OCC model (e.g., happy-for) can be characterized using the values of the associated cognitive variables, and in the sub-section 5.3.2 we have explained how specific values can be assigned to the cognitive variables. Now we enlist the rules for emotion types from the OCC model. Although in input text, *txt,* there might be multiple events, *e,* described and we also deal with such cases to receive the resultant emotion types from *txt*, but the following rules are described for a single event, *e*. We will provide an example involving multiple events in section 5.4. Hence, the rules for the OCC emotion types are given assuming an event *e* described in text *txt*.

By way of example, the actor or author of an event *e* feels the emotion 'Joy' if following condition is true:

[Linguisitc_Token_found_for_Joy(*txt*) and No_Negation_Found (*txt*)] or [*vr*= true and *sr*= "pleased" and *sp*= "desirable"] (i.e., literally 'Joy' means that the author or the agent of the event is 'pleased about the event which is desirable'.)

Since we have the token words for each emotion types, we omit the first condition in the subsequent rules for space limitations. The rules for the emotion are listed as following.

- if (*vr*=true & *sr*="displeased" & *sp*="undesirable" & *de*="self"), "distress" is true.
- if (*vr*=true & *sr*="displeased" & *op*="undesirable" & *af*="liked" & *de*="other"), "sorry-for" is true.
- if (*vr*=true & *sr*="displeased" & *op*="desirable" & *af*="not liked" & *de*="other"), "resentment" is true.
- if (*vr*=true & *sr*="pleased" & *op*="undesirable" & *af*="not liked" & *de*="other"), "gloating" is true.
- if (*vr*=true & *sr*="pleased" & *pros*="positive" & *sp*="desirable" & *status*="unconfirmed" & *de*="self"), "hope" is true.
- if (*vr*=true & *sr*="displeased" & *pros*="negative" & *sp*="undesirable" & *status*="unconfirmed" & *de*="self"), "fear" is true.

- if (*vr*=true & *sr*="pleased" & *pros*="positive" & *sp*="desirable" & *status*="confirmed" & *de*="self"), "satisfaction" is true.
- if (*vr*=true & *sr*="displeased" & *pros*="negative" & *sp*="undesirable" & *status*="confirmed" & *de*="self"), "fears-confirmed" is true.
- if (*vr*=true & *sr*="pleased" & *pros*="negative" & *sp*="undesirable" & *status*="disconfirmed" & *de*="self"), "relief" is true.
- if (*vr*=true & *sr*="displeased" & *pros*="positive" & *sp*="desirable" & *status*="disconfirmed" & *de*="self"), "disappointment" is true.
- if (*vr*=true & *sr*="pleased" & *sa*="praiseworthy" & *sp*="desirable" & *de*="self"), "pride" is true.
- if (*vr*=true & *sr*="displeased" & *sa*="blameworthy" & *sp*="undesirable" & *de*="self"), "shame" is true.
- if (*vr*=true & *sr*="pleased" & *sa*="praiseworthy" & *op*="desirable" & *de*="other"), "admiration" is true.
- if (*vr*=true & *sr*="displeased" & *sa*="blameworthy" & *op*="undesirable" & *de*="other"), "reproach" is true.
- if (*vr*=true & *sp*="desirable" & *sr*="pleased" & *of*="liked" & *oa*="attractive" & event valence="positive" & *de*="other"), "love" is true.
- if (*vr*=true & *sp*="undesirable" & *sr*="displeased" & *of*="not liked" & *oa*="not attractive" & event valence="negative" & *de*="other"), "hate" is true.

The OCC model has four complex emotions, namely, "gratification", "remorse", "gratitude" and "anger". The rules for these emotions are:

- If both "joy" and "pride" are true, "gratification" is true.
- If both "distress" and "shame" are true, "remorse" is true.
- If both "joy" and "admiration" are true, "gratitude" is true.

- If both “distress” and “reproach” are true, “anger” is true.

The cognitive states ‘shock’ and ‘surprise’ are ruled as;

- If both “distress” and *unexp* are true, “shock” is true. (e.g., *the bad news came unexpectedly.*)

If both “joy” and *unexp* are true, “surprise” is true. (e.g., *I suddenly met my school friend in Tokyo University.*)

5.4 Walk-Through Examples for Emotion Recognition

Like [Liu et al. 2003], we also believe that a statement may express more than one emotion type. According to the OCC model, the 22 emotion types and two cognitive states are grouped into seven groups, namely, well-being emotion, fortune of other emotion, prospect based emotion, cognitive state, attribution emotion, attraction emotion, and compound emotion. Hence an input sentence may contain one of the emotion types from each group. In the following, we provide a detailed analysis of emotion recognition from an example text.

Our example sentence is: *“I didn’t see John for the last few hours; I thought he might miss the flight but I suddenly found him on the plane.”*

The output from the Semantic Parser is given below:

Triplet 1: [['Subject Name:', 'i', 'Subject Type:', 'Person', 'Subject Attrib:', []], ['Action Name:', 'see', 'Action Status:', 'Past', 'Action Attrib:', ['negation', 'duration: the last few hours ', 'dependency: and']], ['Object Name:', 'john', 'Object Type:', 'Person', 'Object Attrib:', []]]
Triplet 2: [['Subject Name:', 'i', 'Subject Type:', 'Self', 'Subject Attrib:', []], ['Action Name:', 'think', 'Action Status:', 'Past', 'Action Attrib:', ['dependency: to']], ['Object Name:', '', 'Object Type:', '', 'Object Attrib:', []]]
Triplet 3: [['Subject Name:', 'john', 'Subject Type:', 'Person', 'Subject Attrib:', []], ['Action Name:', 'miss', 'Action Status:', 'Modal Infinitive ', 'Action Attrib:', ['dependency: but']], ['Object Name:', 'flight', 'Object Type:', 'Entity', 'Object Attrib:', ['Determiner: the']]]
Triplet 4: [['Subject Name:', 'i', 'Subject Type:', 'Person', 'Subject Attrib:', []], ['Action Name:', 'find', 'Action Status:', 'Past ', 'Action Attrib:', ['ADV: suddenly', 'place: on the plane']], ['Object Name:', 'john', 'Object Type:', 'Person', 'Object Attrib:', []]]

According to the output, Triplet 2 has “dependency: to” relationship with Triplet 3. Then these two triplets are considered as a combined event. Hence there are three events as indicated below:

- *e1*: “not see john the last few hours”, [agent: I, tense: ‘Past’, 'dependency: and']
- *e2*: “think <no obj>, might miss flight” [agent: John, object: flight, tense: ‘Modal’, dependency: but]
- *e3*: “find john on the plane” [agent: I, tense: ’Past’]

In event *e2*, there are two sub-events and for simplicity we consider the second sub-event’s subject as the agent, action status as the status, and object as the object of the event while assigning values to the cognitive variables for this event. However, while assessing the event valence the sub-events are treated individually, and SenseNet has implemented specific rules to assign contextual valence for the triplets having “dependency: to” relationship to the other. The rest of the analysis is summarized in the following Table 5.4.

Table 5.4 Recognizing the OCC emotions from the sentence “I didn’t see John for the last few hours; I thought he might miss the flight but I suddenly found him on the plane.”

Analysis of the recognition of OCC emotions for the given example sentence			
Events	*e1*	*e2*	*e3*
Event Dependency	dependency: and	dependency: but	
SenseNet Value (returned for each event)	event valence:-9.33 prospect value:-9.11 praiseworthy val:-9.22 agent valence:+5.0 object valence:+4.2	event valence:-8.69 prospect value:-7.48 praiseworthy val:-8.09 agent valence:+4.2 object valence:+2.72	event valence:+9.63 prospect value:+8.95 praiseworthy val:+9.29 agent valence:+5.0 object valence:+4.2
ConceptNet Value	familiarity valence: ‘john’ 0.059% ‘see’ 0.335% action-actor deviation: “I-see”: null	familiarity valence: ‘flight’ 0.113% ‘miss’ 0.14% action-actor deviation: “john-miss”: null	familiarity valence: ‘john’ 0.059% ‘find’ 0.419% action-actor deviation: “I-find”: null

Values of Cognitive Variables	*of*: liked *de*: other *oa*: attractive *sr*: displeased *sp*: undesirable *pros*: negative *stat*: confirmed *unexp*: false *sa*: blameworthy *vr*: true *ed*: low *eoa*: not obvious *edev*: low *ef*: common	*of*: liked *af*: liked *de*: self *oa*: neutral *sr*: displeased *sp*: undesirable *op*: undesirable *pros*: negative *stat*: unconfirmed *unexp*: false *sa*: blameworthy *vr*: true *ed*: low *eoa*: not obvious *edev*: low *ef*: uncommon	*of*: liked *de*: other *oa*: attractive *sr*: pleased *sp*: desirable *pros*: positive *stat*: confirmed *unexp*: true *sa*: praiseworthy *vr*: true *ed*: high *eoa*: obvious *edev*: low *Ef*: common
Apply Rules Phase 1	distress, sorry-for, fears-confirmed, reproach	distress, fear, shame	joy, happy-for, satisfaction, admiration
Apply Rules Phase 2	fears-confirmed, sorry-for, anger	fear, remorse	happy-for, satisfaction, gratitude
Apply 'and'-logic	sorry-for, fears-confirmed, anger		happy-for, satisfaction, gratitude
Apply 'but'-logic	happy-for, relief, gratitude		

The values of the cognitive variables are set according to the explanation given in sub-section 5.3.2 on the basis of the values obtained from SenseNet and ConceptNet modules. Phase 1 shows the list of emotions for each event after applying the rules for simple OCC emotions. Phase 2 shows more refined sets of emotions after applying the rules for complex OCC emotions.

Since the first event is having an ‘and’ relationship with the second one, ‘add’-logic is applied to the set of emotions resolved from the events *e1* and *e2*. The rule of applying ‘add’-logic is to simplify two emotions and keep one of the two emotions by applying the following rules. The rules are developed from the motivation of [Ortony 2003] where one of the authors of the OCC model described regarding the collapsing of OCC-defined emotions into five specializations of generalized good and bad feelings.

