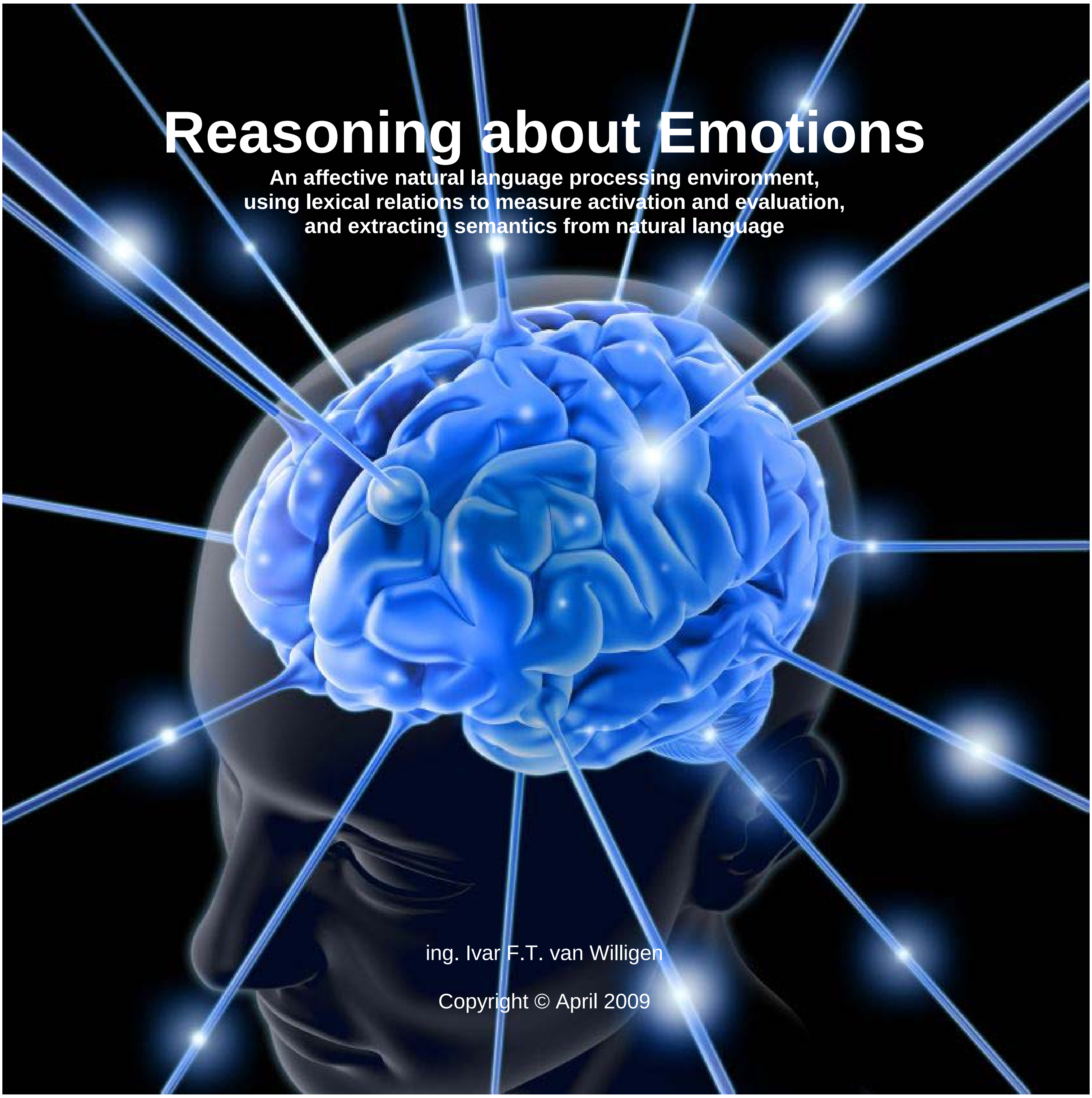

Reasoning about Emotions

**An affective natural language processing environment,
using lexical relations to measure activation and evaluation,
and extracting semantics from natural language**

ing. Ivar F.T. van Willigen

Delft University of Technology

Reasoning about Emotions

**An affective natural language processing environment,
using lexical relations to measure activation and evaluation,
and extracting semantics from natural language**

by

ing. Ivar F.T. van Willigen

**A thesis submitted in partial satisfaction
of the requirements for the degree of**

Master of Science

presented at

**Delft University of Technology,
Faculty of Electrical Engineering,
Mathematics and Computer Science
Man-Machine Interaction Group.**

April 2009

TUDelft

Delft University of Technology

Man-Machine Interaction Group
Faculty of Electrical Engineering,
Mathematics and Computer Science,
Delft University of Technology,
Mekelweg 4, 2628 CD,
Delft, Netherlands

Members of the supervising committee
prof. dr. drs. L.J.M. Rothkrantz
dhr. ir. H.J.A.M. Geers
mw. S. Fitrianie MSc.
dr. ir. P. Wiggers

Keywords
Artificial intelligence, natural language processing,
affect sensing, lexical affinity, WordNet, ConceptNet,
Dictionary of Affect in Language, syntactic parsing,
semantic parsing

Abstract

Reasoning about Emotions
An affective natural language processing environment,
using lexical relations to measure activation and evaluation,
and extracting semantics from natural language

Man-Machine Interaction Group
Faculty of EEMMCS
Delft University of Technology

Members of the supervising committee
prof. dr. drs. L.J.M. Rothkrantz
dhr. ir. H.J.A.M. Geers
mw. S. Fitrianie MSc.
dr. ir. P. Wiggers

In the field of textual affect sensing many methods have been proposed. These methods vary from keyword spotting techniques, lexical affinity, statistical natural language processing and hand-crafted models. Based on a large scale survey, two profounding theories have been selected for investigation. The first is the proposed work of (Kamps & Marx, 2001) which states that the lexical relations found in WordNet (Fellbaum, 1998) can be used to measure the activation and evaluation of words. This theory has been investigated, by implementing various search algorithms, including a multi-threaded bidirectional search algorithm, which enables us to compare the results with manually annotated word sets. Improvements to this theory have been made so that for more words the activation and evaluation values can be calculated, without compromising the results. Secondly the theory of (Liu, Lieberman, & Selker, 2003) has been investigated. This theory is based on a novel technique, by inferencing commonsense knowledge to reason about the emotional content of a given text. No full implementation has been made, but a basis has been created for future implementation. Finally, we have implemented a natural language resource toolbox for affective NLP research, called the NLP Affect Toolbox. This toolbox can be used as a programming library to support and fastly implement future research. It can also be used to conduct experiments and to explore the possibilities of state-of-art (affective) natural language processing by experienced programmers, and through a graphical user interface for others.

Acknowledgments

The Master's thesis lying before you is the result of my graduation project. In November 2007 I started my research project which I concluded in June 2008, due to the vast amount of research done in the field of textual affect sensing. From June 2008 until January 2009 I worked on my graduation project. This project has been conducted for the Man-Machine Interaction group of the faculty of EEMMCS of Delft University of Technology commissioned by prof. dr. drs. L.J.M. Rothkrantz.

I learned many things by combining the various fields of expertise, including psychology, linguistics and of course artificial intelligence. I am very pleased with the results and my contribution to the scientific world, but all of this could not have been possible without the support and assistance of some people. I would like to take the opportunity here to express my gratitude to the following people:

- Leon Rothkrantz, for his support, guidance and broad knowledge in this field of expertise, which helped me greatly in the choices I had to make to successfully conclude this thesis.

- My parents Bea and Loet van Willigen, for their everlasting confidence and all the support they gave me during and before my studies.

- My sister Sigrid and her boyfriend Chris, for always being there and for listening to my ongoing ideas about artificial intelligence.

- And of course my many friends, with whom I always have a good time during the lively discussions, studies and of course the parties.

Contents

List of Figures

List of Tables

Introduction

Chapter 1

Every day, everywhere people are communicating. The word ‘communicate’ is historically related to the word ‘common’. It stems from the Latin verb ‘*communicare’,* which means ‘to share’, ‘to make common’, and which in turn is related to the Latin word for common: ‘*communis’*. When we communicate, we make things common. We thus increase our shared knowledge, our ‘common sense’ – the basic precondition for all community. (Rosengren, 1999)

To communicate people use a language. A language is a system of finite arbitrary symbols combined according to rules of grammar for the purpose of communication. Individual languages use sounds, gestures and other symbols to represent objects, concepts, emotions, ideas, and thoughts.

As one can see from the definitions of language and communication, people use these concepts to share knowledge. Knowledge is what is known. Like the related concepts truth, belief, and wisdom, there is no single definition of knowledge on which scholars agree, but rather numerous theories and continued debate about the nature of knowledge.

Sensing the knowledge of emotions or affect of a person which is expressed in textual communication is the subject of this thesis, also known as textual affect sensing. To be able to do this there first needs to be an understanding of what affect or emotions are, how these are expressed in text and how to automate the process of sensing these features from text.

For thousands of years people have wondered about and researched emotions and still there is no single definition on which all scholars agree. Also the way people express themselves keeps on changing and so does the language they use. These facts make this area of research very difficult and explains the numerous attempts tried to solve this problem.

1.1 Motivation

In the past decades the speed of communication has increased enormously by the introduction of computers and of course the introduction of the information highway, the Internet. The world has met the capabilities of modern computers and tries to automate as much as possible, usually for improvements in efficiency and comfort. Still we experience a lot of drawbacks from the ways in which processes are being automated. This usually arises because of the lack of knowledge or the lack of interactive capabilities computers have, or have been given. The man-machine interaction group of the Delft University of Technology tries to improve this in numerous ways.

The ultimate goal of man-machine interaction (MMI) research is to enable people to interact with machines as they are used to do with other people. The only way to accomplish this is to enable machines to understand people in the same way as people understand each other. This can be done in many different ways, for example by understanding spoken language, written language, the visualization of the world which people perceive and by understanding the goals and motives of people.

As partly described in the introduction of this chapter, people use communication to make things common, to share knowledge, to create common goals and to affect or influence the environment to reach their goals. Emotions and feelings are of great importance in the way people interact and thus the way they affect their environment. Computers and the automated processes are there to support these goals and to enable people to affect the environment better.

Usually we see computers as an apparatus that helps us to work faster or communicate better. The number of capabilities a computer has to support us keeps on growing as well as the information that is manifested throughout the Internet and computers in general. This growing complexity makes the work of MMI research more important than ever. These days the way in which people try to improve the comprehension of this complexity is by improving the user interfaces, by making the interfaces used to cooperate with computers more intuitive and organized. A novel approach is using adaptive interfaces, by learning from the way the user uses applications presenting the capabilities of the application in a different structured way so that the user can find them more easily. Another awe-inspiring new way is to sense the environment or behavior or in general the emotional state of the user and respond to that by offering for example only highly needed information in stressful or pressured situations.

The trend of reinforcing the computer’s interface with knowledge and thereby enabling it to adapt itself helps us to find our way in the jungle of tools, utilities, functions, methods and other forms of capabilities a computer has to offer us. But this is probably not enough. Often users know what they want to accomplish, but do not know how to do this, or even do not know if it is possible. The same can be seen in an even more common situation by searching the Internet. People are searching for something they easily can explain to another person, but cannot figure out how to describe in an unambiguous form of keywords, and therefore get the wrong results from their search query. Again the computer does not understand the user, because the user does not have the ability to express him- or herself in the way people are used to.

In artificial intelligence the way of creating more intelligent computers is often seen as creating intelligent agents. Agents are pieces of intelligent software, which have more characteristics of a human being. They are able to communicate, reason and in some cases able to learn from new information or interaction with the environment. The interface between man and machine also requires a more agent like approach. If an agent could be created, with which people could communicate in the same manner as they do among themselves, computers would become easier to use and more efficient. If, for example, a search query would be ambiguous, the agent could just ask the user in which sense it is meant. But this would probably not even be needed, because in the beginning the query would already be a lot less ambiguous, because the user already is able to explain the query in greater detail before the search is started.

Creating intelligent agents can be done in many different ways. In general a differentiation can be made between agents that are pre-programmed (i.e. static) and agents that are able to learn (i.e. dynamic). The pre-programmed agents use the knowledge extracted from an expert or from a large bulk of information or examples to support users in a domain specific task. The agents that are able to learn need some kind of function to determine if their actions or results are complying to what is needed. Usually such a function is called a heuristic function. This function grades the result of the agent's actions, so that the agent knows if it is doing the right thing, or if it needs to adjust itself to do it right or better the next time.