- ‘hope’ and ‘satisfaction’ are collapsed to ‘satisfaction’
- ‘fear’ and ‘fear-confirmed’ are collapsed to ‘fear-confirmed’
- ‘pride’ and ‘gratification’ are collapsed to ‘gratification’
- ‘shame’ and ‘remorse’ are collapsed to ‘remorse’
- ‘admiration’ and ‘gratitude’ are collapsed to ‘gratitude’
- ‘reproach’ and ‘anger’ are collapsed to ‘anger’
- ‘gratitude’ and ‘gratification’ are collapsed to ‘gratitude’
- ‘remorse’ and ‘anger’ are collapsed to ‘anger’

At this stage, we find all the possible emotions that the sentence is expressing. We believe that a part of a sentence (i.e., in a complex sentence) may express a negative emotion while the other part may express positive emotion and vice versa. So we can say that the example sentence is expressing this set of emotions: {fears-confirmed, sorry-for, anger, happy-for, satisfaction, gratitude}. Yet we proceed further by applying the ‘but’-logic for the emotion. Our rules in this case are:

- ‘negative emotion’ but ‘positive emotion’, accept ‘positive emotion’
- ‘positive emotion’ but ‘negative emotion’, accept ‘negative emotion’

We also extended this rule to some of the emotion types,

- if ‘fears-confirmed’ or ‘fear’ but ‘satisfaction’ is found, then output ‘relief’
- if ‘hope’ but ‘fears-confirmed’ or ‘fear’ is found, then output ‘disappointment’

- if ‘anger’ but ‘gratification’ or ‘gratitude’ is found, then output ‘gratitude’
- if ‘remorse’ but ‘gratification’ or ‘gratitude’ is found, then output ‘gratitude’
- if ‘gratification’ but ‘anger’ or ‘remorse’ is found, then output ‘anger’
- if ‘gratitude’ but ‘anger’ or ‘remorse’ is found, then output ‘anger’

Hence, applying the above rules, eventually yields ‘happy-for’, ‘relief’ and ‘gratitude’ emotions sensed by the agent/subject (in this case “I” or the author himself) with respect to the object(s) of the given example sentence. In the same evaluation process, for the sentence *“I suddenly got to know that my paper won the best paper award.”*, the emotions are ‘gratification’ and ‘surprise’. The sentence *“She failed to pass the entrance examination.”*, outputs ‘anger’ and ‘disappointment’ emotions. For the sentences like (1) *“I saw that Mary had a great experience to ride on the roller coaster.”*, and (2) *“John noticed that Mary could not ride the roller coaster.”*, the system recognizes “happy-for” and “sorry-for” emotions, respectively. The recognized emotions are anchored to the author of the sentence (in (1)), and to John (in (2)), because the value of the direction of emotion (*de*) variable is set to “other” in these cases.

5.5 Evaluation and Discussion

Currently, our system is able to perform sentence level affect sensing. By implementing the OCC model, our system is the first system capable of sensing a broad range of emotions from the text. A system similar to ours is the *EmpathyBuddy* described in [Liu et al. 2003]. That system is also based on a commonsense knowledgebase (i.e., ConceptNet), and it seems to be the best performing system for sentence-level affect sensing that sense happy, fear, sad, anger, disgust, and surprise emotions. For each sentence it calculates a vector containing the percentage value afferent to each emotion.

By way of example, we provide input and output for *EmpathyBuddy* and our system.

Input: I avoided the accident luckily.

Output of *EmpathyBuddy*: fearful (26%), happy (18%), angry (12%), sad (8%), surprised (7%), disgusted (0%)

Output of ours: sentence valence +11.453; emotions: [gratification, relief, surprise]

We evaluated our system to assess the accuracy of sentence-level affect sensing when compared to human-ranked scores (as "gold standard") for 200 sentences. The sentences were collected from internet based sources for reviews of products, movies, news, and email correspondences. It was given to five human judges to assess each sentence. In the experimental setup we have two systems, Our System and *EmpathyBuddy*. Each judge received the output from both of the systems for each of the sentences from the list of 200 sentences. Upon receiving the outputs a judge could accept either both outputs or anyone of the two or rejected both. For example, for the input sentence *"She is extremely generous, but not very tolerant with people who don't agree with her.",* three judges out of five accepted the output of our system, two accepted the output of *EmpathyBuddy*. Since the majority of the judges accepted the output of our system, vote for this sentence was counted for our system. Similarly for a given sentence if the majority of the judges accepted the both outputs of the two systems, vote for that sentence was counted for both of the systems. In this manner the vote for each sentence is counted and "gold standard" score is prepared. This experimentation yield the following result reported in Table 5.5. Since there has not been done any predefined classifications of the sentences by the human judges, we cannot calculate recall and precision based on the predefined scoring or separated emotion classes.

To measure inter-agreement among judges, we used Fleiss' Kappa statistic. The obtained Kappa value (i.e., κ=0.82) indicates that the overall agreement among the judges for scoring the sentences by the two systems are reliable. Those who collected data from different sources and the judges who scored the sentences were not aware of the architecture and working principle of the systems.

Table 5.5 Preliminary experimental result of the two systems

Data-Set of 200 Sentences				
	Our System	*EmpathyBuddy*	Both	Failed to Sense
Number of Sentences accepted to be correct	41	26	120	13
Total number of Sentences correctly sensed	161	146		
Accuracy	80.5%	73%		

The approach adopted by *EmpathyBuddy* is well-founded because from a given textual description concept(s) are extracted and mapped to already annotated concept(s) where the annotated concept(s) are created by using large-scale real-world knowledge of commonsense. The annotated concepts usually have inherent affective connotation (e.g., happy or sad etc.). An approach that is based on first extracting concepts from text, and then linking concepts to emotional affinity works well when the sentence(s) are semantically simple and descriptive. But it fails for the sentences, such as *"You will hardly get a bad shot with this camera."* Such an approach fails because it does not consider the semantic structure of the sentence.

5.6 Summary

In this chapter, we described how the OCC emotion model can be implemented using linguistic tools for the task of affect recognition from the text. The OCC emotion model explains twenty two emotion types and two cognitive states by interplaying among several cognitive variables. Several heuristics are proposed to assign values to those cognitive variables, and emotion rules based on them are defined in accord with the OCC emotion rules. An initial experimental result of sensing different OCC-defined emotions from the input text obtained an accuracy of 80.5% when compared to the result from human judges as the gold-standard.

Our approach overcomes problems of other similar approaches because we consider the semantic structure of sentences. Our approach is more robust and superior to the commonsense based approach for the following reasons:

- Employs the commonsense knowledgebase to assign words either a negative or positive score,
- Considers the semantic structure of the sentence,
- Applies rules to assign the contextual valence of the so-called concepts (i.e., semantic verb-frame in this case) described in the sentence,
- Senses emotions according to the cognitive theory of emotions having explicit reasons for the particular emotions being detected from the input text.

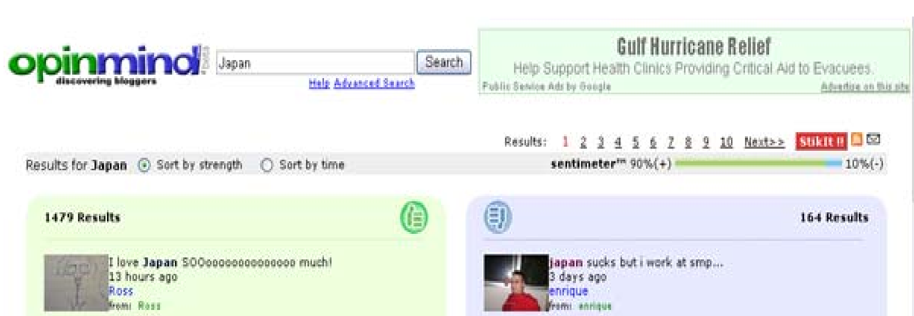
We believe that such a linguistic approach would strengthen human-computer interaction for various applications like developing socially intelligent user interfaces for various applications, including intelligent text processing for informative augmentation of search engine responses to analysis of public opinion trends and customer feedback, socially and emotionally intelligent applications etc. In order to facilitate further testing, we have implemented a web-based user interface so that any user can input a chunk of text and receive outputs from our system and that of other competing systems for emotion recognition from text.

Chapter Six: Developed Applications

Based on the developed theory and approach to sense affective information from text mentioned in the aforementioned chapters several applications have been developed. Summary of the developed applications is given below.

6.1 SenseNet

- It can detect negative, positive or neutral sentiment from the input text.
- The application has the following phases.
  - **Parse:** Connexor Syntactic Parse has been used to obtain Semantic information stored in Triplet(s) format. These triplet(s) are the main computational elements.
  - **Process:** Process the triplets in terms of identifying dependencies between the triplets and assign values to the linguistic components (e.g. verbs etc.)
  - **Assess:** Calculate contextual valence to the phrases using the developed dictionary of our ranked words and rules.
  - **Classify:** Based on the valence input sentence(s) are classified into negative, positive or neutral. An interactive browser shows the result supported with iconic representation of the input text.
- The application works based on the following developed Linguistic Resources:
  - Scored list of named-entities. Following is an example of retrieving sentiment of a named-entity (i.e., Japan) as explained in chapter three.



- Excerpts from the database of scored verb, adjective, adverb and noun are given as following for an idea.

Table 6.1 Excerpts from database of ranked words

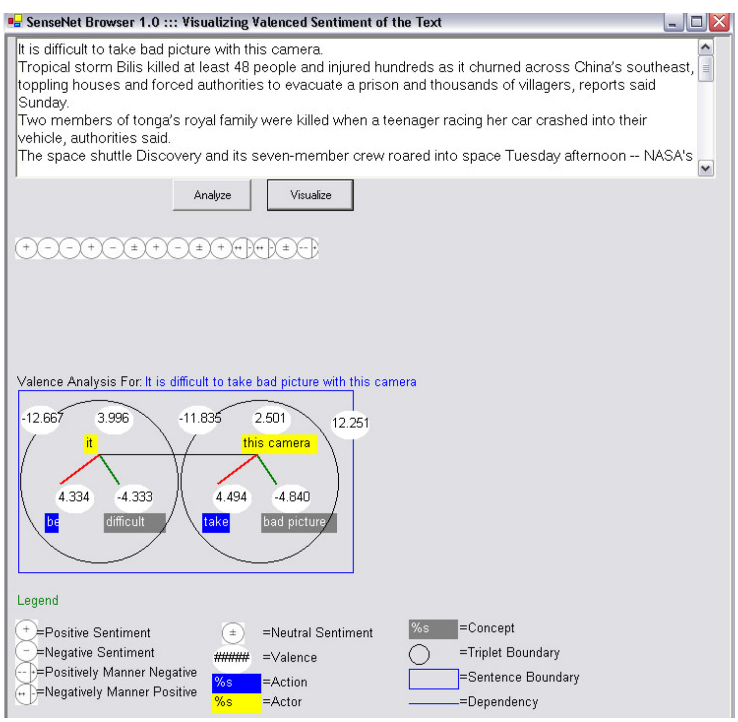
Excerpt from verb database				
rob	-5	-1.25	-3.125	
invent	3.25	2.5	1.875	
testify	3.74481161457	4.17273393522	3.96387481571	
catch	3.620689655	4.094827586	3.857758621	
coach	5	5	5	
sleep	5	5	5	
go	3.916666667	4.375	4.145833333	
follow	4.190705128	4.439102564	4.314903846	
be among	0.0	0.0	0.0	
hate	-5	-1.25	-3.125	affective
milk	3.333333333	3.75	3.541666667	
consider	4.444444444	4.722222222	4.583333333	
injure	-3.63378538881	-1.00975998854	2.24928659281	
hold	3.819444444	4.201388889	4.010416667	
depend	3.75	4.375	4.0625	
calculate	4.583333333	4.791666667	4.6875	
uphold	3.80327745384	4.15892249757	3.914029276	
handle	4.226190476	4.404761905	4.31547619	
unpack	5	5	5	
flash	3.125	3.90625	3.515625	
scold	-5	-1.25	-3.125	

Excerpt from adjective database			
infatuation	3.333333333	appreciative	5
cooperative	5	concerned	3.75
dainty	5	cushy	5
galvanized	5	young	2
abundant	5	sparkling	2.5
hate	-5	humored	5
mellow	4.5	unruffled	5
devoted	5	fervor	3.75
dreadful	-5	dirty	-5
unfeeling	-2.5	sorry	-4.083333333
uneven	-0.833333333	pleasing	5

Excerpt from adverb database	Excerpt fro noun database
angrily -5	pane train 0
pessimistically -5	magic news donovan 0
inquisitively 5	pride 3.62029999975
fearfully -5	hormone 2.63960452366
doubtfully -5	risk 3.749241148
rarely -5 <except>	rise 3.85775
praiseworthily 5	jack 3.65974796037
tightly 5	politician -3.487
next 3.33333333333	school 4.3356
wonderfully 5	investigator 5.0
disgustingly -5	wednesday 3.51183333322
more 5	pincu 0
malevolently -5	bernanke 0
exuberantly 5	force 0.66859977226
expansively 5	mcclaren 0
sharply 5	gator 0
gladly 5	human experience 1.335
	panda 3.13690476205
	asia 4.4213458334
	spokesman 2.95512628056
	surprise there 0

Following is a screen-shot of the GUI of the application. It shows the sentiment analysis for the input sentence *"It is difficult to take bad photo with this camera."*

Figure 6.1 SenseNet GUI

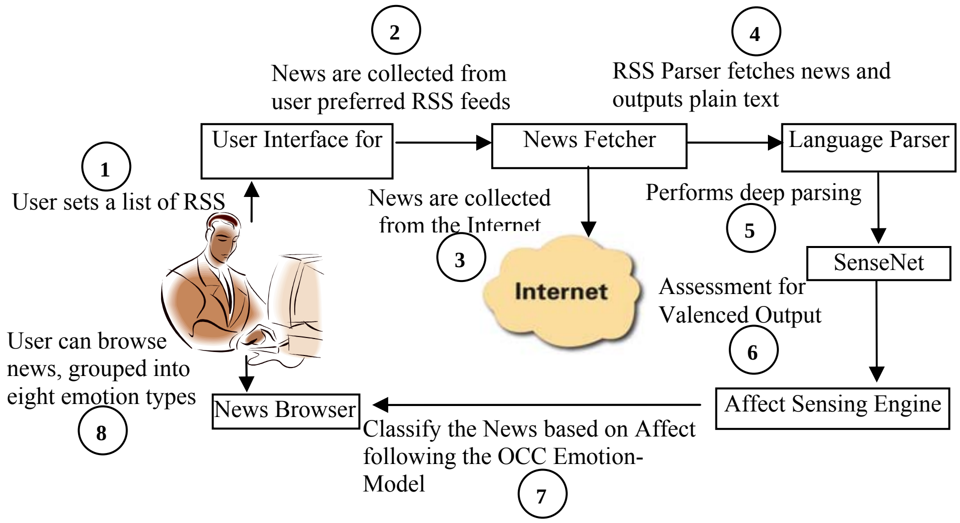


6.2 ASNA: Affect Sensitive News Agent

The system Affect Sensitive News Agent (ASNA) is developed as a news aggregator (i.e., a news-browser) that fetches news employing several RSS news-feeds and auto-categorizes the news according to eight emotion-types plus a neutral category for quicker and intuitive understanding of news topic. The primary goal in developing the system is to demonstrate the feasibility of categorizing news stories according to their emotional affinity using natural language processing techniques. The classification and synthesized retrieval of the large amount of news articles from the Web has been a topic attracting much research effort (e.g., [Jacobs 1992], [Maria and Silva 2001], [Bacan et al. 2005], [Antonellis et al. 2006]) but none has ever considered to sense affective information from news-texts for grouping those on the basis of affective senses and largest drawback of these systems are that they are all based on static corpora of published news articles. We have followed a deep approach to synthesize news-text and classified those according to the concept of emotion types.

ASNA Architecture: First a user chooses the sources of news according to his/her domain of interest. In this case we used RSS feeds as the sources for the news. After the news sources are selected, News Fetcher collects the news as tuples of news topic and brief story corresponding to the topic by parsing the results returned by the RSS feeds. Then the plain-text tuples are parsed by a language parser. We have implemented a deep parsing technique to output tuple(s) of Subject, Subject Type, Subject Attributes; Action, Action Status, Action Attributes; and Object, Object Type, Object Attributes for each line of text. The output of language parser is assessed by a linguistic tool SenseNet mentioned before. SenseNet considers each tuple as a Sense and outputs a numerical value for each lexical-unit (e.g. sentence). Affect Sensing Engine then classifies the news-texts according to eight emotion-types namely, Happy, Sad, Hopeful, Fearful, Admirable, Shameful, Loveable, and Hatred plus a Neutral category. Finally a user can browse the news according to the emotion groups.

Figure 6.2 Architecture of ASNA



News Browser: The news browser finally enlists the news according to emotion-types and a user can browse news thereby. Figure 6.3 shows a snap-shot of the emotion sensitive news browser having 9 buttons. Clicking on any button shows a list of news summary corresponding to a specific emotion-affinity. An avatar reads out the news summary and a user can also view the full story of the news on this browser by clicking either on the headline or the image associated with the news.

Figure 6.3 ASNA News Browser

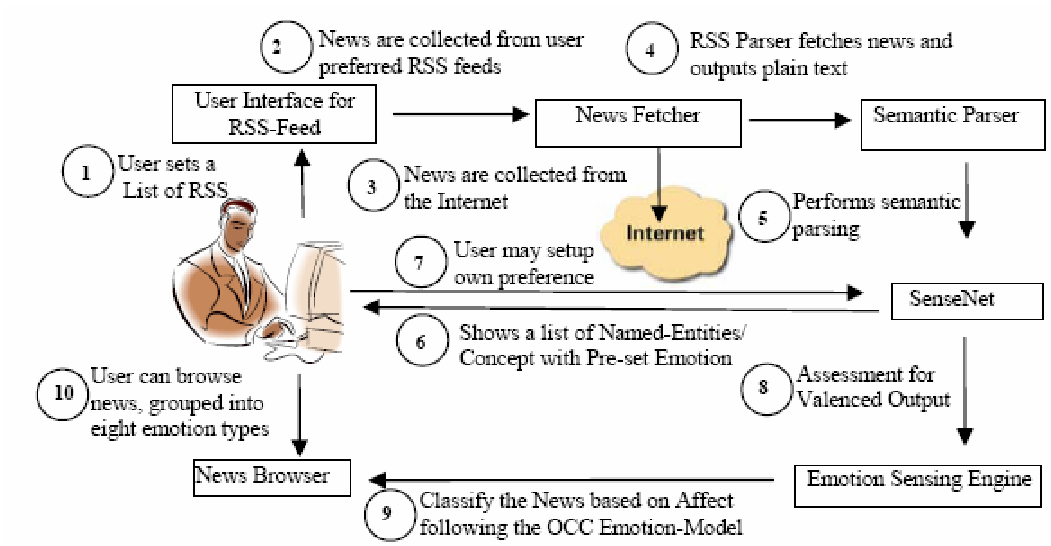
6.3 ESNA: Emotion Sensitive News Agent

ESNA system, which is an extension of ASNA, followed a pipelined architecture with the following stages: Parse, Process, Solicit, Assess and Classify.

- The *Parse* stage implements a deep parsing technique to extract different linguistic components (e.g. actor, verb, object etc.) and their relationships within the input sentence(s).
- The *Process* stage assigns contextual valence to the linguistic components (e.g. verb or object) by employing the linguistic resource, the SenseNet. This stage also creates a list of named entities detected from the news-text.
- The *Solicit* stage presents the list of named entities with preset emotional attitude to the user, and provides the user the option to reset his/her personal feeling towards those. The 'Solicit' phase was not integrated to the ASNA system.
- The *Assess* stage assigns values to the variables underlying the emotion rules by assessing the values obtained from Process stage and consulting user preferences for certain named entities.
- The Classify stage implements the rules to realize a linguistic version of the OCC emotion model for emotion analysis. In ESNA we only considered a subset of eight emotional categories like ASNA system. According to a rule, e.g. emotion-type 'Happy', the input "*Italy claim world cup triumph.*" would be classified as a 'happy' news by ASNA system, but it might be classified as a 'sad' news if someone has already set one's 'negative' or 'dislike' preference towards "Italy" in the genre of sports in ESNA system.

ESNA Architecture: ESNA has ten operational steps. First the user chooses the sources of news according to his/her interest. After the news sources are selected, News Fetcher collects the news as tuples by parsing the results returned by the RSS feeds. Each tuple contains the news-headline and a brief story corresponding to the headline. Then the text tuples are parsed by a Semantic Parser. We have implemented a semantic parsing technique that performs dependency analysis on the words and outputs triplet(s) of subject, verb, and object according to each semantic verb-frame of the input sentence(s). The output of semantic parser is assessed by a linguistic tool called SenseNet that we have developed. SenseNet offers the user a list of named entities that are obtained from the news item and also assigns a prior emotion towards each named entity. A user can change the prior emotion and may setup his/her feeling towards that entity. SenseNet outputs a numerical value for each lexical-unit (e.g. sentence) and also assign values to the cognitive variables that deal with the rules for the emotions. Emotion Sensing Engine has implemented rules for the eight OCC-emotion types and these rules are evaluated to classify the news according to the eight emotion types.

Figure 6.4 Architecture of Emotion Sensitive News Agent (ESNA)



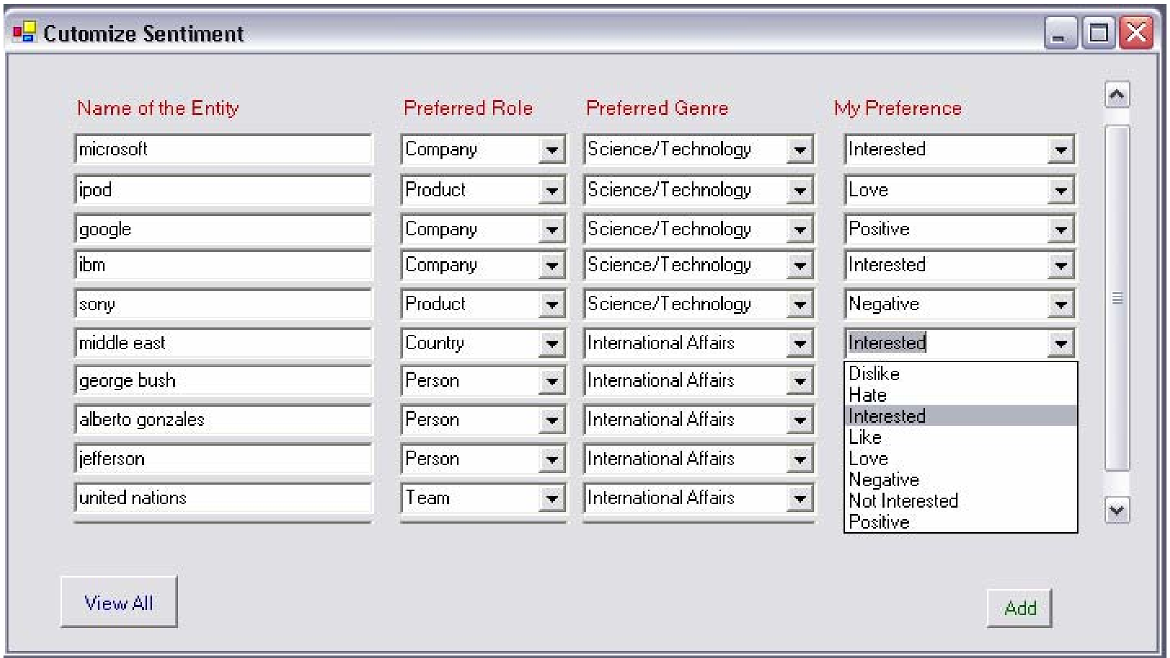
User-Centric Emotion Recognition: The system maintains a list of scored named entities. The information of an entity is stored as the following format: Named-entity [Role, Concept, Genre, General-Sentiment]. The field 'Role' indicates any of the values from the list: {Company, Concept, Country, Object, Other, Person, Product, Service, Team} and 'Concept' represents a ConceptNet keyword to represent the concept of the entity. 'Genre' indicates any of the 15 genres (e.g. Politics, Sports, Technology etc.) taken from the news domain. 'General-Sentiment' indicates any of the value from the list {Dislike; Hate; Interested; Like; Love; Negative; Not-Interested; Positive} to indicate pre-set emotion towards the named entity. We did not use any named entity recognizer to identify a named entity, and make the simplifying assumption that anything for which ConceptNet fails to assign valence is a named entity. To assign General-Sentiment we have developed a tool that can extract sentiment from Opinmind. The algorithm to assign a negative or positive value to indicate the above enumerated values is:

```
If (positive vote > negative vote) then
Begin
  If  (0% ≤ positive vote ≤ 30%)
    Pre-set emotion = 1 //indicates 'Positive' feeling
  Else if (30% < positive vote ≤ 60%)
    Pre-set emotion = 2 //indicates 'Interested'
  Else if (60% < positive vote ≤ 80%)
    Pre-set emotion = 3 //indicates 'Like'
  Else if (80% < positive vote ≤ 100%)
    Pre-set emotion = 4 //indicates 'Love'
  End
```