Humans have the same basic mechanisms of learning. They observe their environment, imitate behavior of others, and try out new action patterns. Often they get responses upon their actions from the environment, which they use to adjust their following actions and evaluate the previous actions with. In inter-human communication the emotional responses are of great importance for this learning mechanism. For this reason the importance of recognizing emotions from many different modalities has gained much interest in the field intelligent agent research.

If machines are able to sense emotions from their users, they can use this as a heuristic function to determine the quality of their behavioral patterns and adjust them if needed, i.e. learn from their users.

1.2 Challenges

To solve the problem addressed above, to make computers as easy to interact with as people can among each other, a lot of time and research is needed. The time span in which this thesis work needs to be done is limited, so the problem to solve needs to be demarcated. In the scope of this thesis the choice has been made to look at written communication and particularly the way people use text to express their emotional state, i.e. textual affect sensing. Text is a common way to communicate with and also used in computers as the primary means of communication, e.g. e-mail, chat, etc. Since the increase in the use of chat as a way of communication, the drawbacks of using text to easily explain an emotional state have emerged. Therefore a novel way has been thought up: using emoticons.

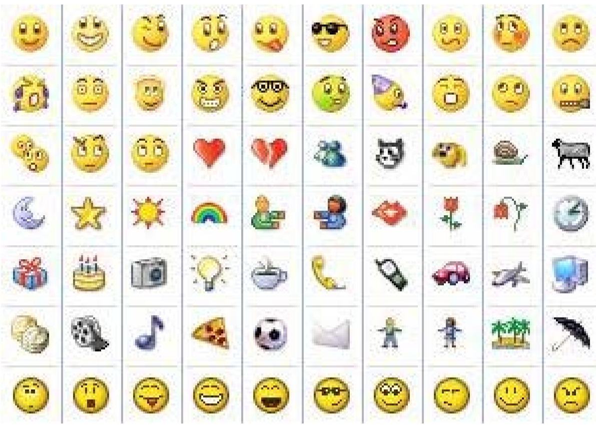

Figure 1: MSN emoticons

Emoticons are used to enable people to express their emotions faster than by using text. The emoticons are representing an idea of a feeling which is common or made common by all people and thus creating specific classification classes. An emoticon can therefore say much more than words can easily express. The idea of specific common classes of emotions is based on the psychological concept of basic emotions (e.g. anger, joy, sadness, surprise, disgust and fear) (Ekman, Friesen, & Ellsworth, 1972). A lot of other concepts also have been given an icon to speed up the process of communicating.

In normal human-human interaction a lot of communication is done via non-verbal communication. Because chat only substitutes the verbal part of communication, people lack the ability to express themselves through the non-verbal part of communication. Like in fictional literature often is seen, the way of expressing emotions through text is usually very comprehensive and takes a lot of words.

People also use their ability of empathy to sense the emotions of others by placing themselves in the situation others tell them about. This ability gives insight in the possible emotions someone is feeling, while no directs clues of these emotions can be found in the verbal as well as in the non-verbal communication.

In general, in the field of textual affect sensing, two ways of sensing affect can be distinguished. The first is sensing emotions or affect by means of word usage and the second is sensing the meaning of the text, i.e. the semantics of text. The first way has been researched extensively, numerous ways of solving this challenge have been proposed and tested as described in chapter 3. Usually these methods use some kind of keyword spotting technique to filter out the emotion containing words or phrases which then are boldly used to generalize the given piece of text. In more sophisticated methods, negation detection, co-reference resolving and other natural language processing methods are used to improve the result. The method of sensing affect by making use of the semantics of text is a more novel approach which for now has only been proposed by (Liu, Lieberman, & Selker, 2003). Their approach uses commonsense knowledge to understand what kind of emotions are usually sensed when certain events happen in peoples' lives. For example when someone says: "I just had a car accident", no emotional words are used, but by using commonsense we all can reason that this person is not feeling very positive and possibly is angry or sad.

For both approaches a basic understanding of the semantics of the text is needed. For example if someone is talking about the emotions of someone else, the recognizer needs to understand this. Or if someone is saying: "I am not happy", the recognizer needs to understand that the person is **not** happy. For people these are trivial facts, but these are very necessary for sensing affect from text.

1.3 Goals

The goals set for the research of this thesis can be generally divided into three aspects of sensing affect from text. The three divisions are general natural language processing, using statistical lexical relations for improving keyword spotting techniques and semantic extraction as a basis for commonsense reasoning over emotions expressed in text.

Because all textual affect sensing techniques require a basic understanding of the grammatical relations amongst words used in a piece of text, it is very useful to create a platform that can do this, which also can be used for general affect sensing research. This platform needs to be able to split texts into sentences and sentences into words to extract grammatical relations between words. By using these grammatical relations negation detection, semantical linking and other semantical extraction can be done more easily.

The proposed theory of (Kamps & Marx, 2001) needs to be investigated, for this method can improve the keyword spotting techniques drastically. Usually lists of emotional keywords and their hand-annotated values for the emotions are used to calculate the sensed affect from text. The theory of (Kamps & Marx, 2001) proposes an automated way of calculating activation (i.e. arousal) and evaluation (i.e. valence), but is not fully investigated. If this theory can be validated, it enables not only to calculate the activation and evaluation for specific emotional words, but can also give further insight to other basic questions, e.g. are emotional words a composition of basic emotions, like anger, sadness, joy or disgust; like blue, red and green are for all other colors.

Sensing affect by means of using commonsense knowledge is a very promising novel approach in the field of textual affect sensing. To be able to sense affect from text which does not contain any emotional keywords is something that people also use when they sense affect in communication. (Liu, Lieberman, & Selker, 2003) propose a method of doing this. By grading concepts with emotional weights, instead of grading words, this method can reason that for example the concept “having a car crash” is something that is not enjoyable. Unfortunately the implementation of this proposal is too extensive to be done for this thesis work. Therefore a basis will be founded to be used in future research, in the scope of this thesis.

So to summarize the discussed goals, determined to demarcate the thesis research, the following list:

- Creation of a natural language processing platform, for further research in the field of NLP and textual affect sensing, which is easy to use, expandable, easy to access and powerful.
- Investigation of the proposed theory of (Kamps & Marx, 2001). Is this theory a way to calculate activation and evaluation for (emotional) words?; Can there be any improvements?; Can this theory support keyword spotting techniques?
- Investigation of the proposed theory of (Liu, Lieberman, & Selker, 2003) and creation of the basis of this method for further research.

1.4 Thesis outline

This thesis is structured in five parts: “Theoretical Background”, “Model and Algorithm”, “Implementation and Experiments”, “Final results” and “Appendices”. These parts are in their turn structured in chapters.

The theoretical background part discusses various theories and concepts, that all are related to textual affect sensing. The first chapter of this part is about emotions, what emotions are and possible structures in which emotion could be structured. After this an overview is given of the various textual affect sensing methods. The third chapter describes the basic natural language processes, which form the basis of most computational language research. The last chapter in this part describes various corpora, which often are used in textual affect sensing research.

The second part: “Model and Algorithm” describes the model of the proposed research and the various algorithms that are applied to implement this model. This part consists of two chapters. The first chapter describes the model and algorithm of the proposed environment for textual affect sensing, called “NLP Affect Toolbox”. The second chapter describes the model and algorithm for the research proposed to investigate the lexical relations defined in WordNet to measure the activation and evaluation of a word, as proposed by (Kamps & Marx, 2001).

In the part “Implementation and Experiments” the implementation of the NLP Affect Toolbox will be described in the first chapter. This includes all tools and algorithms necessary to do the proposed research. The second chapter describes the experiments done and results found for the investigation of the proposed research.

In the fourth part: “Final results” the conclusion of all work done for this research is discussed. Also recommendations are made for future research. This part concludes the main body of this thesis.

In the last part the appendices can be found. In the appendices various lists of words, used to investigate the proposed research can be found. Next to these some lists of labels and tags can be found, which are used by various functionalities of the NLP Affect Toolbox.

Part I

Theoretical Background

“One can measure the importance of a scientific work
by the number of earlier publications
rendered superfluous by it.”

- David Hilbert

Emotions

Chapter 2

Princeton WordNet:

Emotion: *Any strong feeling*

Van den Bos, Gary B. (2006). APA Dictionary of Psychology. Washington, DC: American Psychological Association:

Emotion: *Complex reaction pattern, involving experiential, behavioral, and physiological elements, by which the individual attempts to deal with a personally significant matter of event.*

Although there is no exact definition or defining theory which describes emotions on which all scholars agree, there are numerous proposed ideas that describe possible mechanisms by which emotions are generated. In this chapter an overview is given of the different ideas that have contributed to the current notion of emotions. The field of research of emotions is very old, it more or less started with Aristotle and Darwin and is still practiced today.

Through time also possible spaces for representing emotions are proposed, which are important from a mathematical point of view. The idea is that there are a fixed number of basic emotions, that can be combined to make up other emotions, like the primary colors (e.g. red, blue and green) can be used to make all other colors. This idea has even been extended by primary, secondary and tertiary emotions. Numerous of sets of basic emotions are thought up and will be discussed below.

At last the activation (i.e. arousal) and evaluation (i.e. valence) space will be discussed. This idea has been proposed by (Russell & Lanius, 1984) which states that all emotions can be placed on a two dimensional space (i.e. activation, evaluation) and thus stating that these are the two dimensions from which emotions are composed.

2.1 What are emotions

The literature on emotions is rich and spans several disciplines, often with no obvious overlap or consolidating outlook. Our view of emotions has been shaped by the philosophy of Rene Descartes, the biological concepts of Charles Darwin and the psychological theories of William James, only to mention a few of the gurus of human sciences (Fragopanagos & Taylor, 2005).

Looking back at the history of emotional theories we should mention Aristotle, who classified emotions into opposites and explained the physiological and hedonic qualities associated with emotions. Later, Rene Descartes introduced the idea that a few emotions (or passions) underlie the whole of human emotional behavior. After studying the relationship between emotions and facial expressions and bodily movements, Charles Darwin drew the conclusion that emotions are strongly linked to their survival value. He also suggested that emotions have been inherited from animal precursors. During 1880's, the American psychologist William James and the Danish physiologist Carl G. Lange independently reached the conclusion that emotions arise from perception of the physiological state after he had closely examined the peripheral components of emotions such as somatic arousal. As a response to this theory the Cannon-Bard theory was proposed, by Walter Cannon and Philip Bard, which states that people first experience the emotion and act upon this feeling. So, when we see a bear, we do not become afraid because we start running, but we become afraid and then start running. A very extensive review of these classic theories as well as more contemporary ones can be found in (Solomon, 2003).

More recently, (Arnold, 1960) and (Lazarus, 1968) introduced the cognitive appraisal theory of emotions by proposing that emotions arise when a stimulus, event or situation is cognitively assessed to be carrying a personal meaning. This personal meaning is determined by personal goals and concerns and shaped by past experiences. (Fragopanagos & Taylor, 2005)

Moreover, depending on the outcome of this cognitive appraisal an appropriate emotional response is generated. In this way appraisal theorists bring together the high-level cognitive components of emotional processing and the more low-level limbic and somatic response components that together form a complex circuitry that allows us to experience emotions even in the absence of explicit awareness of the emotion-arousing episode. (Fragopanagos & Taylor, 2005)

Another long-standing debate in emotion theory, which persists to date, is whether emotions are innate or learned. At one extreme, evolutionary theorists believe in the Darwinian tradition that evolution has crafted emotions in the brain as a result of a long environment-driven adaptation to better serve the behavioral imperatives of our ancestors (Ekman, 1994), (Izard, 1992), (Neese, 1990), (Tooby & Cosmides, 1990). Strong differentiation of emotional states within the limbic system lends some support to this approach, although such differentiation need not necessarily be genetically hard-wired or be based on discrete emotion-specific brain systems. Indeed at the other extreme, many theorists take the social constructivist approach (Averill, 1980), (Orthony & Turner, 1990) which emphasizes the role of higher cortical processes (such as those involved in complex social behavior) in differentiating emotions. This camp does not accept that the strong differentiation of emotions in the limbic system is innate; but rather that it is conceivable that the limbic system contains areas that are differentially sensitive to the arousal level (activation) and to the valence (evaluation) of stimuli or events to which the subject is exposed in a non-emotion-specific way. This would allow for social influence to shape emotional responsiveness and would justify the emotional variance reported to exist across different cultural populations. At the same time, it would suggest that people from within the same social population should perceive emotion coherently. (Fragopanagos & Taylor, 2005)