Similarly if the negative vote is greater than the positive vote, the value like -1; -2; -3; or -4 is assigned to set negative pre-set emotion towards the input named-entity. The range of the values to decide pre-set emotion has been taken heuristically.

Initially a list of 2000 entries is manually created and scored using Opinmind. Usually the value of 'General Sentiment' is idiosyncratic and arguable, and hence these values are shown to the user with the corresponding pre-set emotions. Figure 6.5 shows the interface where a user can add his or her particular sentiment towards a specific entity. In the text-box the name of the entity is displayed or typed, the preferred role of the entity is selected from any of the values like, Company; Concept; Country; Object; Other; Person; Product; Service; or Team that the dropdown list shows. The preferred genre of the entity is selected from the 15 genres. Finally the sentiment towards that entity can be chosen from the eight pre-set emotions discussed before.

Figure 6.5 User Interface to Customize Sentiment towards Named-Entities



User-defined sentiments towards the named-entities are stored against that specific user's profile as profile files. When a user is identified with the system, the respective profile of that user is loaded and system consults that profile to classify the news. Example of two profiles are given below.

Profile for Username: *mamshaikh*

Name entity	**Preferred role**	**Preferred Genre**	**My Sentiment**
Microsoft	Company	Science/Technology	Positive
Play-station	Product	Science/Technology	Interested
iPod	Product	Science/Technology	Like
Bin Laden	Person	War/Terrorism	Hate
Brazil	Team	Sports	Love
Tiger Woods	Person	Sports	Interested
Maria Sharapova	Person	Sports	Positive

Hizbullah	Team	International Affairs	Negative
David Beckham	Person	Sports	Love
Sri Lanka	Team	Sports	Negative
Panda	Other	Animal/Nature	Interested
Japan	Country	International Affairs	Positive

Profile for Username: *russell*

Name entity	**Preferred role**	**Preferred Genre**	**My Sentiment**
Microsoft	Company	Science/Technology	Negative
Play-station	Product	Science/Technology	Interested
iPod	Product	Science/Technology	Not Interested
Bin Laden	Person	War/Terrorism	Hate
Brazil	Team	Sports	Negative
Tiger Woods	Person	Sports	Love
Maria Sharapova	Person	Sports	Negative
Hizbullah	Team	International Affairs	Positive
David Beckham	Person	Sports	Dislike
Sri Lanka	Team	Sports	Like
Panda	Other	Animal/Nature	Interested
Japan	Country	International Affairs	Positive

In the given example profiles above for the named-entity 'Microsoft', the user '*mamshaikh'* has setup a 'positive' sentiment but *'russell'* has setup a 'negative' sentiment. Hence for the news like, *"Microsoft released new test versions of Windows Server 2008 and the Windows Vista Service Pack, two highly anticipated technologies."* is listed as 'happy' news for the user *'mamshaikh'* but for *'russell'* it is listed as 'sad' news.

Algorithm. The core algorithm underlying the ESNA system can be summarized as following:

Input: $P=\{S_1, S_2, ... S_n\}$ // a set of sentences. Each *P* indicates a news story.
Output: E // indicates the emotion detected from *P*
Pseudo Code for Processing:

```
Procedure getNewsEmotion (P)
Begin
  emotionSet ={} //null set
  for each Si in P do //assume 1 ≤ i ≤ n
     tripletSeti = getSemanticParsing (Si) //output of Parser is a set of Triplets for each sentence
      valencedReaction = getSentimentFromSenseNet(tripletSeti) // returns a value between ± 15
      reaction = getSelfReactionOfEvent (tripletSeti)  // returns "pleased" or "displeased"
      presumption = getSelfPresumptionOfEvent (tripletSeti) // returns "desirable" or undesirable"
      prospect = getProspectOfEvent (tripletSeti) // returns "positive" or "negative"
      appraisal = getSelfApprisalOfEvent(tripletSeti) // returns "praiseworthy" or "blameworthy"
      eventStatus = getEventStatus (tripletSeti) // returns "present" or "past" or "future"
      objectAppealing=getAppealingnessOfEntity(tripletSeti)//returns "attractive" or "unattractive"
      objectFondness = getFondnessOfEntity(tripletSeti) //returns "liking" or "disliking"
      presetEmotion =getPresentEmotionOfEntity(tripletSeti) // returns a value between ± 4
      emotionOfTheSentence = getSentenceEmotion (reaction, presumption, prospect, appraisal,
eventStatus, objectAppealing, objectFondness, presetEmotion)
       //return an emotion for the sentence.
      emotionSet = emotionSet  ∪ {emotionOfTheSentence}
loop until all sentences are processed
newsEmotion = pickBestEmotion (emotionSet)  //get the highest emotion from the set
return newsEmotion
End Procedure
```

In order to explain the idea how user-centric emotion classification is achieved, we present an example rule that is used in the function *getSentenceEmotion()* to decide "happy" or "sad".

- IF valencedReaction > 5.0 and reaction = "pleased" and presumption ="desirable" and presetEmotion >0 THEN sentenceEmotion= sentenceEmotion ∪ {"happy"}
- ELSE IF valencedReaction < -5.0 and reaction = "displeased" and presumption ="undesirable" and presetEmotion < 0 THEN sentenceEmotion= sentenceEmotion ∪ {"sad"}
- ELSE IF valencedReaction > 5.0 and reaction = "pleased" and presumption ="desirable" and presetEmotion < 0 THEN sentenceEmotion= sentenceEmotion ∪ {"sad"}// a positive event may be classified as "sad" based on the user's preference.
- ELSE IF valencedReaction < -5.0 and reaction = "displeased" and presumption ="undesirable" and presetEmotion > 0 THEN sentenceEmotion= sentenceEmotion ∪ {"happy"} // a negative event may be classified as "happy" based on the user's preference.

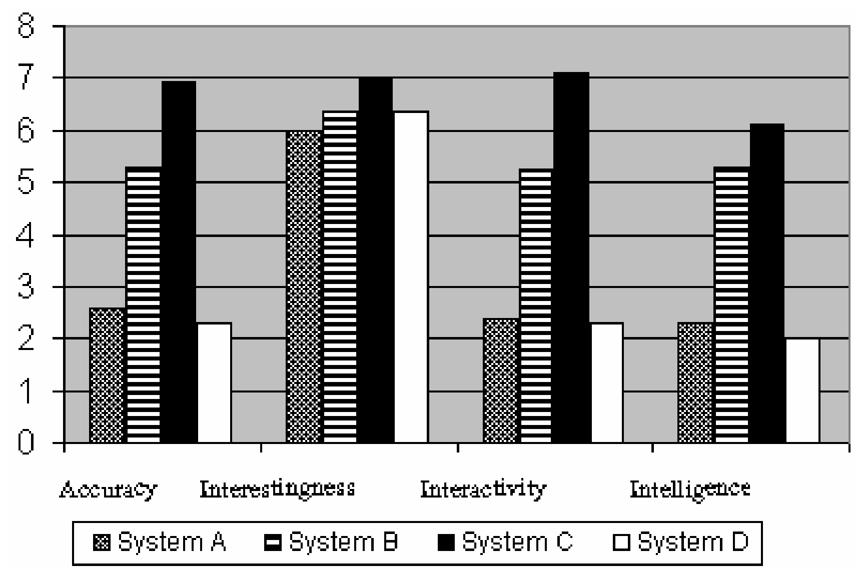
In the example above, we assume that in order to decide a sentence having affective strength the SenseNet output (i.e., valencedReaction) for the sentence should be either grater than 5 or less than -5. The value for the cognitive variable *self_reaction (sr)* is assessed by the function getSelfReactionOfEvent as “Pleased” or “Displeased” if the valence of the concerned event is assessed either positive or negative by considering the scores of verbs (*V* or *AV*) and concepts (*CON*) stored in the knowledgebase.

Preliminary Evaluation: We conducted a small user study with 7 participants (4 females, 3 males; all are university students) to quantitatively measure the performance of the system. In order to do this we developed 4 versions of the system with exactly the same user-interfaces but varying functionalities. For a list of fetched news:

- System A categorizes the news randomly,
- System B categorizes without consulting user-preferential information,
- System C categorizes consulting user-preferential information, and
- System D enlists user-preference reversely (e.g. if someone has set “Love” as a preference towards “David Beckham”, system considered it as “Hate”).

The participants in the study (using within-subject design) were not informed about the different versions of the system, but they were told that all the four systems do the same things in different ways. Each user interacted with the four systems for 7 days and each time they could select the news sources as well as setup preferences towards certain entities according to their choices. For each day at a particular time (e.g. in morning) a person was given one of the four systems (e.g. System B); in the same day at another time (e.g. before noon) the same person was given another system (e.g. System D) and in the same manner the other systems were assigned to the same user. After every session everyone filled a survey form to assign numerical values (0 to 10) according to one’s scoring towards the questions. There were four questions, asking about accuracy of classification, interestingness of the system, interactivity of the system in terms of how obliging the system was to synchronize with the personal preferences in the classification and finally the score for the intelligence of the system. The average score of the user-opinions towards the four characteristics (i.e., Accuracy, Interestingness, Interactivity and Intelligence) of the systems are summarized in Figure 4.