Aside from emotions in their narrow sense, emotional states can be related to other structures that have similar affective qualities but quite different time courses. Moods, for instance, have a longer life than emotions and can therefore affect behavior on a larger time scale. Moreover, moods are not generated instantaneously in response to a particular object, as emotions are. Thus moods are usually experienced in a more global and diffused fashion. Nevertheless, in language the same emotional word might describe a short-lived emotion or a more protracted mood. For instance, the word ‘sad’ can be used to describe an emotion in response to some disappointing news but can also be used to describe the mood of a griever. Emotional traits have an even longer life as they reflect enduring inclinations to enter certain emotional states. Again the word-label ‘happy’ can be assigned to an emotion, a mood or a trait equally well. Thus it is clear that an automatic emotion recognizer would benefit from the use of more than one temporal scale of analysis of the signs of emotional states. In this way the emotional states recognized at each instant can be attributed to the appropriate cause (emotion, mood, trait, etc.) and mixed effects can be disentangled. (Fragopanagos & Taylor, 2005)

2.2 Basic emotions

Following a long tradition going back to Descartes and Darwin that supports the existence of a small, fixed number of discrete (basic) emotions, Tomkins proposed in 1962 (Tomkins, 1962) that there exist nine basic affective states (two are positive, one is neutral and six are negative), each indicated by a specific configuration of facial features. This assumption has been perpetuated by many researchers who followed (Ekman, Friesen, & Ellsworth, 1972), (Izard, 1971), (Oatley & Johnson-Laird, 1987) with each researcher producing their own list of basic emotions that are different in the number and the type of basic emotions with those on the others’ lists. This disparity is to say the least confusing in trying to understand the characteristics of the internal representations of the various emotional states considered to be most crucial for the development of an automatic emotion recognition system. Furthermore, while one would expect a set of basic emotions to be consistently recognized across cultures in other words, being universal-evidence suggests that there is minimal universality at least in the recognition of emotions from facial expressions (Russell J. , 1994) although this view has been challenged by (Ekman, 1994). (Fragopanagos & Taylor, 2005)

As said above basic emotions are a small set of emotions that have a special status. Which emotion is given the special status depends on the point of view of the research it has been derived from. When for example we look at emotions from the point of view of a biologist, we tend to look at the chemical reactions the body creates. But when looked from the point of view of an physiologist, we tend to look at the reaction in behavior of the person. These are two straight forward examples, some more bases of inclusion are: “universal facial expressions”, “relation to instinct” or “density of neural firing”.

In time various researchers have come up with a set of basic emotions. These vary, among other aspects, because of the point of view of the research in which they should or would be used. Below a selection of the most important sets of basic emotions:

Table 1: Various sets of basic emotions

Reference	Basic emotions	Basis for inclusion
(Arnold, 1960)	Anger, aversion, courage, dejection, desire, despair, fear, hate, hope, love, sadness	Relation to action tendencies
(Eckman, Friesen, & Ellsworth, 1982)	Anger, disgust, fear, joy, sadness, surprise	Universal facial expressions
(Frijda, 1986)	Desire, happiness, interest, surprise, wonder, sorrow	Forms of action readiness
(Gray, 1982)	Rage and terror, anxiety, joy	Hardwired
(Izard, The face of emotions, 1971)	Anger, contempt, disgust, distress, fear, guilt, interest, joy, shame, surprise	Hardwired
(James, 1884)	Fear, grief, love, rage	Bodily involvement
(McDougall, 1926)	Anger, disgust, elation, fear, subjection, tender-emotion, wonder	Relation to instincts
(Mowrer, 1960)	Pain, pleasure	Unlearned emotional states
(Oatley & Johnson-Laird, 1987)	Anger, disgust, anxiety, happiness, sadness	Do not require propositional content
(Panksepp, 1982)	Expectancy, fear, rage, panic	Hardwired
(Plutchik, 1980)	Acceptance, anger, anticipation, disgust, joy, fear, sadness, surprise	Relation to adaptive biological processes
(Tomkins, Affect theory, 1984)	Anger, interest, contempt, disgust, distress, fear, joy, shame, surprise	Density of neural firing
(Watson J. , 1930)	Fear, love, rage	Hardwired
(Weiner & Graham, 1984)	Happiness, sadness	Attribution independent

2.3 Primary and secondary emotions

We turn to discuss the division of emotions into two categories: ‘primary’ and ‘secondary’ emotions. What (Damasio, 1994) calls ‘primary’ emotions are the more primitive emotions such as startle-based fear, as well as innate aversions and attractions. These are said to arise automatically in the low-level limbic circuit. On the other hand, ‘secondary’ emotions are more subtle and sophisticated in that they require the involvement of cognitive processing to arise. These are likely to involve high-level cortical processing and even require conscious awareness.

This division of emotions directly relates to the issues discussed above. Thus the primary emotions are equivalent to the basic emotions, a thesis strongly supported by the theorists who support the basic emotions, and who usually also argue that these emotions are evolutionary crafted in the limbic system. The secondary emotions would be argued, by the supporters of innate basic emotions, to be a blend of basic emotions much in the way that different colors can be created by the mixture of red, green and blue.

On the other hand, the social constructivists would argue that secondary emotions are social constructs built on a set of rudimentary emotions such as startle and affinity / disgust. It is crucial to fully appreciate this division of emotions into primary and secondary since it is the secondary emotions that we are more concerned within the design of human–computer interfaces. However primary emotions, such as anger, certainly can surface in the sorts of interactions we are considering here, even in the interaction of a human with their computer. (Fragopanagos & Taylor, 2005)

Table 2: Primary and secondary emotions

Primary emotion	Secondary emotion	Tertiary emotions
Love	Affection	Adoration, affection, love, fondness, liking, attraction, caring, tenderness, compassion, sentimentality
	Lust	Arousal, desire, lust, passion, infatuation
	Longing	Longing
Joy	Cheerfulness	Amusement, bliss, cheerfulness, gaiety, glee, jolliness, joviality, joy, delight, enjoyment, gladness, happiness, jubilation, elation, satisfaction, ecstasy, euphoria
	Zest	Enthusiasm, zeal, zest, excitement, thrill, exhilaration
	Contentment	Contentment, pleasure
	Pride	Pride, triumph
	Optimism	Eagerness, hope, optimism
	Enthrallment	Enthrallment, rapture
	Relief	Relief
Surprise	Surprise	Amazement, surprise, astonishment
Anger	Irritation	Aggravation, irritation, agitation, annoyance, grouchiness, grumpiness
	Exasperation	Exasperation, frustration
	Rage	Anger, rage, outrage, fury, wrath, hostility, ferocity, bitterness, hate, loathing, scorn, spite, vengefulness, dislike, resentment
	Disgust	Disgust, revulsion, contempt
	Envy	Envy, jealousy
	Torment	Torment
Sadness	Suffering	Agony, suffering, hurt, anguish
	Sadness	Depression, despair, hopelessness, gloom, glumness, sadness, unhappiness, grief, sorrow, woe, misery, melancholy
	Disappointment	Dismay, disappointment, displeasure
	Shame	Guilt, shame, regret, remorse
	Neglect	Alienation, isolation, neglect, loneliness, rejection, homesickness, defeat, dejection, insecurity, embarrassment, humiliation, insult
	Sympathy	Pity, sympathy
Fear	Horror	Alarm, shock, fear, fright, horror, terror, panic, hysteria, mortification
	Nervousness	Anxiety, nervousness, tenseness, uneasiness, apprehension, worry, distress, dread

2.4 Activation – Evaluation space

An alternative solution to the problem of representing emotional states is using a continuous 2D space to which the emotional states are mapped. One dimension of this space corresponds to the valence of the emotional state and the other to the arousal or activation level associated with it. Cowie and colleagues (Cowie, et al., 2001) have called this representation the ‘activation–evaluation space’. This bipolar affective representation approach is supported in the literature (Carver, 2001), (Russell & Barrett, 1999) as well being well founded through cognitive appraisal theory. An emotional state is ‘valenced’, i.e. is perceived to be positive or negative depending on whether the stimulus, event or situation that caused this emotional state to ensue was evaluated (appraised) by the agent of the emotional state as beneficial or detrimental. This appraisal process that assigns the positive or the negative sign to the emotional state is a key idea in cognitive appraisal theory (Ellsworth, 1994). The arousal effect of emotion on the other hand goes back as far as Darwin, who suggested that emotion predisposes us to act in certain ways. More recently from an appraisal-theoretic point of view, (Frijda, 1986) proposed that emotions are to be equated with action tendencies. Thus rating an emotional state on an activation scale, i.e. the strength of the drive to act as a result of that emotional state, is an appropriate complement to the valence rating. These two values together will yield a robust but flexible solution to the issue of the most appropriate emotional state representation to be used. (Fragopanagos & Taylor, 2005)

It is also possible to relate the explicit emotional categorical labeling of emotional states to the activation–evaluation space values by representing the emotional labels themselves as points on this space. In such a translation, basic emotional labels would not map on to the activation–evaluation space uniformly. Rather they tend to form a roughly circular pattern. This is a feature which has inspired Plutchik to suggest that this may be an intrinsic structural property of emotion. So he described emotion using an angular measure ranging from acceptance (0) to disgust (180) and from apathetic (90) to curious (270), as well as the distance from the centre, which thereby defines the strength of the emotion. More generally speaking, although the activation–evaluation space is a powerful tool to describe emotional states, there will always arise some loss of information from the collapse of the structured, high-dimensional space of the possible emotional states to a rudimentary 2D space. Moreover, different results can be obtained through the different ways of performing this collapse. (Fragopanagos & Taylor, 2005)

There now exist a number of two-dimensional structures of core affect, each given a different interpretation. Figure 2 shows four available structures (rotated and reoriented to emphasize their similarity to the structure in Figure 3). From the names used in each structure, one might think that each describes different phenomena. Yet, their creators assumed that the various structures describe the same space, sometimes with a 45° rotation. And indeed, the same data set can be analyzed to yield the pleasure and arousal orientation and then rotated to yield one of the schemes at 45° (Mayer & Gaschke, 1988).

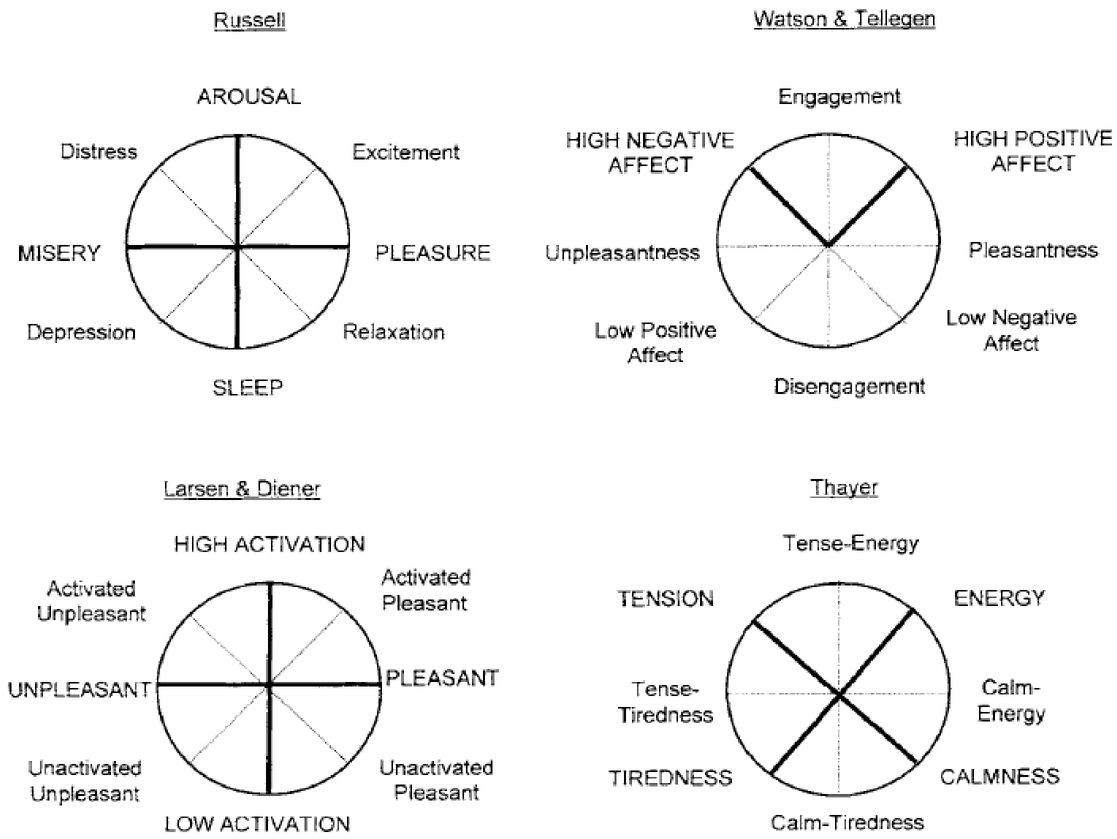

Figure 2: Four descriptive models of affect

(Russell & Barrett, Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant, 1999)

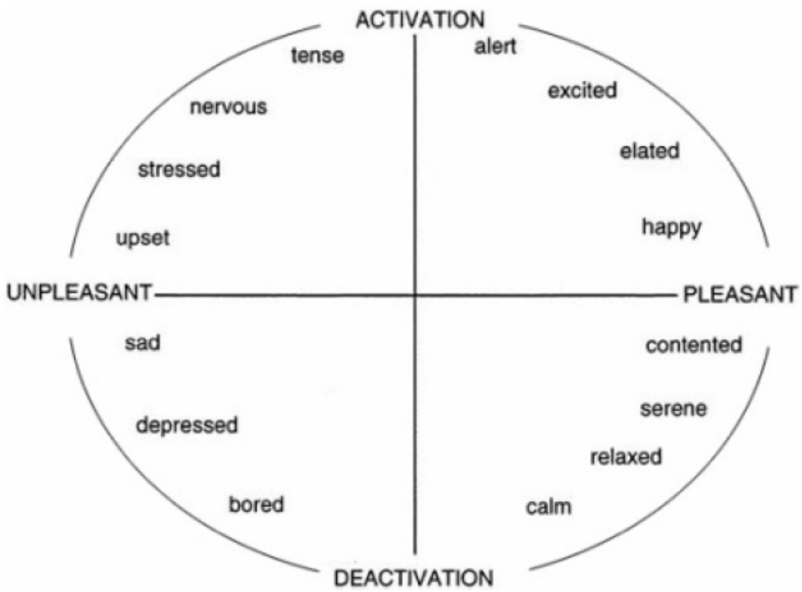

Figure 3: A Graphical representation of the circumplex model of affect (Russell)

Methods of textual affect sensing

Chapter 3

Sentiment has been studied at three different levels: word, sentence, and document level. There are methods to estimate positive or negative sentiment of words (Turney, 2002), (Esuli & Sebastiani, 2005), phrases and sentences (Kim & Hovy, 2006), (Wilson, Wiebe, & Hoffmann, 2005), and documents (Hu & Liu, 2004).
Previous approaches for assessing sentiment from text are based on one or a combination of the following techniques: keyword spotting, lexical affinity (Valitutti, Strapparava, & Stock, 2004), (Kim & Hovy, 2005), statistical methods (Pennebaker, Mehl, & Niederhoffer, 2003), a dictionary of affective concepts and lexicon, commonsense knowledgebase (Liu, Lieberman, & Selker, 2003), fuzzy logic (Subasic & Huettner, 2001), knowledge-base from facial expression (Fitrianie & Rothkrantz, 2006), machine learning (Kim & Hovy, 2006), (Wiebe, Wilson, & Cardie, 2005), domain specific classification (Nasukawa & Yi, 2003), and valence assignment (Polanyi & Zaenen, 2004), (Wilson, Wiebe, & Hoffmann, 2005), (Shaikh, Prendinger, & Ishizuka, 2007).

Some researchers proposed machine learning methods to identify words and phrases that signal subjectivity. For example, (Wiebe & 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) and (Wiebe, 2000) concentrated on learning adjectives and adjectival phrases, whereas (Wiebe, Wilson, & Cardie, 2005) focused on nouns. (Riloff, Wiebe, & Wilson, 2003) extracted patterns for subjective expressions as well. (Shaikh, Prendinger, & Ishizuka, 2007)

3.1 Keyword spotting

The most naïve approach and probably also the most popular because of its accessibility and economy. Text is classified into affect categories based on the presence of fairly unambiguous affect words like “distressed”, “enraged,” and “happy.” Elliott’s Affective Reasoner (Elliott, 1992), for example, watches for 198 affect keywords (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, Clore, & Collins, 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, “today was a happy day,” as being happy, it will likely fail on a sentence like “today wasn’t a happy day at all.” 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: “My husband just filed for divorce and he wants to take custody of my children” certainly evokes strong emotions, but use no affect keywords, and therefore, cannot be classified using a keyword spotting approach. (Liu, Lieberman, & Selker, 2003)

3.2 Lexical affinity

Slightly 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. (Liu, Lieberman, & Selker, 2003)

3.3 Statistical affect sensing

This is another approach which 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, Silverman, Hartley, Bugaj, & Ross, 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. (Liu, Lieberman, & Selker, 2003)

With statistical natural language processing the research is usually done by manually annotating corpora and hereafter using a machine learning algorithm to create a classifier. Because of the subjectivity of emotions the process of manually annotating corpora is not a trivial one. Often different people annotate the corpus differently. This is mainly due to the different interpretation of the labels used to annotate. Because of this the use of the bipolar circumplex of (Watson & Tellegen, 1985) is more convenient because the number of labels has been significantly decreased and is less open for different interpretations.

3.4 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 affective states through hand-crafted models of affect based on psychological theories about human needs, goals, and desires. (Dyer, 1987) 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. (Liu, Lieberman, & Selker, 2003)

3.5 Approach based on large-scale real-world knowledge

(Liu, Lieberman, & Selker, 2003) proposed an approach based on large-scale real-world knowledge. They used the Open Mind Commonsense Corpus (Singh, 2002) to reason about affect expressed in text. This way they have tried to make it possible to sense affect in sentences that don't contain any emotional words. The approach entails the notion that there is some user-independent commonality in people's affective knowledge of and attitudes toward everyday situations and the everyday world which is somehow connected to people's commonsense about the world. Support for this can be found in the works of, inter alia, Aristotle, Damasio, Ortony (Ortony, Clore, & Collins, 1988), W. James (James, 1884), and Minsky (Minsky, 2006). Aristotle, Damasio, and Ortony have explained that emotions are an integral part of human cognition of the everyday world, and Minsky has gone further to suggest in The Emotion Machine, that much of people's affective attitudes and knowledge is an integral part of their commonsense model of the world. Psychologist William James also noted that, just as with the rest of commonsense, the recognition of emotion in language depends on traditions and cultures, so people may not always understand another culture's expression of emotions.

Natural Language Processing

Chapter 4

Natural language processing (NLP) is a disciplinary of computer science, which tries to understand natural language (i.e. language that people use) by means of computers. Often this field is confused with computational linguistics, because of the overlap these areas of research have. Computational linguistics is a field of computer science, which deals with all kind of languages, e.g. programming languages or domain specific languages. In the scope of this thesis work, natural language is the topic of interest.

In this chapter various common NLP techniques are discussed, which are necessary to extract information from text. Sometimes these processes seem rather trivial, like splitting texts into sentences, but even in these simple tasks there is a lot of knowledge and research involved, often because of the many anomalies that can occur.

In general a distinction can be made between extracting syntactical (i.e. the structure) and semantical (i.e. the meaning) aspects from text. The past decades research in automatically extracting syntax from text has made much progress. Each language knows it's own rules of grammar, which makes this area of research easier. Extracting semantics however is still very difficult. A few of the problems that arise in doing this are; word-sense disambiguation, sometimes words have multiple meanings, people disambiguate the meaning from the context in which it is used; co-reference solving, often pronouns or other words refer to other words in the same sentence or in other sentences. Next to problems like these, which have not yet been fully solved, are problems like, what is the best structure to place the meaning or information in when it is extracted; what aspects or dimensions does meaning have.

In the field of textual affect sensing, which can be seen as a sub-field of understanding semantics, these problems also play key roles. For the research done for this thesis many existing NLP techniques are used and implemented, these are described in the following paragraphs.

4.1 Part-Of-Speech tagging

Part-of-speech tagging is the process of marking up the words in a text as corresponding to a particular part-of-speech, based on both its definition, as well as its context. A simplified form of this is commonly taught to school-age children, in the identification of words as noun, verb, adjective, preposition, pronoun, adverb, conjunction and interjection. (Proxem, 2008)

Some words can represent more than one part-of-speech at different times. This is not rare, as in natural languages, a huge percentage of word-forms are ambiguous. For example, even *"dogs"* which is usually thought of as a just a plural noun, can also be a verb: *"the sailor dogs the hatch"*. (Proxem, 2008)

In Figure 4 the gold labels are the *part of speech* labels. A complete list of *parts of speech* and their abbreviation can be found in appendix A.

4.2 Chunking

Chunking is the step next to tagging. Chunking is an analysis of a sentence, which identifies its constituents (e.g. noun groups, verbs, etc.), but does not specify their internal structure, or their role in the main sentence. (Proxem, 2008) Chunks can be seen as various sub-phrases; e.g. verb phrase, noun phrase, etc.

In the figure below (Figure 4), we can see the graphical representation of a parsed sentence (i.e. a syntax tree). In the representation the blue labels are the chunks. This sentence exists of a *noun phrase* and a *verb phrase*. The *verb phrase* again exists of a verb and a *adjective phrase*. A list of all possible phrases and their abbreviation can be found in appendix B.

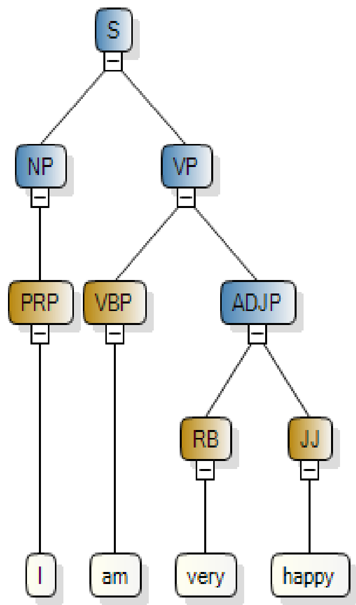

Figure 4: Example tree of parsed sentence

4.3 Splitting

Taggers and chunkers can be applied on arbitrary long texts; on the other hand, it is highly recommended to use a parser on a single sentence (that gives better results). You can use a sentence splitter to cut a text into sentences. (Proxem, 2008)

4.4 Syntax dependency extraction

Syntax dependency extraction is the process of finding syntactical relations between words. In Figure 5 we see the same sentence, but now also the syntax dependencies are extracted, which are shown below the words of the sentence. As we can see the word “I” is the noun subject of the word “happy”. And the word “very” is an adverbial modifier of the word “happy”. For the analysis of text this is very useful information. A list of all syntax dependencies can be found in appendix C.

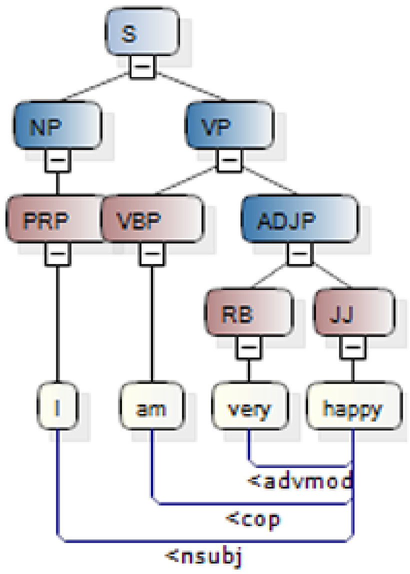

Figure 5: Example tree of dependencies in a sentence

4.5 Deep syntax dependency extraction

Deep syntax dependency extraction is a process similar to syntax dependency extraction, but tries to find a deeper structure in the sentence, instead of only finding syntactical relations between words.

For example, the sentences *“Eve loves Adam”* and *“Adam is loved by Eve”* mean roughly the same thing and use similar words. Some linguists (in particular Noam Chomsky) have tried to account for this similarity by positing that these two sentences are distinct surface forms that derive from a common deep structure. In the following figure, the syntactic dependencies are above the words, and the deep syntax dependencies are under the words. (Proxem, 2008)

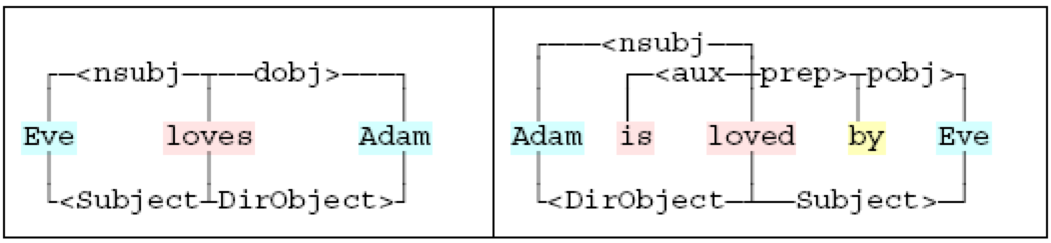

Figure 6: Example deep syntax tree

A complete list of deep syntax dependency labels can be found in appendix C.