Figure 6.6 the summary of user study



System C (which is our target system) showed the higher score on all the dimensions. System D had the worst scores. At present system C takes several minutes (avg. 240 sec.) to compute an output, mainly because we have to start several underlying systems, such as ConceptNet and SenseNet to load the initial knowledgebase. Therefore, we deliberately inserted delays to the systems A, B and D in order to achieve comparable experimental conditions. Although we performed the comparative usability study of the system over a small-sized group, we believe that the overall assessment of the targeted system would not be much different from the obtained result because the data is normal for each system with respect to the daily scores for those four characteristics given by each individual. Moreover the f-ratio of two-way between groups

ANOVA for the interestingness of the System A (i.e., with random classification) and System C (i.e., target system) is not significant; similarly the f-ratio for intelligence between System B and System C is also not significant. But the f-ratios for interactivity and intelligence are significant between System C and System D which again re-established the theory that the people most naturally interact with their computers in a social and affectively meaningful way [Reeves and Nass, 1998].

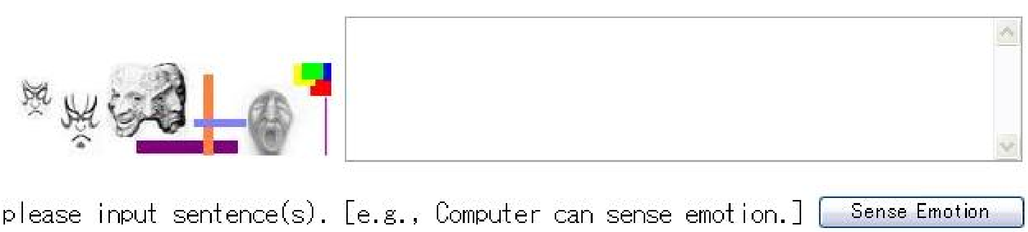
We conclude that the system ESNA is interesting, interactive, intelligent and accurate to some extent. In the area of personalized affective news, we have usually found two types of systems: one type classifies news according to taxonomical categories, and the other one considers news topics as story events to assess sentiment (positive or negative or neutral) and limited emotional reactions (suspense or curiosity). But none of those ever considered classifying news articles into a broad range of emotion categories like we do. So the ESNA system would definitely help news readers to grasp news articles in an interesting and more personalized manner.

6.4 Online System for Textual Affect Sensing

We have deployed the system as a web application so that anyone can input sentence(s) and test our system. The system is running at this location: http://masum.testita.com/emotest

The user interface of the system is given in Figure 6.7

Figure 6.7 user-interface of the online system



When a text is input to the given textbox and a user clicks on the “Sense Emotion” button, the text is posted to the two systems to process. The first system (i.e., System 1) is the system called as *Empathybuddy* mentioned in [Liu et al. 2003] and the second system (i.e., System 2) is ours. The output enlists the outputs obtained from both systems. For example, for a given news text, *"An earthquake measuring 6.0 on the Richter scale shook the northern part of Indonesia's Sulawesi island on Tuesday, but there were no immediate reports of damage or casualties, the meteorology agency said."*, the output obtained is given in Figure 6.8.

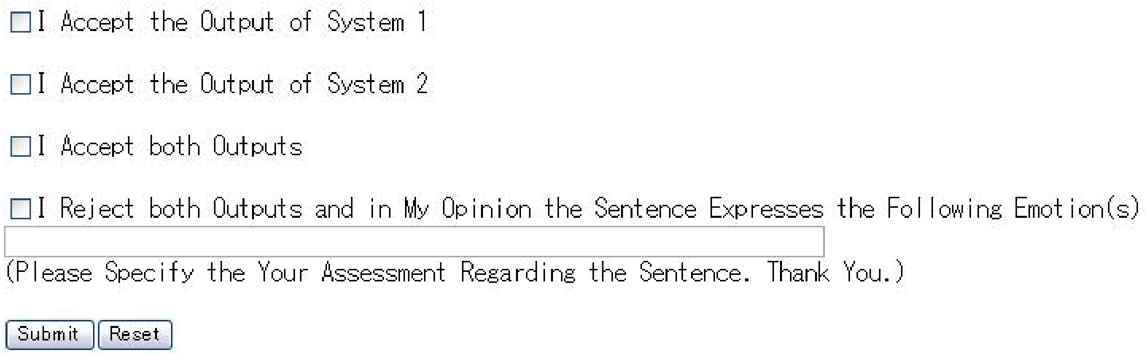
Figure 6.8 the output interface of the online system

Input Sentence: **An earthquake measuring 6.0 on the Richter scale shook the northern part of Indonesia's Sulawesi island on Tuesday, but there were no immediate reports of damage or casualties, the meteorology agency said.**	
Output of System 1	Output of System 2
angry:36.39 **fearful**:28.42 **surprised**:18.17 **sad**:11.28 **happy**:0.00 **disgusted**:0.00 The Sentence Primarily Expresses **angry** Emotion.	**distress**:(60.04%) **disappointment**:(30.02%)

The characteristics and output format of both systems are discussed in chapter five. Basically the output of System 1 always shows the score of five basic emotions (i.e., knows as Ekman-emotion) for each sentence. According to the first output it means two things. Firstly, the sentence carries 'angry', 'fearful', 'surprise' and 'sad' emotions and secondly, one can consider that the primary emotion of the sentence is 'anger'. Since the System 2 deals with different sets of named emotions (i.e., the emotions define by the OCC emotion theory), the name of the emotions are different from that of System 1 but System 2 also have the similar types of emotion tags like System 1. For example, "sad" of System 1 is similar to "distress" for System 2. The output of System 2 indicates all the emotions that are likely to be expressed by the input text and the percentage value indicates the strength of that emotion. Thus the output of System 2 indicates that the sentence is expressing strong distress and then less amount of disappointment.

Upon receiving the output a user is asked to participate for the following survey. The survey from is given as Figure 6.9.

Figure 6.9 Survey form



☐ I Accept the Output of System 1

☐ I Accept the Output of System 2

☐ I Accept both Outputs

☐ I Reject both Outputs and in My Opinion the Sentence Expresses the Following Emotion(s).

(Please Specify the Your Assessment Regarding the Sentence. Thank You.)

Submit Reset

In this survey process we are collecting the users' testing report for the systems for the purposes like, comparing the performance, analyzing errors of the systems. This online experimentation is also helping to increase the database of the System 2 in terms of ranking new words.

Chapter Seven: Summary and Conclusion

The "Affective Computing" discipline developed several mechanisms for emotion sensing, including the processing of various physiological signals obtained from wearable sensors. This research adds an original contribution in the domain of affective computing originating in Natural Language Processing (NLP) for the task of affect sensing from text. The approaches and developed systems described in this research propose a novel method to recognize sentiment as well as emotion at the sentence level. The system first performs semantic processing and then applies rules to assign contextual valence to the linguistic components in order to obtain sentence-level sentiment valence. The system is well-founded because we have 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 when compared to other approaches. The summary of our experimental result is given in Table 7.1.

Table 7.1 the summary of experimental results using different datasets

<table>
<tr><td>Datasets</td><td colspan="3">Our Approach</td><td colspan="2">Other Best Performed Approach</td></tr>
<tr><td></td><td>Accuracy (%)</td><td>Recall (%)</td><td>F-Score (%)</td><td colspan="2">Accuracy (%)</td></tr>
<tr><td>Dataset A</td><td>82</td><td>R1=85.56
R2=79.31
R3=78.26</td><td>Fp=89.02
Fn=80.70
Fnu=64.29</td><td colspan="2">Our created dataset</td></tr>
<tr><td>Dataset B</td><td>91.53</td><td>R1=90.62
R2=92.44</td><td>Fp=91.34
Fn=91.72</td><td colspan="2">84.09</td></tr>
<tr><td rowspan="3">Dataset C</td><td rowspan="3">85.5</td><td rowspan="3">R1=79.7
R2=91.3</td><td rowspan="3">Fp=83.54
Fn=87.28</td><td>Machine Learning 1</td><td>90.4</td></tr>
<tr><td>Machine Learning 2</td><td>86.0</td></tr>
<tr><td>Non Mac. Learning</td><td>66.7</td></tr>
<tr><td>Dataset D</td><td>81.75</td><td>81.97</td><td>81.76</td><td colspan="2">Our created dataset</td></tr>
</table>

In the Table 7.1, recall values R1, R2, and R3 mean the recall percentages for the classes of positive, negative, and neutral sentences respectively. Similarly the F-core values Fp, Fn, and Fnu mean the F-Score percentages for the classes of positive, negative, and neutral classes respectively. Since Dataset A and Dataset D are our personal datasets, we don't have empirical study report on those datasets by others. All the datasets are considered as "Gold-Standard" because the texts inside the datasets are ranked by human judges. Our approach outperformed the machine learning approach for Dataset C. This we consider as the primary success of our approach because this dataset contains subjective or opinion bearing sentence(s) which are pre-annotated as either negative or positive sentences. This is the standard dataset for the task of "sentence-level" sentiment analysis. Hence experimental result obtained for Dataset B establishes the idea that our approach is better than machine learning approach for relatively smaller amount of input text where there subjective sentences. On the other hand, Dataset C contains both subjective and objective sentences taken from movie reviews. From the empirical results reported in other studies, our approach could not outperform the machine learning approaches. The reason is, our approach considers each of the sentences in a given paragraph subjectively and assign scores for each of them. Thus for a negatively reviewed movie, due to having more positive sentences to describe a positive plot of the movie obtains an overall positive score for the paragraph from our system. Since machine learning approaches adopt some patterns and clues to separate the subjective sentences from the objective sentences, their performance is better due to the mechanism of exclusion of such sentences. Hence applying the algorithm to separate the subjective sentences from the objective sentences of a given paragraph our system can be improved to deal with such cases.

Our primary objective is to sense emotion from a given sentence and towards this approach machine learning algorithms cannot perform well due to lack of domain independent training datasets and failure to adopt semantic patterns (e.g., a negative actor connected with a positive action or two interrelated negative senses etc.) of the input text while assigning a score to the text. Therefore, we have implemented a rule based system that can overcome those two limitations. Our system, *SenseNet*, applies a numerical-valence based analysis on the input text. The main idea of *SenseNet* is to form the computational model of the input text by considering the semantic verb-frames as the triplets of subject-verb-object (SVO); assign numerical value to each lexical unit based on their lexical sense affinity; assess the value of the sense(s) (i.e., the SVOs), and then finally output sense-valence for each lexical-unit (e.g., words, sentences). The scoring of the words is performed by applying commonsense knowledge and internet based resources. There are many applications to process texts where there are relatively more subjective or opinionated sentences in the input text (e.g., blogs, news, email, product review etc.) and hence for those applications our approach can be applied with higher reliability and acceptability. For example, one of the application areas could be affective content searching. Provided that we have a corpus of texts (e.g., blogs) and then using our approach we can automatically annotate each sentence in terms of 'positive' or 'negative' sentiment and furthermore to named emotions like OCC-emotion model defined emotions. Then a search query like, "list all the 'happy' sentences" or "show all 'angry' opinions" can be performed. This type of information might be helpful to the market researchers to know about different affective attitudes expressed by people in blogs regarding products or services.