By using these deep dependencies, predicates can be constructed. These give us even more insight in the semantics of the sentence. The found predicate, for the sentences above is:

love(**Subject**: Eve, **DirectObj**: Adam)

4.6 Co-reference extraction

In linguistics, co-reference occurs when multiple expressions in an utterance have the same referent. Co-reference can concern nouns as like as verbs. For instance, in *"Oswald killed Kennedy; this assassination was awful"*, *"assassination"* refers to the former killing.

Pronominal anaphora is a special case of co-reference, where pronouns refer to an antecedent. For instance, in *"Pam went home because she felt sick"*, *"she"* is an anaphora that refer to *"Pam"*.

An example of a sentence where the co-references are found:

"Once upon a time there was an ***old woman*** *who had a lazy* ***son****.* ***She*** *was forever scolding* ***him****, but it made no difference -* ***he*** *spent all* ***his*** *time lying in the sunshine, ignoring* ***her****.* ***His*** *main job was to look after* ***her*** *goats, but* ***he*** *preferred to sleep in the sun."*

4.7 Frames

The sub-categorization frame of a word is the number and type of arguments that it co-occurs with (i.e. the number and kind of other words that it selects when appearing in a sentence). So, in *"Indiana Jones ate chilled monkey brain"*, *"eat"* selects (or subcategorizes for) *"Indiana Jones"* (as a subject) and *"chilled monkey brain"* (as a direct object).

4.8 Thematic roles

A thematic role is the semantic relationship between a predicate (e.g. a verb) and an argument (e.g. the noun phrases) of a sentence. Thematic roles include:

- AGENT: deliberately performs the action (e.g. "Bill ate his soup quietly").
- EXPERIENCER: receives sensory or emotional input (e.g. "The smell of lilies filled Jennifer's nostrils").
- THEME/PATIENT: undergoes the action (e.g. "The falling rocks crushed the car").
- INSTRUMENT: used to carry out the action (e.g. "Jamie cut the ribbon with a pair of scissors").
- CAUSE: mindlessly performs the action (e.g. "An avalanche destroyed the ancient temple").
- LOCATION: where the action occurs (e.g. "Johnny and Linda played carelessly in the park").
- SOURCE: where the action originated (e.g. "The rocket was launched from Central Command").

Corpora

Chapter 5

In linguistics, the term corpus is used for a large set of texts, which is used for statistical analyses. This term has been adopted in the field of artificial intelligence for large databases of information, which usually also is used for statistical analyses or as data for statistical learning algorithms like neural networks.

In this chapter various corpora are described which are being used in the field of natural language processing and textual affect sensing. These corpora are not large databases of unstructured information which should be used for statistical analyses, but are the result of extensive analyses.

The following corpora are discussed in this chapter:

- WordNet; a large thesaurus of English words, grouped together by their polysemy / synonymy properties.
- Dictionary of Affect in Language (DAL); a hand annotated set of emotional words, annotated with a value for activation and evaluation.
- ConceptNet; a corpus of commonsense knowledge composed to support reasoning about affect expressed in text.

5.1 WordNet

One of the most significant attempts to realize a large scale lexical knowledge base is WordNet, a thesaurus for the English language based on psycholinguistics principles and developed at the Princeton University by George Miller (Miller, 1990); (Fellbaum, 1998). WordNet organizes lexical information in terms of word meanings, rather than word forms. It has been conceived as a computational resource, improving some of the drawbacks of traditional dictionaries, such as the circularity of the definitions and the ambiguity of sense references. English nouns, verbs, adjectives and adverbs (about 130,000 lemmas for all the parts of speech in version 1.6) are organized into synonym classes (synsets), each representing one underlying lexical concept. Lemmas are organized in synonym classes (about 100,000 synsets). WordNet can be described as a "lexical matrix" with two dimensions: a dimension for lexical relations, that is relations holding among words and thus language-specific, and a dimension for conceptual relations, which hold among senses (in WordNet they are called synsets) and that, at least in part, can be considered independent from a particular language. In Table 3 an example of a lexical matrix is reported. Word form refers to the physical utterance or inscription; word meaning refers to a lexicalized concept. F1 and F2 are synonymous, while F2 also is polysemous. Polysemy and synonymy are problems gaining access to information in the mental lexicon.

Table 3: WordNet Lexical matrix

Word meaning	**Word forms**				
	F_1	F_2	F_3	…	F_n
M_1	E_{11}	E_{12}			
M_2		E_{22}			
M_3			E_{33}		
…				…	
M_m					E_{mn}

(Valitutti, Strapparava, & Stock, 2004)

The most important lexical relation for WordNet is the similarity of meaning, since the ability to recognize synonymy among words is a prerequisite to build synsets and therefore meaning representation in the lexical matrix. Two expressions are synonymous if substitutivity is valid (in other words if the substitution of one with the other does not change the truth value of a phrase). It is important to note that defining synonyms in terms of substitutivity requires partitioning WordNet into nouns, verbs, adjectives and adverbs. This is consistent with the psycholinguistic evidence that nouns, verbs, adjectives and adverbs are independently organized in the human semantic memory. Obviously if a word pertains to more than one synset, this gives an indication of its polysemy.

Next to the polysemy / synonymy relation defined in WordNet, lemma's and synsets are also related by various other relations:

Antonymy
This is another familiar relation among words. It provides the organizing principle for adjectives. The antonym of a word *w* in general is *not-w*. However there can be exceptions to this interpretation: for instance, while "rich" and "poor" are antonyms, the statement that someone is not rich does not implies that he is poor.

Hyperonymy / Hyponymy
This corresponds to the well known ISA relation. In a different way from synonymy and antonymy, hyperonymy (and its inverse hyponymy) is a relation between meanings, so it holds among synsets. As an example the synset {apple tree} is a hyponymy of the synset {tree}, which in turn is an hyponymy of {plant}. This relation provides the organizing principle for the noun hierarchy. Given a Hyperonymy/Hyponymy hierarchy it is possible to calculate the "coordinate-terms" for a given synset. For example, among

the “coordinate-terms” for {horse} there are the synsets {mule} and {zebra}, which are common hyponyms of the synset {equine, equid}.

Meronymy / Holonymy
This represents the relation between a whole and its parts. It is a relation among synsets. Three types of holonymyc relations, along with their meronymyc inverse, are used in WordNet: member-of (e.g. {tree} is member-of {forest}); part-of (e.g. {kitchen} is part-of {apartment}); substance-of (e.g. {hydrogen} is substance-of {water H2O}).

Entailment
This is a semantic relation used for defining the verb hierarchy. From a logic point of view a proposition P “entails” a proposition Q if there is no state of the world in which P is true and Q is false. As an example the synset {snore} implies the synset {sleep}.

Troponymy
The entailment relation is at the base of the definition of the “troponymy” relation, which holds among verbs: in fact synset S1 is troponym of synset S2 if S1 implies S2 and if S1 is temporally co-extended with S2 (e.g. the synset {walk} is a troponym of the synset {move}).

(Valitutti, Strapparava, & Stock, 2004)

5.2 Whissell’s Dictionary of Affect in Language

Whissell’s dictionary of affect in language (DAL) is an annotated dictionary of words, which all have been given a value for activation and a value for evaluation. The annotation process for this corpus has been done as follows.

Word selection
Words included in the DAL set were selected in an ecologically valid manner. There were three steps involved in the selection.

Step 1: The Kucera and Francis 1969 corpus of 1,000,000 words was sampled from print media in the early 1960’s. Words from this corpus with frequencies greater than 10, which also appeared in more than one subsample were included in the DAL list. This insured that the starting words in the set would not be rare ones, or ones specific to one type of print source. Proper names were removed from the sample.

Step 2: The word set was then compared to four text samples generated by individuals rather than media. It was also compared to a large sample from juvenile literature. Unique words found in these sources were added to the list.

All of the samples employed at this step had been gathered by researchers at Laurentian University:

1. Students’ retelling of a story, 16309 words (source: Terri-Lynn Dittburner, Dr. M. Persinger)
2. Interviews on the topic of abuse, 6085 words (source: Carolyn Djaferis)
3. Adolescents’ descriptions of their emotions, 15929 words (source: Louise Wood)
4. University students’ essays, 14807 words (source: Katie Lemega)
5. Juvenile fiction of the 50’s, 60’s, 70’s, 80’s and 90’s, 82865 words (source: Micheal Dewson and Laurie Steven)

Step 3: The DAL list which contained approximately 8700 words at the end of step 2 was tested on 16 new, blindly selected, samples. It was also tested on a corpus of 350,000 words of English text collected by Whissell from many sources. The DAL demonstrated a hit rate or matching rate of approximately 90%. The hit rate of 90% meant that one would expect nine out of ten words in most English texts to be matched by the DAL.

Word rating
Rating Dimensions
The words of the DAL list were rated along the dimensions of PLEASANTNESS, ACTIVATION and IMAGERY. In each case the scale used was a three-point scale.

(1) Unpleasant	(2) In between	(3) Pleasant
(1) Passive	(2) In between	(3) Active
(1) Hard to imagine	(2) In between	(3) Easy to imagine

Method
Roughly 50% of the rating for Pleasantness and Activation were gathered using a computer-administered task. The remaining 50% and all ratings for Imagery were gathered in a paper and pencil task. Different volunteers rated different numbers of words, and some rated words along more than one dimension. Occasionally volunteers returned to be retested on a second set of words. Most volunteers were able to make about 200 rating judgments before showing signs of boredom, inattention or fatigue (the task was self-paced and could be terminated).
The data used to create the DAL involved more than 186,000 different rating judgments about words. Each word was rated for Activation and Pleasantness an average of 8 times and for Imagery 5 times.

(Whissell, 1989)

5.3 ConceptNet

ConceptNet is the largest freely available, machine-useable commonsense resource, developed by (Liu & Singh, 2004) MIT. Structured as a network of semi-structured natural language fragments, ConceptNet presently consists of over 250,000 elements of commonsense knowledge. ConceptNet was inspired dually by the range of commonsense concepts and relations in Cyc (Lenat, 1995), and by the ease-of-use of WordNet (Fellbaum, 1998), and to combine the best of both worlds. As a result, it adopted the semantic network representation of WordNet, but extended the representation in several key ways.

First, they extended WordNet's lexical notion of nodes to a conceptual notion of nodes, but they kept the nodes in natural language, because one of the primary strengths of WordNet in the textual domain is that its knowledge representation is itself textual. ConceptNet's nodes are thus natural language fragments which are semi-structured according to an ontology of allowable syntactic patterns, and accommodate both first-order concepts given as noun phrases (e.g. "potato chips"), and second-order concepts given as verb phrases (e.g. "buy potato chips").

Second, they extended WordNet's small ontology of semantic relations, which are primarily taxonomic in nature, to include a richer set of relations appropriate to concept-level nodes. At present there are 19 semantic relations used in ConceptNet, representing categories of, inter alia, temporal, spatial, causal, and functional knowledge. By combining higher order nodes with this relational ontology, it is possible to represent richer kinds of knowledge in ConceptNet beyond what can be represented in WordNet (Figure 7). For example, a layman's common sense observation that "you may be hurt if you get into an accident" can be represented in ConceptNet as EffectOf("get into accident", "be hurt"). Note that because the knowledge presentation is semi-structured natural language, there are often various ways to represent the same knowledge. This is a source of ambiguity, but by maintaining some ambiguity there is greater flexibility for reasoning.

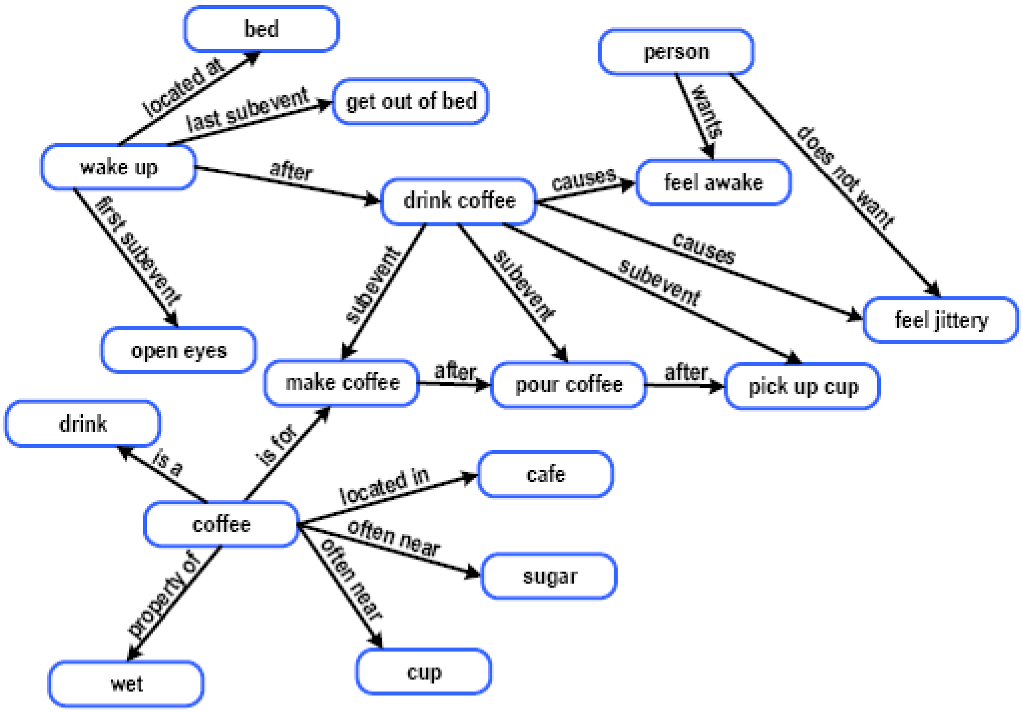

Figure 7: An excerpt from ConceptNet's semantic network

Third, they supplement the ConceptNet semantic network with some methodology for reasoning over semi-structured natural language fragments. This methodology prescribes techniques for managing the ambiguity of natural language fragments, and for determining the context-specific similarity of nodes. For example, sometimes the nodes “buy food” and “purchase groceries” should be synonymous in an inference chain, and other times, not.

Fourth, they supplement the ConceptNet semantic network with a toolkit and API which supports making practical commonsense inferences about text, such as context finding, inference chaining, and conceptual analogy.

The various relations defined in ConceptNet are structured as follows:

Table 4: ConceptNet relations

Category	Semantic Relations
K-Lines	ConceptuallyRelatedTo (e.g. ‘bad breath’ - ‘mint’)
	ThematicKLine (e.g. ‘wedding dress’ - ‘veil’)
	SuperThematicKLine (e.g. ‘western civilisation’ - ‘civilisation’)
Things	IsA (e.g. ‘horse’ - ‘mammal’)
	PropertyOf (e.g. ‘fire’ - ‘dangerous’)
	PartOf (e.g. ‘butterfly’ - ‘wing’)
	MadeOf (e.g. ‘bacon’ - ‘pig’)
	DefinedAs (e.g. ‘meat’ - ‘flesh of animal’)
Agents	CapableOf (e.g. ‘dentist’ - ‘pull tooth’)
Events	PrerequisiteEventOf (e.g. ‘read letter’ - ‘open envelope’)
	FirstSubeventOf (e.g. ‘start fire’ - ‘light match’)
	SubeventOf (e.g. ‘play sport’ - ‘score goal’)
	LastSubeventOf (e.g. ‘attend classical concert’ - ‘applaud’)
Spatial	LocationOf (e.g. ‘army’ - ‘in war’)
Causal	EffectOf (e.g. ‘view video’ - ‘entertainment’)
	DesirousEffectOf (‘sweat’ - ‘take shower’)
Functional	UsedFor (e.g. ‘fireplace’ - ‘burn wood’)
	CapableOfReceivingAction (e.g. ‘drink’ - ‘serve’)
Affective	MotivationOf (e.g. ‘play game’ - ‘compete’)
	DesireOf (‘person’ - ‘not be depressed’)

(Liu & Singh, Commonsense Reasoning in and Over Natural Language, 2004)

Related work Chapter 6

In this chapter, two promising theories are described, which have been selected on basis of an extensive literature study done before starting the research done for this thesis. These two theories both have not yet been fully explored and may improve the state-of-art textual affect sensing techniques.

The first theory is proposed by (Kamps & Marx, 2001) and is about the measurement of the activation and evaluation of emotional words. According to (Kamps & Marx, 2001) it is possible to use the lexical affinity between words, as can be found in the structure and information of WordNet, to measure the activation and evaluation. They propose a distance function to measure the lexical affinity and propose multiple functions that use this distance function to measure the activation and evaluation values.

The second theory is proposed by (Liu, Lieberman, & Selker, 2003), which states that sensing the affect from text can be reinforced with commonsense knowledge. So by exploiting ConceptNet, which holds much commonsense knowledge, they try to reason about the possible emotional information embedded in text. This would even be possible when no emotional words are used in text.

6.1 Kamp & Marx

(Kamps & Marx, 2001) proposed a novel way to automaticly calculate the affective values for emotional words. The affective dimensions used are based on the factorial analysis of extensive empirical tests (Osgood, 1957), which found that the three major factors that play a role in the emotive meaning of a word, are evaluation, potency and activity. This research also set the basis for the circumplex of affect (Russell & Lanius, 1984).

(Kamps & Marx, 2001) try to exploit the lexical relations found in WordNet to measure these factors. The organization of WordNet is not a conventional alphabetical list, but a large interconnected network of words (resembling the organization of human lexical memory). Because of this property, the distance on this graph between words can be seen as the affinity between the words or similarity between word meanings. So by calculating the distance between words, an affinity scale can be set up.

> *Two words w_0 and w_n are n-related if there exists a $(n+1)$-long sequence of words $(w_0,\ w_1, \dots, w_n)$ such that for each i from 0 to $n-1$ the two words w_i and w_{i+1} are in the same SYNSET.*

So for example the adjectives ‘good’ and ‘proper’ are 2-related since there exists a 3-long sequence (good, right, proper). Words can of course be related by many different sequences, or by none at all. The main interest in their research is on the minimal path length between two words.

> *Let MPL be a partial function such that $MPL(w_i, w_j) = n$ if n is the smallest number such that w_i and w_j are n-related.*

According to (Kamps & Marx, 2001) this function is a metric, that is, it gives a non-negative number $MPL(w_i, w_j)$ such that

i. $MPL(w_i, w_j) = 0$ if and only if $w_i = w_j$,
ii. $MPL(w_i, w_j) = MPL(w_j, w_i)$, and
iii. $MPL(w_i, w_j) + MPL(w_j, w_k) \geq MPL(w_i, w_k)$.

For words that are not connected by any sequence, the result of the MPL function is undefined. The minimal path-length function is a straightforward generalization of the synonymy relation. The synonymy relation connects words with similar meaning, so the minimal distance between words says something about the similarity of their meaning. However further experimentation quickly revealed that this relation is very weak. It turns out that the similarity of meaning waters down remarkably quick. A stricking example of this is that we also find that ‘good’ and ‘bad’ themselves are closely related in WordNet; there exists a 5-long sequence (good, sound, heavy, big, bad).

For this reason they have set up a different function, which is based on the fact that for every word that is connected to ‘good’ it also is connected to ‘bad’. This function calculates the relative distance from one word to two base words, which are eachothers antonym (e.g. ‘good’, ‘bad’).

> *Let’s define a partial function TRI of w_i, w_j, and w_k (with $w_j \neq w_k$) as*
>
> $$TRI(w_i, w_j, w_k) = \frac{MPL(w_i, w_k) - MPL(w_i, w_j)}{MPL(w_k, w_j)}$$
>
> *If any of $MPL(w_i, w_j)$, $MPL(w_i, w_k)$, or $MPL(w_k, w_j)$ is undefined, then $TRI(w_i, w_j, w_k)$ is undefined.*

By means of this function they set up three different functions to calculate the evaluation, activation and potency, by chosing the words w_j and w_k.

For evaluation:

Let's define a partial function EVA of w as $EVA(w) = TRI(w, good, bad)$. The function EVA^* is defined as follows:

$$EVA^*(\langle w_1, \ldots, w_n \rangle) = \begin{cases} EVA(w) & if\ defined \\ 0 & otherwise \end{cases}$$

For activation:

The function ACT^* of a word w is defined as follows:

$$ACT^*(w) = \begin{cases} TRI(w, active, passive) & if\ defined \\ 0 & otherwise \end{cases}$$

For potency:

The function POT^* of a word w is defined as follows:

$$POT^*(w) = \begin{cases} TRI(w, strong, weak) & ifdefined \\ 0 & otherwise \end{cases}$$

In the related paper of (Kamps & Marx, 2001), they state clearly that they do not claim that these functions assign a precise measure of for example 'goodness' or 'badness' of individual words, but rather expect that it allows differentiation between words predominantly used for expressing for example positive, negative or neutral opinions.

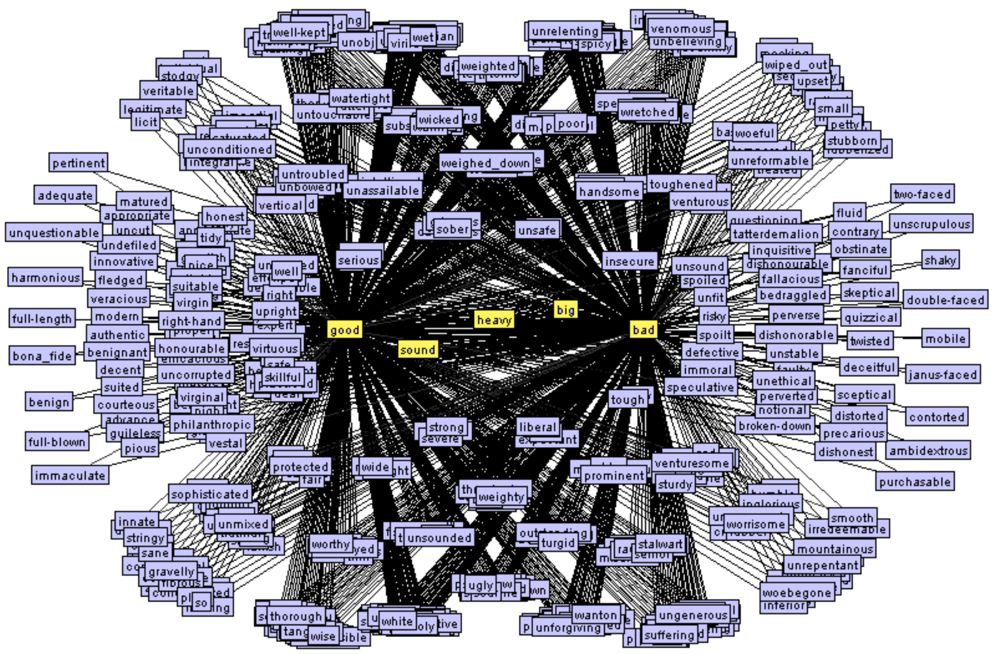

Figure 8: Path from "good" to "bad"

6.2 Liu, Lieberman and Selker

(Liu, Lieberman, & Selker, 2003) proposed a novel way for textual affect sensing. They propose to do this by means of exploiting commonsense knowledge, rather than using keyword spotting techniques that only work when specific keywords occur in the text. So for example the sentence “I just had a car accident.” does not contain any emotional keyword, but does contains affective information. A person that just had a car accident is certainly not happy, and probably sad or frightened. This kind of evaluation of emotional content embedded in text, can be extracted by using commonsense knowledge and by reasoning over this knowledge.

For this method (Liu, Lieberman, & Selker, 2003) needed a lot of affective commonsense knowledge, e.g. “A person wants popularity” or “A consequence of riding a rollercoaster may be excitement”. They started by extracting a subset of sentences from the OMCS (Open Mind Common Sense) corpus, which contained some affective commonsense. For this they used heuristics, mainly through keyword spotting of known emotional words (e.g. “happy”, “sad”, “frightening”). This subset represented 10% of the OMCS corpus.

After this, they implemented a small society of commonsense-based linguistic affect models. The output of all models are in the form of [a happy, b sad, c anger, d fear, e disgust, f surprise] based on the six basic emotions according to (Ekman, 1993). In each tuple a-f are scalers greater than 0.0, representing the magnitude of the valence of the entry with respect to a particular emotion. The following models are used.

<u>*Subject-Verb-Object-Object Model*</u>

This model represents a declarative sentence as a subject-verb-object-object frame. For example, the sentence “Getting into a car accident can be scary,” would be represented by the frame:

[<Subject>: ep_person_class, <Verb>: get_into, <Object1>: car accident, <Object2>:]*

whose value is:

[0 happy, 0 sad, 0 anger, 1.0 fear, 0 disgust, 0 surprise]

In this example, “scary” is refered to as an “emotion ground” because it confers an affective quality to the event in the sentence by association.

In this sentence, there are two verb chunks, “getting into,” and “can be.” “Can be” refers to the relation between an event and an emotion, so this relation is used to assign the event “getting into a car accident” a value. For the event phrase, the subject is omitted, but from this relation (sentence template) in OMCS it is known that the implicit subject is a person, so they fill the subject slot with a default person object. The verb is “get_into” insofar as it is a phrasal verb. The object1 slot is a noun chunk in this case, but may be an adjective chunk. The object2 slot is empty in this example, but in general, either object slot may be noun and adjective chunks, prepositional phrases or complement clauses.