Another aim of this research is to give computer programs a skill known as "emotional intelligence" with the ability to understand human emotion and to respond to it appropriately. We have extended the positive or negative sentiment recognition system into a full-fledged emotion recognition system, which may classify named emotions rather than positive or negative sentiments. We have followed the OCC emotion-model [Ortony et al. 1988] by applying different linguistic tools and heuristics to sense a rich set of affective information from the text. We also took into account user-specific preferences (e.g. personal opinions about particular entities) that are processed by the system to analyze subjective statements in a personalized manner. Such approach of affect sensing is tested by developing affective news browsers namely, ASNA and ESNA.

References

[Antonellis et al. 2006] Antonellis, I., Bouras, C., and Poulopoulos, V.: Personalized news categorization through scalable text classification. Proceedings of the 8th Asia-Pacific Web Conference, pp. 391-401, Harbin, China, 2006

[Bacan et al. 2005] Bacan, H., Pandzic, I. S., and Gulija, D.: Automated News Item Categorization. Proceedings of the 19th Annual Conference of the Japanese Society for Artificial Intelligence, pp. 251-256, Springer-Verlag, Kitakyushu, Japan, 2005

[Bartneck 2003] Bartneck, C.: Interacting with and embodied emotional Character, Proceedings of Designing Pleasurable Products and Interfaces (DPPI-03), pp. 55-60, Pittsburgh, 2003

[Benamara et al. 2006] Benamara, F., Cesarano, C., Picariello, A., Reforgiato, D., and Subrahmanian, V.S.: Sentiment Analysis: Adjectives and Adverbs are better than Adjectives Alone. Proceedings of International Conference on Weblogs and Social Media (ICWSM 07), pp. 203-206, Seattle, Washington, 2007

[Chengwei and Gencai 2001] Chengwei, Y., and Gencai, C.: An emotion development agent model based on OCC model and operant conditioning. Proceedings of International Conference on Info-tech & Info-net (ICII-01), pp. 246-250, Beijing, China, 2001

[Dyer 1987] Dyer, M.G.: Emotions and Their Computations: Three Computer Models. Cognition and Emotion, Vol. 1, No. 3, 323-347, 1987

[Elliott 1992] Elliott, C.: The Affective Reasoner: A Process Model of Emotions in a Multi-agent System. PhD thesis, The Institute for the Learning Sciences, Technical Report No. 32, Northwestern University, May 1992

[English Club] English Vocabulary at URL: http://www.englishclub.com/vocabulary

[Eriksson 2006] Eriksson, B.: Sentiment classification of movie reviews using linguistic parsing. Available online at this URL: http://pages.cs.wisc.edu/~apirak/cs/cs838/eriksson_final.pdf.

[Esuli and Sebastiani 2005] Esuli, A. and Sebastiani, F.: Determining the semantic orientation of terms through gloss analysis. Proceedings of the ACM International Conference on Information and Knowledge Management, (CIKM-05), pp. 617-624, Bremen, Germany, 2005

[Fellbaum 1999] Fellbaum, C. (ed.): WordNet: An Electronic Lexical Databases, MIT Press, Cambridge, Massachusetts, 1999

[Fitrianie and Rothkrantz 2006] Fitrianie, S. and Rothkrantz, Leon J.M.: Constructing Knowledge for Automated Text-Based Emotion Expressions, Proceedings of CompSysTech, Tarnovo, Bulgaria, 2006

[Fleiss 1971] Fleiss, J. L.: Measuring nominal scale agreement among many raters, Psychological Bulletin. 76(5):378-382, 1971

[Goertzel et al. 2000] Goertzel, B., Silverman, K., Hartley, C., Bugaj, S., and Ross, M.: The Baby Webmind Project. Proceedings of Artificial Intelligence and the Simulation of Behaviour (AISB-2000), Birmingham, England, 2000

[Grefenstette et al. 2004] Grefenstette, G., Qu, Y. Evans, D., and Shanahan, J.: Validating the Coverage of Lexical Resources for Affect Analysis and Automatically Classifying New Words along Semantic Axes, Computing Attitude and Affect in Text: Theory and Applications, In eds. J. Shanahan, Y. Qu, and J. Wiebe, 93-107. The Information Retrieval Series Vol.20, Netherlands: Springer Verlag, 2004

[Hatzivassiloglou and McKeown 1997] Hatzivassiloglou, V. and McKeown, K. R.: Predicting the Semantic Orientation of Adjectives. Proceedings of the 35th annual meeting on Association for Computational Linguistics (ACL), pp.174-181, Madrid, Spain, 1997

[Hatzivassiloglou and McKeown 2002] Hatzivassiloglou, V. and McKeown, K. R.: Predicting the semantic orientation of adjectives. Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and the 8th Conference of the European Chapter of the ACL, pp. 174-181, Madrid, Spain, 2002

[Hatzivassiloglou and Wiebe 2000] Hatzivassiloglou, V. and Wiebe, J.M.: Effects of adjective orientation and gradability on sentence subjectivity. Proceedings of the 18th International Conference on Computational Linguistics (COLING 2000), pp. 299-305, Saarbrücken, Germany, 2000

[Heise 2007] Heise, D.R.: Expressive Order Confirming Sentiments in Social Actions, New York: Springer, 2007

[Hu and Liu 2004] Hu, M. and Liu, B.: Mining and summarizing customer reviews. Proceedings of the International conference on knowledge discovery and data mining (KDD-04), pp. 168-177, Seattle, WA, 2004

[Jacobs 1992] Jacobs, P.: Joining statistics with NLP for text categorization. Proceedings of the Third Conference on Applied Natural Language Processing, Association for Computational Linguistics (ACL), pp. 178-185, Morristown, NJ, 1992

[Johnson et al. 2006] Johnson, C. R., Fillmore, C. J., Petruck, M. R. L., Baker, C. F., Ellsworth, M., Ruppenhofer, J., and Wood, E. J., eds., FrameNet II: Theory and Practice. Available online at URL: http://framenet.icsi.berkeley.edu/book/book.pdf, Accessed May 2006.

[Kamps and Marx 2002] Kamps, J. and Marx, M.: Words with Attitude. Proceedings of the 1st International Conference of Global WordNet, Mysore, India, 2002

[Kennedy and Inkpen 2006] Kennedy, A. and Inkpen, D.: Sentiment Classification of Movie Reviews using Contextual Valence Shifter, Computational Intelligence, 22(2):110–125, 2006

[Kim and Hovy 2005] Kim, S. M., and Hovy, E.: Automatic Detection of Opinion Bearing Words and Sentences. Proceedings of the International Joint Conference on Natural Language Processing, (IJCNLP-05), pp 61-66, Jeju Island, Korea, 2005

[Kim and Hovy 2006] Kim, S. M. and Hovy, E. H.: Identifying and Analyzing Judgment Opinions. Proceedings of the Human Language Technology-North American chapter of the Association for Computational Linguistics annual meeting (HLT-NAACL 06), pp. 200-207, New York, 2006

[Knobloch et al. 2004] Knobloch, S., Patzig, G., Mende, A., and Hastall, M.: Affective News-Effects of Discourse Structure in Narratives on Suspense, Curiosity, and Enjoyment While Reading News and Novels. Communication Research, 31(3):259-287, 2004

[Koppel and Shtrimberg 2004] Koppel, M. and Shtrimberg, I.: Good News or Bad News? Let the Market Decide. Proceedings of the AAAI Spring Symposium on Exploring Attitude and Affect in Text, pp. 86-88, Palo Alto, CA, 2004

[Liu and Singh 2004] Liu, H. and Singh, P.: ConceptNet: A Practical Commonsense Reasoning Toolkit, BT Technology Journal 22(4), 211-226. Kluwer Academic Publishers, 2004

[Liu et al. 2003] Liu, H., Lieberman, H. and Selker, T.: A Model of Textual Affect Sensing using Real-World Knowledge. Proceedings of the International conference on Intelligent User Interface (IUI-03), pp. 125-132, ACM, Miami, USA,, 2003

[Machinese Syntax] Machinese Syntax, official website, http://www.connexor.com/connexor/

[Mandler 1975] Mandler, G.: Mind and Emotion. New York, NY: Wiley, 1975

[Maria and Silva 2001] Maria, N., and Silva, M. J.: Theme-based Retrieval of Web News. Proceedings of the Third International Workshop on the Web and Databases, pp. 26-37, Springer, Athens, Greece, 2001

[Mihalcea and Liu 2006] Mihalcea, R. and Liu, H.: A corpus-based approach to finding happiness, Computational approaches for analysis of weblogs. Proceedings of the Association for the Advancement of Artificial Intelligence (AAAI) Spring Symposium, Boston, Massachusetts, 2006

[Minsky 2006] Minsky, M.: The Emotion Machine, Simon & Schuster, New York, 2006

[Mr. Bubb in Space 2002] Web-site to download the application http://www.zoesis.com/mrbubb/

[Mullen and Collier 2004] Mullen, T. and Collier, N.: Sentiment analysis using support vector machines with diverse information sources. Proceedings of the International Conference on Empirical Methods in Natural Language Processing (EMNLP-2004), pp. 412-418, Barcelona, Spain, 2004

[My Yahoo!] My Yahoo! URL: http://my.yahoo.com

[Nasukawa and Yi 2003] Nasukawa, T., and Yi, J.: Sentiment Analysis: Capturing Favorability Using Natural Language Processing. Proceedings of the Conference on Knowledge Capture (K-CAP 2003), ACM, pp. 70-77, Florida, 2003

[Opinmind 2006] Opinmind, Discovering Bloggers, http://www.opinmind.com/

[Ortony 2003] Ortony, A.: On making believable emotional agents believable. In R. Trappl, P. Petta & S. Payr (Eds.), Emotions in humans and artifacts. Cambridge, MA.: MIT Press. pp. 189-211, 2003

[Ortony et al. 1988] Ortony, A., Clore, G.L. and Collins, A.: The Cognitive Structure of Emotions, Cambridge University Press, 1988

[Pang and Lee 2004] Pang, B. and Lee, L.: A Sentimental Education: Sentiment Analysis Using Subjectivity Summarization Based on Minimum Cuts. Proceedings of the 42nd Annual Meeting of the Association of Computational Linguistics, (ACL-04), pp. 271-278, Barcelona, Spain, 2004

[Pang and Lee 2005] Pang, B. and Lee, L.: Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. Proceedings of the 43rd Annual Meeting of the Association of Computational Linguistics, (ACL-05), pp. 115-124, Michigan, 2005

[Pang et al. 2002] Pang, B., Lee, L., and Vaithyanathan, S.: Thumbs up? Sentiment Classification using Machine Learning Techniques. Proceedings of the International Conference on Empirical Methods in Natural Language Processing, (EMNLP-02), pp 79-86, Philadelphia, PA, 2002

[Pennebaker et al. 2001] Pennebaker, J. W., Francis, M. E., and Booth, R. J.: Linguistic inquiry and word count: LIWC (2nd ed.) [Computer Program]. Mahwah, NJ: Erlbaum, 2001

[Pennebaker et al. 2003] Pennebaker, J. W., Mehl, M. R., and Niederhoffer, K. G.: Psychological aspects of natural language use: Our words, our selves. Annual Review of Psychology, 54:547-577, 2003

[Picard 1997] Picard, R. W.: Affective Computing. Cambridge, MA: The MIT Press, 1997