This example does not cover the model’s treatment of negation or multiple SVOOs in one sentence. Negation is handled as a modifier to a subject, object, or verb. If there are multiple verb chunks in a sentence, and thus multiple SVOOs, then each a heuristic disambiguation strategy will try to infer the most relevant candidate and discard the rest.

The strength of this model is accuracy, as it preserves sentence-level event context. SVOO is the most specific of our models, and best preserves the accuracy of the affective knowledge. Proper handling of negations prevents opposite examples from triggering an entry. The limitation of SVOO however, is that because it is rather specific, it will not always be applicable.

Concept-Level Unigram Model
For this model, concepts are extracted from each sentence. Concepts in this case are verbs, noun phrases, and standalone adjective phrases. Concept, which are obviously affectively neutral by themselves (e.g. "get," "have") are excluded using a stop list. Each concept is given the value of the emotion ground in the sentence. For example, in the sentence: "Car accidents can be scary," the following concept is extracted and is given a value:

[<Concept>: "car accident"]

Value:

[0 happy, 0 sad, 0 anger, 1.0 fear, 0 disgust, 0 surprise]

Negations are handled roughly by fusing the prefix "not_" to the affected verb. Noun phrases which contain adjectival modifiers are generalized by stripping the adjectives. Then, both the original and generalized noun phrases are added to the model, with the generalized noun phrase necessarily receiving a discounted value.

Concept-level unigrams are not as accurate as SVOOs because they relate concepts out of sentence-level context to certain affective states. However, this model is more often applicable than SVOO because it is more independent of the surface structure of language (the specific syntax and word-choices through which knowledge is conveyed).

Concept-Level Valence Model
This model differs from the above-mentioned concept-level unigram model in the value. Rather than the usual six-element tuple, the value is just a vector between –1.0 and 1.0, indicating that a word has positive or negative connotations.

Associated with this model are hand-coded rules to interpret concept-level valences at the event level. For example, knowing the valences of "wreck" and "my car," we can deduce that the sentence, "I wrecked my car" has negative affect. To make this deduction, they invoke the following rule:

narrator	neg-verb	pos-object → neg-valence
I	WRECKED	MY CAR

Although this model does not result in a complete mapping into the six emotions, it produces a more reliable gist than the concept-level unigram model because it takes event-level context into account. It is useful in disambiguating a story sentence that the other models judge to fall on the cusp of a positive emotion and a negative emotion.

Modifier Unigram Model
This model assigns a six-emotion tuple to each verb and adverbial modifier found in a sentence. The motivation behind this is that sometimes modifiers are wholly responsible for carrying the emotion of a verb or noun phrase, like in the sentences:

"Moldy bread is disgusting", "Fresh bread is delicious".

After implementing the described models they use some smoothing models to generalize the various emotive tuples derived from the sentences to generate an overall sense of affect.

(Liu, Lieberman, & Selker, A Model of Textual Affect Sensing using Real-World Knowledge, 2003)

Part II

Model and algorithm

"The arousing of prejudice, pity, anger, and similar emotions has nothing to do with the essential facts, but is merely a personal appeal to the man who is judging the case."

- Aristotle

NLP Affect Toolbox Chapter 7

Because of the wide range of research done in the field of natural language processing and textual affect sensing, the means of keeping an overview and exploring the different theories becomes a tough job. In natural language processing the trend has grown to combine several techniques and theories into a manageable environment, that can be used as a basis for research done in this area. For textual affect sensing, such an environment does not exist yet.

In this chapter a model is sketched for this textual affect sensing environment, called "NLP Affect Toolbox". The first paragraph will give more insight in the selection of the different tools and corpora. The second paragraph will desribe the model of this environment.

7.1 Selection of tools and corpora

A selection of tools has been made by doing an in-depth literature study of the theories and corpora used in the field of NLP and textual affect sensing. The following tools and corpora have been chosen to be implemented in the toolbox:

Table 5: NLP Affect Toolbox – corpora

Corpora
WordNet (version 3.0)
ConceptNet (version 2.1)
Dictionary of Affect in Language

Table 6: NLP Affect Toolbox – tools

Tools
WordNet browser
Minimal Path Length calculator
Cloud distance calculator
DAL browser
ConceptNet browser
Syntactical tree plotter
Semantic text parser
Activation – evaluation plotter

Table 7: NLP Affect Toolbox – NLP

Natural Language Processing
Part-Of-Speech tagging
Chunking
Splitting
Syntax dependency extraction
Deep syntax dependency extraction
Co-reference extraction
Frame / predicate extraction
Thematic role extraction

Table 8: NLP Affect Toolbox – affect sensing

Textual affect sensing
Minimal Path Length
Cloud distance
Deep semantic parsing

The corpora are selected because of the number of theories in which they are used. They are also important for the research done for this thesis. WordNet is used for basic natural language processes, e.g. part-of-speech tagging, but can also be used for measuring lexical affinity between words as desribed by (Kamps & Marx, 2001). ConceptNet is a large corpus of commonsense knowledge created and used by (Liu, Lieberman, & Selker, 2003) to reason about affective information embedded in text. The dictionary of affect in language created by (Whissell, 1989) is a large set of manually annotated emotional words, which often is used in keyword spotting techniques to textual affect sensing. The NLP affect toolbox will contain these three corpora because they support many different theories of textual affect sensing.

For almost all theories proposed to solve the problem of textual affect sensing a basic set of natural language processing tools is needed. These tools are generally used to extract syntactical and semantical properties of a given text, which are used to extract the affect embedded in the text. For the NLP Affect toolbox a selection has been made of syntactical and semantical natural language processes. First of all, the most basic processes, e.g. splitting, chunking and part-of-speech tagging, which are used to extract the words from texts and to give them the right part-of-speech (e.g. adjective, noun, verb). The tagged words can then be used to extract (deep) syntax dependencies. The semantical processes selected are co-reference, frame / predicate and thematic role extraction. All of these processes extract many features from text to be used by textual affect sensing processes. These processes are all discussed in chapter 4.

The tools that are described in the tables above can be seen as functions of the toolbox and will be accessable through a graphical user interface in the toolbox. This interface gives people, who are not familiar with programming environments, insight in the different corpora and the ability to see what the different natural language processes and textual affect sensing processes can do. This interface can be used in further research and are used in the research and experiments done for this thesis.

The three textual affect sensing functionalities are created to support the research done for this thesis. They make use of the other underlying processes that will be implemented in the toolbox. The minimal path length and cloud distance tool will be used to investigate the lexical relations in WordNet to measure the activation and evaluation, this will be described in chapter 8.

7.2 Model

In this paragraph the model of the NLP affect toolbox is described. First of all an overview is given of the various tools and the corpora, and how they relate to each other. This overview relates to the various interfaces the NLP affect toolbox will contain. After this an overview is given of all functionalities the environment contains. This relates to the NLP affect toolbox as an programming library for textual affect sensing research.

7.2.1 Interface

The figure below gives an overview of the various tools implemented in the NLP affect toolbox by means of a graphical user interface. These tools can thus be used by anyone without the knowledge of programming. The different corpora are included in this overview to show how they relate to the various tools.

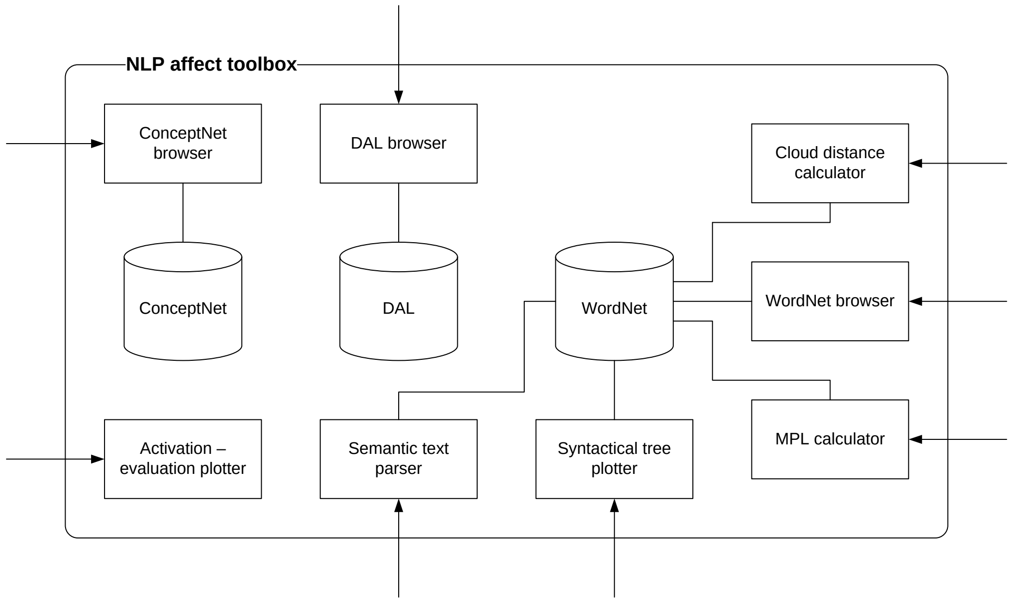

Figure 9: Interface overview NLP affect toolbox

The browser tools are rather straightforward and can be used to lookup information from the corpora. In this way insight can be gained of the underlying structure of the information in the corpora. These tools can also be used just to see what kind of information the various corpora contain.

The semantic text parser and the syntactical tree plotter give insight in the various natural language processes that are implemented in the toolbox. They show the capabilities of the current state-of-art NLP, which form the basis for textual affect sensing theories. These tools can be used to see what kind of features can be extracted from text and how this works for different kinds of sentences.

The MPL calculator and cloud distance calculator tools will be used in the research for the lexical relations of WordNet to measure the activation and evaluation, this will be described in chapter 8.

7.2.2 Functionalities

In the figure below a more detailed view is given of the functionalities of the NLP affect toolbox. In this figure the browser functionalities are left out. The browser functionalities are straightforward lookups of the data contained in the various corpora. These processes will all be described in detail in chapter 9, where the implementation of the toolbox is put forward.

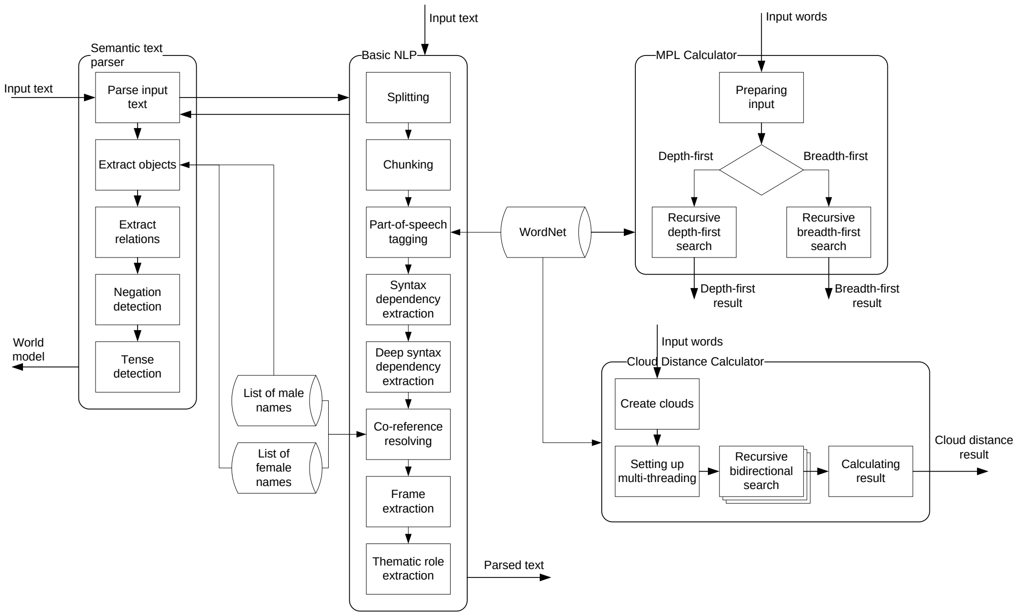

Figure 10: Functional overview NLP affect toolbox

Lexical relations to measure activation and evaluation

Chapter 8

In this chapter the proposed research for the investigation of the lexical relations to measure the activation and evaluation of a word, as proposed by (Kamps & Marx, 2001) is described in detail. First this lexical relation to be used, which is embedded in the WordNet corpus, will be further explained by means of an example. Secondly the algorithms are explained by which the work of (Kamps & Marx, 2001) will be implemented. Furthermore two possible improvements to this theory are discussed and explained.