[Polanyi and Zaenen 2004] Polanyi, L. and Zaenen, A.: Contextual valence shifters. Computing Attitude and Affect in Text: Theory and Applications, In eds. J. Shanahan, Y. Qu, and J. Wiebe, 1-10. The Information Retrieval Series Vol.20, Netherlands: Springer Verlag, 2004

[Prendinger and Ishizuka 2005] Prendinger, H. and Ishizuka, M.: The Empathic Companion: A Character-based Interface that Addresses User's Affective States. Journal of Applied Artificial Intelligence, 19(3-4), 267-285, 2005

[Reeves and Nass 1998] Reeves, B. and Nass, C.: The Media Equation. How People Treat Computers, Television and New Media Like Real People and Places. Cambridge University Press, 1998

[Riloff et al. 2003] Riloff, E., Wiebe, J., and Wilson, T.: Learning Subjective Nouns Using Extraction Pattern Bootstrapping. Proceedings of the 7th International Conference on Natural Language Learning, (CoNLL-03), pp. 25-32, Edmonton, Canada, 2003

[Rosis and Grasso 2000] Rosis, F. and Grasso, F.: Affective natural language generation. Affective Interactions, Towards a New Generation of Computer Interfaces, in ed. A. Paiva, 204-218. New York: Springer-Verlag, 2000

[Russell et al. 2003] Russell, J. A., Bachorowski, J. A., and Fernandez-Dols, J. M.: Facial and Vocal Expressions of Emotion. Annual Review of Psychology, 54:329-349, 2003

[Sebastiani 2002] Sebastiani, F.: Machine Learning in Automated Text Categorization, ACM Computing Surveys. 34(1):1-47, 2002

[Sentiment!] Corpora Software, http://www.corporasoftware.com

[Shaikh et al. 2006a] Shaikh, M. A. M., Prendinger, H., and Ishizuka, M.: A Cognitively Based Approach to Affect Sensing from Text. Proceedings of the 10th International Conference on Intelligent User Interface, (IUI-06), pp. 349-351, Sydney, Australia, 2006

[Shaikh et al. 2006b] Shaikh, M. A. M., Prendinger, H., and Ishizuka, M.: ASNA: An Intelligent Agent for Retrieving and Classifying News on the Basis of Emotion-Affinity. Proceedings of the International Conference on Intelligent Agents, Web Technologies and Internet Commerce, (IAWTIC-06), pp. 133-138, Sydney, Australia, 2006

[Shaikh et al. 2007a] Shaikh, M. A. M., Prendinger, H., and Ishizuka, M.: SenseNet: A Linguistic Tool to Visualize Numerical-Valance Based Sentiment of Textual Data. Proceedings of the International Conference on Natural Language Processing, (ICON-07), pp. 147-152, Hyderabad, India, 2007

[Shaikh et al. 2007b] Shaikh, M. A. M., Prendinger, H., and Ishizuka, M.: Assessing Sentiment of Text by Semantic Dependency and Contextual Valence Analysis. Proceedings of the International Conference on Affective Computing and Intelligent Interaction,(ACII-07), pp. 191-202, Lisbon, Portugal, 2007

[Shanahan et al. 2006] Shanahan, J.G., Qu, Y. and Wiebe, J. (Eds.): Computing Attitude and Affect in Text: Theory and Applications, Springer, Netherlands, 2006

[Stock and Strapparava 2003] Stock, O. and Strapparava, C.: Getting Serious about the Development of Computational Humor. Proceedings of the 18th International Joint Conference on Artificial Intelligence (IJCAI-03), pp. 59-64, Acapulco, Mexico, 2003

[Strapparava et al. 2007] Strapparava, C., Valitutti, A. and Stock, O.: Dances with Words. Proceedings of the International Joint Conference on Artificial Intelligence (IJCAI-07), pp. 1719-1724, Hyderabad, India, 2007

[Subasic and Huettner 2001] Subasic, P. and Huettner, A.: Affect Analysis of Text Using Fuzzy Semantic Typing, IEEE Transactions on Fuzzy Systems. 9(4):483- 496, 2001

[Turing 1950] Turing, A.: Computing Machinery and Intelligence. Mind 59(236):433–460, 1950

[Turney 2002] Turney, P.: Thumbs Up or Thumbs Down? Semantic Orientation Applied to Unsupervised Classification of Reviews. Proceedings of the 40th Annual Meeting on Association for Computational Linguistics, (ACL-02), pp. 417-424, Pennsylvania, 2002

[Turney and Littman 2003] Turney, P.D. and Littman, M.L.: Measuring praise and criticism: Inference of semantic orientation from association. ACM Transactions on Information Systems (TOIS), 21(4):315-346, 2003

[Valitutti et al. 2004] Valitutti, A., Strapparava, C., and Stock, O. : Developing Affective Lexical Resources, PsychNology Journal. 2(1):61-83, 2004

[Vincent et al. 2006] Vincent, N., Dasgupta, S., and Arifin, S. M. N.: Examining the Role of Linguistic Knowledge Sources in the Automatic Identification and Classification of Reviews. Proceedings of the COLING/ACL, pp. 611-618, Sydney, Australia, 2006

[Wallace 2004] Wallace Richard: Be Your Own Botmaster, 2nd Edition, ALICE A. I. Foundation, 2004

[Weizenbaum 1966] Weizenbaum, J.: ELIZA-A computer program for the study of natural language communication between man and machine, Communications of the ACM 9(1):36-45, 1966

[Wiebe 2000] Wiebe, J.: Learning subjective adjectives from corpora. Proceedings of the 12th International Conference on Innovative Applications of Artificial Intelligence, (IAAI-00), pp. 735-740, Texas, 2000

[Wiebe et al. 2001] Wiebe, J., Bruce, R., Bell, M., Martin, M., and Wilson, T.: A corpus study of evaluative and speculative language. Proceedings of the 2nd SIGdial Workshop on Discourse and Dialogue, (Vol.16), pp. 1-10, Aalborg, Denmark, 2001

[Wiebe et al. 2004] Wiebe, J.M., Wilson, T., Bruce, R., Bell, M., and Martin, M.: Learning subjective language. Computational Linguistics, 30(3):277-308, 2004

[Wiebe et al. 2005] Wiebe, J., Wilson, T., and Cardie, C.: Annotating expressions of opinions and emotions in language. Language Resources and Evaluation, 39(2-3):165-210, 2005

[Wiebe and Mihalcea 2006] Wiebe, J. and Mihalcea, R.: Word Sense and Subjectivity. Proceedings of the Association for Computational Linguistics Conference, (ACL-06), pp. 1065-1072, Sydney, Australia, 2006

[Wilson et al. 2005] Wilson, T., Wiebe, J., and Hoffmann, P.: Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. Proceedings of the International Conference on Human Language Technology and Empirical Methods in Natural Language Processing, (HLT/EMNLP-05), pp. 347-354, Vancouver, Canada, 2005

[Yu and Hatzivassiloglou 2003] Yu, H. and Hatzivassiloglou, V.: Towards answering opinion questions: Separating facts from opinions and identifying the polarity of opinion sentences. Proceedings of the International Conference on Empirical Methods in Natural Language Processing, (EMNLP-03), pp. 129-136, Sapporo, Japan, 2003

[Zhe and Boucouvalas 2002] Zhe, X. and Boucouvalas, A. C.: Text-to-emotion engine for real time internet communication. Proceedings of the third International Symposium on Communication Systems, Network and Digital Signal Processing, (CSNDSP-02), pp. 164-168, Staffordshire, UK, 2002

APPENDIX A: PSEUDO CODE

Pseudo Code for Assessing Contextual Valence considering adjective, adverb, negation, conditionality is given below.

```
function  getValence (P)
outputValence={}
Begin
for each S_i in P do //assume 1 ≤ i ≤ n
  tripletSet_i = getSemanticParsing (S_i)
          //the output of Semantic Parser is a set of triplets for each sentence.
  for each triplet T_j, in tripletSet_i do    //we assume 1 ≤ j ≤ m, m triplets
    actorValence = ContextualValenceAttrib (actorPriorValence, actorAttributes)
    actionValence =ContextualValenceAttrib (actionPriorValence,  actionAttributes)
    objectValence = ContextualValenceAttrib (objectPriorValence, objectAttributes)
    actionObjectPairValence=setActionObjectPairVal (actionValence, objectValence)
    tripletValence = setTripletValence (actorValence, actionObjectPairValence)
    tripletValence = handleNegationAndConditionality (tripletValence, T_j)
    tripletDependency = if the token "dependency" is found then 'true' else 'false' is set
    tripletDependencyType = 'to_dependency' or 'not_to_dependency' based on tag
    tripletResult_j = {tripletValence, tripletDependency, tripletDependencyType}
  loop until all triplets are processed
  contextualValence = processTripletLevelContextualValence (tripletSet_i)
  m = sizeof(contextualValence)
  sentimentScore = average(Σ_{k=1}^{m} abs(contextualValence_k))
  valenceSign = getResultantValenceSign (contextualValence)
  SentenceValence_i = sentimentScore * valenceSign
  outputValence=  outputValence  ∪ SentenceValence_i
loop until all sentences are processed
valence = getParagraphValence (SentenceValence)
outputValence = valence ∪ {SentenceValence}
End
```

```
function  processTripletLevelContextualValence (tripletSet_i)
Begin
M= sizeOf(tripletSet_i)
ContextualValence =[ ]
for k = 1 to M-1 do
    R_1 := tripletResult_k
    R_2 := tripletResult_k+1
    if R_1.tripletDependency = true and  R_1.tripletDependency != "to_dependency"
        ContextualValence_k = setContextualValence (R_1.tripletValence, R_2.tripletValence,
"Not_To_Dependency")
    else if  R_1.tripletDependency=false
        ContextualValence_k = R_1.tripletValence
end loop k
for k = 1 to M-1 do
    R_1 := tripletResult_k
    R_2 := tripletResult_k+1
    if R_1.tripletDependency = true and  R_1.tripletDependencyType = "to_dependency"
      if ContextValence_k+1 != null then
      Begin
        ContextualValence_k = setContextualValence(R_1.tripletValence, ContextValence_k+1,
"To_Dependency")
        ContextualValence_k+1= null
      End
   Else
ContextualValence_k = setContextualValence(R_1.tripletValence, R_2.tripletValence,
"To_Dependency")
end loop k
return ContextualValence
End
```

APPENDIX B: EXPERIMENTAL RESULT FOR DATASET A

The summary of experimental result for Dataset A using different range to signal neutrality of sentences is given below.

Neutral Range	Class	Accuracy	Precision	Recall	F-Score	Average Precision	Average Recall	Average F-Score
0	Positive	83.5	81.731	**94.444**	87.629	55.716	62.899	59.082
	Negative		85.412	**94.253**	89.617			
	Neutral		.001	0	0			
-0.5 to 0.5	Positive	**84.0**	84.845	93.333	88.889	66.007	65.427	64.301
	Negative		88.172	94.253	**91.111**			
	Neutral		25	8.696	12.903			
-1.0 to 1.0	Positive	**84.0**	86.598	93.333	89.840	66.870	66.493	65.998
	Negative		**89.011**	93.103	91.011			
	Neutral		25	13.043	17.143			
-1.5 to 1.5	Positive	83.5	87.234	91.111	89.130	70.837	69.334	69.722
	Negative		87.778	90.805	89.266			
	Neutral		37.5	26.087	30.769			
-2.0 to 2.0	Positive	83	90	90	90	71.537	71.096	71.288
	Negative		86.517	88.506	87.5			
	Neutral		38.095	34.783	36.364			
-2.5 to 2.5	Positive	82.5	91.954	88.207	**90.395**	72.207	72.858	72.473
	Negative		86.207	86.207	86.207			
	Neutral		38.462	43.478	40.816			
-3.0 to 3.0	Positive	82	91.765	86.667	89.143	74.587	76.765	75.409
	Negative		83.721	82.759	83.237			
	Neutral		48.276	60.870	53.846			
-3.5 to 3.5	Positive	82	**92.771**	85.556	89.017	**76.486**	**81.042**	**78.002**
	Negative		82.143	79.310	80.702			
	Neutral		**54.545**	78.261	64.286			
-4.0 to 4.0	Positive	79	90.123	81.111	85.380	73.988	79.861	75.607
	Negative		80.488	75.862	78.107			
	Neutral		51.351	82.609	63.333			
-4.5 to 4.5	Positive	75	87.342	76.667	81.657	71.005	77.913	72.787
	Negative		74.390	70.115	72.189			
	Neutral		51.282	86.957	64.516			
-5.0 to 5.0	Positive	72	87.013	74.444	80.240	68.716	76.706	70.507
	Negative		69.136	64.368	66.667			
	Neutral		50	91.304	**64.615**			
-5.5 to 5.5	Positive	67	81.081	66.667	73.170	64.571	74.030	66.226
	Negative		65.823	59.770	62.651			
	Neutral		46.809	95.652	62.857			
-6.0 to 6.0	Positive	63.5	80	62.222	70	61.953	72.465	63.331
	Negative		60.759	55.172	57.831			
	Neutral		45.098	**100**	62.162			
-6.5 to 6.5	Positive	59.5	79.411	60	68.354	58.592	69.425	59.516
	Negative		54.545	48.276	51.220			
	Neutral		41.818	100	58.974			
-7.0 to 7.0	Positive	56	76.923	55.556	64.516	55.752	66.794	56.029
	Negative		52	44.828	48.148			
	Neutral		38.333	100	55.423			

Neutral Range	Class	Accuracy	Precision	Recall	F-Score	Average Precision	Average Recall	Average F-Score
-7.5 to 7.5	Positive	51	72.131	48.889	58.278	51.750	63.040	51.044
	Negative		49.296	40.230	44.304			
	Neutral		33.824	100	50.549			
-8.0 to 8.0	Positive	49	75	46.668	57.534	51.164	61.533	48.927
	Negative		47.826	37.931	42.308			
	Neutral		30.667	100	46.939			
-8.5 to 8.5	Positive	45.5	72.222	43.333	54.167	48.411	58.889	45.518
	Negative		44.615	33.333	38.158			
	Neutral		28.395	100	44.231			
-9.0 to 9.0	Positive	41.5	70	38.889	50	45.373	55.875	41.717
	Negative		39.683	28.736	33.333			
	Neutral		26.437	100	41.818			
-9.5 to 9.5	Positive	37.5	65.957	34.444	45.255	42.693	52.861	37.583
	Negative		38.889	24.138	29.787			
	Neutral		23.232	100	37.704			
-10.0 to 10.0	Positive	35	62.222	31.111	41.481	40.709	50.983	35.052
	Negative		38	21.839	27.737			
	Neutral		21.905	100	35.938			
-10.5 to 10.5	Positive	33.5	65.854	30	41.221	40.664	49.847	33.578
	Negative		35.417	19.540	25.185			
	Neutral		20.721	100	34.329			
-11.0 to 11.0	Positive	30.5	64.865	26.667	37.795	38.670	47.586	30.521
	Negative		31.818	16.091	21.374			
	Neutral		19.328	100	32.394			
-11.5 to 11.5	Positive	27	63.333	21.111	31.667	39.530	44.968	26.800
	Negative		38.710	13.793	20.339			
	Neutral		16.547	100	28.395			
-12.0 to 12.0	Positive	22.5	52.174	13.333	21.239	38.941	41.609	21.829
	Negative		50	11.494	18.692			
	Neutral		14.650	100	25.556			
-12.5 to 12.5	Positive	20	55.556	11.111	18.519	41.004	39.719	18.826
	Negative		53.846	8.046	14			
	Neutral		13.609	100	23.958			
-13.0 to 13.0	Positive	18.5	53.333	8.889	15.238	44.356	38.595	16.951
	Negative		66.667	6.897	12.5			
	Neutral		13.068	100	23.116			
-13.5 to 13.5	Positive	16	60	6.667	12	49.122	36.705	13.534
	Negative		75	3.448	6.594			
	Neutral		12.366	100	22.010			
-14.0 to 14.0	Positive	14.5	71.429	5.556	10.309	38.956	35.568	11.376
	Negative		33.333	1.149	2.222			
	Neutral		12.105	100	21.596			
-14.5 to 14.5	Positive	12.5	50	2.222	4.255	20.619	34.074	8.484
	Negative		.001	0	0			
	Neutral		11.856	100	21.198			
-15.0 to 15.0	Positive	11.5	.001	0	0	3.834	33.333	6.876
	Negative		.001	0	0			
	Neutral		11.5	100	20.628			

Publication List of the Author

Mostafa Al Masum Shaikh, Helmut Prendinger, and Mitsuru Ishizuka:
Sentiment Assessment of Text by Analyzing Linguistic Features and Contextual Valence Assignment. Journal of Applied Artificial Intelligence (JAAI), Submitted in October 2007

Mostafa Al Masum Shaikh, Helmut Prendinger, and Mitsuru Ishizuka:
A Linguistic Interpretation of the OCC Emotion Model for Affect Sensing from Text. Invited paper for a book chapter (to be published by Springer-Verlag)

Mostafa Al Masum Shaikh, Helmut Prendinger and Mitsuru Ishizuka:
Linguistic Interpretation of Emotions for Affect Sensing from Text. Accepted for the International Joint Conference on Natural Language Processing (IJCNLP), to be held at Hyderabad, India on 7-12 Jan, 2008

Mostafa Al Masum Shaikh, Helmut Prendinger and Mitsuru Ishizuka:
Emotion Sensitive News Agent: An Approach Towards User Centric Emotion Sensing from the News. Proceedings of the 2007 IEEE/WIC/ACM International Conference on Web Intelligence (WI'07), pp.614-620, Silicon Valley, CA, USA, (2007.11)

Mostafa Al Masum Shaikh, Helmut Prendinger and Mitsuru Ishizuka:
Assessing Sentiment of Text by Semantic Dependency and Contextual Valence Analysis. Affective Computing and Intelligent Interaction (A. Pavia, P. Prada and R. W. Picard (Eds.)), Proceedings of the International Conference of Affective Computing and Intelligent Interface (ACII-2007), Lisbon, Portugal, Springer LNCS 4738, pp.191-202 (2007.9)

Mostafa Al Masum Shaikh, Helmut Prendinger and Mitsuru Ishizuka:
Roles of Emotions: A Linguistic Interpretation of an Emotion Model for Affect Sensing from Texts. Affective Computing and Intelligent Interaction (A. Pavia, P. Prada and R. W. Picard (Eds.)), Proceedings of the International Conference of Affective Computing and Intelligent Interface (ACII-2007), Springer LNCS 4738, pp.737-738 (2007.9)

Shaikh Mostafa Al Masum, Helmut Prendinger and Mitsuru Ishizuka:
SenseNet: A Linguistic Tool to Visualize Numerical-Valance Based Sentiment of Textual Data. Proceedings of the 5th International Conference on Natural Language Processing (ICON-2007), pp.147-152, Hyderabad, India (2007.1)

Shaikh Mostafa Al Masum, Md. Tawhidul Islam, Mitsuru Ishizuka and Helmut Prendinger:
Implementation of Affect Sensitive News Agent (ASNA) for Affective Classification of News Summary. Proceedings of the International Conference on Computer and Information Technology (ICCIT2006), Dhaka, Bangladesh (2006.12)

Shaikh Mostafa Al Masum and Ishizuka Mitsuru:
Integrating Natural Language Understanding and a Cognitive Approach to Textual Emotion Recognition for Generating Human-Like Responses. Proceedings of the International Conference on Computer and Information Technology (ICCIT2006), Dhaka, Bangladesh (2006.12)

Shaikh Mostafa A. Masum, Md. Tawhidul Islam and Mitsuru Ishizuka:
ASNA: An Intelligent Agent for Retrieving and Classifying News on the Basis of Emotion-Affinity. Proceedings of the International Conference on Intelligent Agents, Web Technologies and Internet Commerce (IAWTIC'06), pp.133-138 (IEEE Computer Soc.), Sydney (2006.11)

Shaikh Mostafa Al Masum, Helmut Prendinger and Mitsuru Ishizuka:
A Cognitively Based Approach to Affect Sensing from Text. Proceedings of the 10th International Conference on Intelligent User Interface (IUI 2006), Sydney, Australia, pp.349-351 (2006.02)

Shaikh Mostafa Al Masum and Mitsuru Ishizuka**:**
An Affective Role Model of Software Agent for Effective Agent-Based e-Learning by Interplaying between Emotions and Learning. Proceedings of the 1st International Conference on Web Information Systems and Technologies (WEBIST 2005), Miami, Florida, USA, pp449-456

Shaikh Mostafa Al Masum and Mitsuru Ishizuka**:**
Making Topic-Specific Report and Multimodal Presentation Automatically by Mining the Web Resources. Proceedings of IEEE/WIC/ACM International Conference on Web Intelligence (WI2006), pp.240-246, Hong Kong (2006.12)

Shaikh Mostafa Al Masum, Mitsuru Ishizuka, and Md. Tawhidul Islam**:**
'Auto-Presentation': A Multi-Agent System for Building Automatic Multi-Modal Presentation of a Topic from World Wide Web Information. Proceedings of IEEE/WIC/ACM International Conference on Intelligent Agent Technology (WI/IAT2005), Compiegne, France, pp.246-249 (2005.9)

Shaikh Mostafa Al Masum and Mitsuru Ishizuka**:**
Software-Agent architecture for Interactive E-Learning. Proceedings of the International Conference on Web Information Systems and Technologies (WEBIST 2005), Miami, USA, pp. 525-528, (2005.5)

Shaikh Mostafa Al Masum and Md. Tawhidul Islam**:**
Designing and Implementing an Interactive Education Counselor by Agent Technology. Proceedings of the International Conference on Intelligent Agent, Web Technology, Internet Commerce (IAWTIC 2004), Gold Coast, Australia, (2004.7)

Shaikh Mostafa Al Masum and A. B. M. Baizid**:**
A Proposed Text Summarization Using Inheritable Knowledge Graph. Proceedings of the International Conference on Intelligent Agent, Web Technology, Internet Commerce (IAWTIC 2004), Gold Coast, Australia, (2004.7)

Shaikh Mostafa Al Masum and A. S. M. Mahbub Morshed**:**
Object Oriented Hybrid Software Engineering Process (SEP) model for Small Scale Software Development Firms. Proceedings of the 2nd International Conference on Computer Science and Applications (ICCSA), San Diego, (2004.6)