8.1 The lexical relation in WordNet explained

As said before WordNet is a large corpus of English words. These words are related by their lexical meaning. The lexical meaning is the semantical concept behind a word. These concepts are called synsets in WordNet. Each synsets has a number of words related to it, by which it can be expressed, called senses. And in its turn each word can belong to multiple synsets, which it can express. For example the word “good”. This word has three word forms, namely: noun, adjective and adverb.

The noun “good” has 4 senses:

1	benefit	“for your own good”; “what’s the good of worrying?”
2	moral excellence or admirableness	“there is much good to be found in people”
3	that which is pleasing or valuable or useful	“weigh the good against the bad”; “among the highest goods of all are happiness and self-realization”
4	articles of commerce	

The adjective “good” has 21 senses:

1	having desirable or positive qualities especially those suitable for a thing specified	"good news from the hospital"; "a good report card"
2	having the normally expected amount	"gives good measure"; "a good mile from here"
3	morally admirable	
4	deserving of esteem and respect	"ruined the family's good name"
5	promoting or enhancing well-being	"the experience was good for her"
6	agreeable or pleasing	"we all had a good time"; "good manners"
7	of moral excellence	"a genuinely good person"
8	having or showing knowledge and skill and aptitude	"a good mechanic"
9	thorough	"had a good workout"; "gave the house a good cleaning"
10	with or in a close or intimate relationship	"a good friend"
11	financially sound	"a good investment"
12	most suitable or right for a particular purpose	"a good time to plant tomatoes"
13	resulting favorably	"it's a good thing that I wasn't there"; "it is good that you stayed"
14	exerting force or influence	"a warranty good for two years"
15	capable of pleasing	"good looks"
16	appealing to the mind	"good music"
17	in excellent physical condition	"good teeth"; "I still have one good leg"
18	tending to promote physical well-being; beneficial to health	
19	not forged	"a good dollar bill"
20	not left to spoil	"the meat is still good"
21	generally admired	"good taste"

The adverb “good” has 2 senses:

1	(often used as a combining form) in a good or proper or satisfactory manner or to a high standard (`good' is a nonstandard dialectal variant for `well')	"a task well done"; "the party went well"; "he slept well"
2	completely and absolutely (`good' is sometimes used informally for `thoroughly')	“he was soundly defeated"; "we beat him good"

So all together the word “good” is a member of 27 synsets. If we take a look for example at the synset of the 11th sense of the adjective “good”, there are 4 words to express this concept (synset):

Word	Number of senses
dependable	4
good	27
safe	7
secure	11

According to the theory of (Kamps & Marx, 2001) the words “good” and “safe” are thus 1-related. Using this relation WordNet can be seen as a large undirected graph. As (Kamps & Marx, 2001) have described in their theory, this relation could be used to measure the level of activation and evaluation by using the ACT and EVA function as described in paragraph 6.1.

Below a graphical representation is given of all words that are related to the word “good” via the synset relation defined in WordNet. Such a representation can be made by the application found on www.visualthesaurus.com.

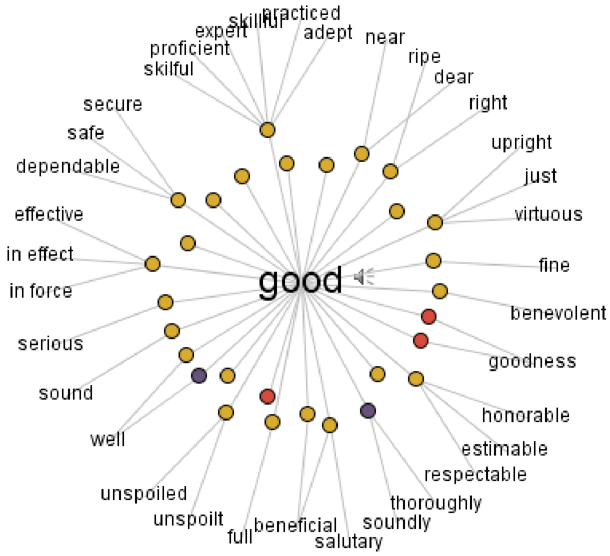

Figure 11: Graphical representation of the lexical relations of "good"

8.2 Proposed research

The theory proposed by (Kamps & Marx, 2001) is very significant for textual affect sensing, in particular for various keyword spotting techniques. This theory can also be extended to support other theories, for example in the theory of (Liu, Lieberman, & Selker, 2003) the affective values for the concepts in ConceptNet can be improved by calculating their lexical affinity.

Only (Kamps & Marx, 2001) did not investigate the method thoroughly enough. They state for example:

> *"Note that we do not claim that the EVA function assigns a precise measure of the 'goodness' or 'badness' of individual words (if such a thing is possible at all). Rather, we can only expect that it allows us to differentiate between words that are predominantly used for expressing positive opinions (values close to 1), or for expressing negative opinions (values close to -1), or for neutral words (values around 0)."*

Therefore a proposition is made to investigate the measurement of activation and evaluation by calculating the lexical affinity between words in WordNet. First of all this investigation will be about the quality of the ACT and EVA function. A comparison will be made between the calculated values of the functions and manually annotated values from various sources. Because they do not claim that these functions calculate a precise measure, a comparison will be made in two different ways. First a numerical comparison will be made (i.e. comparing the precise values), and secondly a classification comparison will be made (i.e. is a for example positive calculated word also positive annotated manually).

Because (Kamps & Marx, 2001) claim that it allows us to differentiate between words that are predominantly used for expressing positive and negative affect an investigation will also be done for this kind of words. From the DAL corpus lists of words can be obtained that are rated as very positive, negative, active and passive, these lists will be compared to the calculated values of the ACT and EVA functions for these words.

In initial experiments done with the proposed functions a few problems arose. The first problem seen was that for many words no path exists at all. The second problem was the question if only adjectives should be used to find paths between words. To solve the first problem two novel approaches were constructed. These approaches are called "shallow cloud method" and "deep cloud method". They will be explained in detail in the next paragraph. These methods will also be included in the investigation, by comparing the results to the manually annotated word sets and the results of the ACT and EVA functions. The second problem will be solved by comparing the results of the ACT, EVA, shallow cloud and deep cloud method when only adjectives are used, with the results of these methods when all words are used.

<u>So to summarize the proposed research, for the investigation of the proposed theory of (Kamps & Marx, 2001), the following research questions are formulated:</u>

- Can the lexical affinity scale as proposed by (Kamps & Marx, 2001) be used to measure the activation and evaluation of words?
- Can the measurement of activation and evaluation be used to classify words as positive, negative, active and passive?
- Does the shallow cloud method improve the measurement? (this can be seen as improving in the number of words that can be calculated and as improving the measurement itself)
- Does the deep cloud method improve the measurement?
- Does the measurement improve when all words (instead of only adjectives) are used to calculate the activation and evaluation values?

In chapter 10 these questions will be answered by means of several experiments.

8.3 Algorithm

8.3.1 Basic search algorithms

To answer the research questions setted up in the previous paragraph a lot of MPL calculations need to be done. Because of the size of WordNet, version 3.0 has 117659 synsets, the complexity of the MPL function for time and for space is not trivial. To cope with this complexity a number of artificial intelligence search techniques are applied, including depth-first, breadth-first and bidirectional. Below the various techniques will be explained.

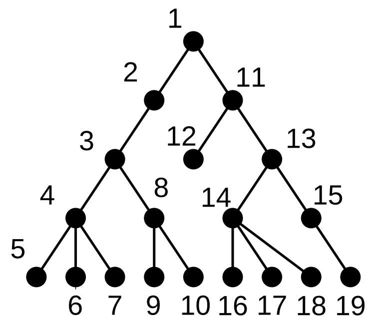

Figure 12: Depth-first search

Depth-first search is a very commonly used technique to walk through trees. The search algorithm always expands the deepest node in the current fringe of the search tree. The numbers next to the nodes show the order in which the algorithm checks the nodes.

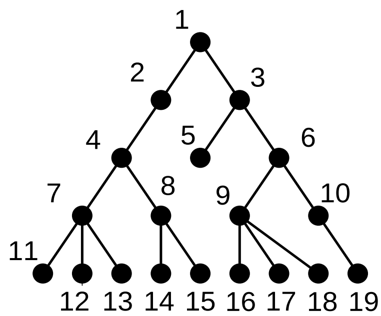

Figure 13: Breadth-first search

Breadth-first is a strategy in which the root node is expanded first, then all the successors of the root node are expanded next, and so on. The numbers next to the nodes show the order in which the algorithm checks the nodes again.

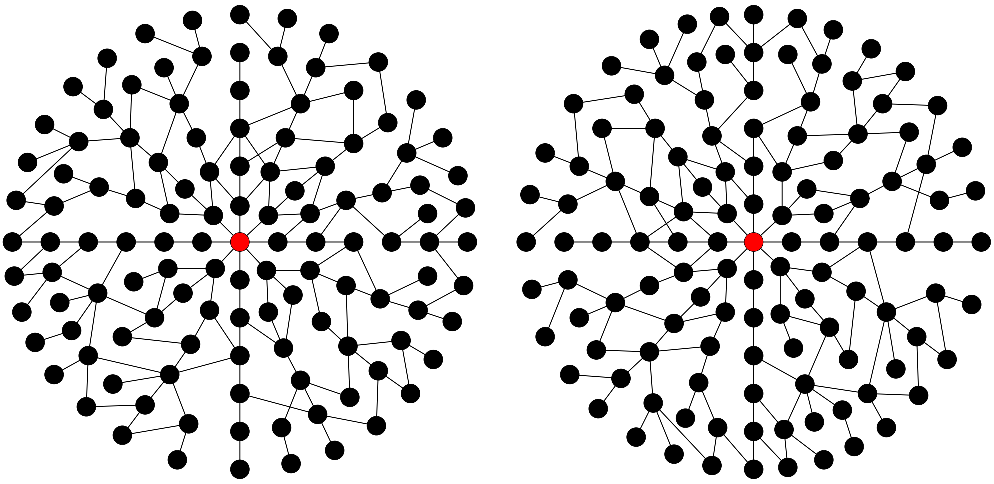

Figure 14: Bidirectional search

The idea behind bidirectional search is to turn two simultaneous breadth-first searches, one forwards from the initial node and the other backward from the goal node, stopping when the two searches meet in the middle.

The space and time complexities of these search algorithm are as follows:

Table 9: Search algorithm space and time complexity

Algorithm	Time complexity	Space complexity
Depth-first	$O(b^m)$	$O(bm)$
Breadth-first	$O(b^{d+1})$	$O(b^{d+1})$
Bidirectional	$O(b^{\frac{d}{2}})$	$O(b^{\frac{d}{2}})$

In which, b is the branching factor; d is the depth of the shallowest solution; m is the maximum depth of the search tree. (Russell & Norvig, 2003)

8.3.2 Shallow and deep cloud method

As said in paragraph 8.2, one of the initial problems with the MPL function was that for many words there is no shortest path and so no activation or evaluation can be calculated. A possible solution for this problem is to use the other relations that WordNet has. Among the relations in WordNet next to the synset (polysemy / synonymy) relation are:

Table 10: WordNet relation types

Relation type	Description	Shallow cloud method	Deep cloud method
AdjectiveClusterHead	Adjectives in WordNet are arranged in clusters containing head synsets and satellite synsets. Most head synsets have one or more satellite synsets, each of which represents a concept that is similar in meaning to the concept represented by the head synset		X
AdjectiveClusterMember	The members of the cluster, if the synset is a cluster head		X
AdjectiveParticiple	The participle adjective		X
AdjectiveSimilar	A similar adjective		X
Antonym	Opposite		
Derivation	A word that is derived from the current word	X	X
Hypernym	A word that is more generic or abstract than a given word (color is hypernym of red)		X
Hyponym	A word that is more specific than a given word (red is hyponym of color)		
InstanceHypernym	A word that is more generic or abstract than a given word (color is hypernym of red)		X
InstanceHyponym	A word that is more specific than a given word (red is hyponym of color)		
SeeAlso	A word similar in meaning to the a given word	X	X

Two possible improvements have been thought up, which differ in the number of extra relations they use. These methods only use the extra relations to expand the number of starting and goal words. This is why the methods are called “shallow cloud method” and “deep cloud method”.

So graphically represented, in Figure 15, the two red nodes (i.e. words) are the starting and goal node. From these nodes, by using the relations as describes above, direct connected nodes are found. The starting node together with its direct connected nodes is called the starting cloud. And the goal node and its direct connected nodes is called the goal cloud.

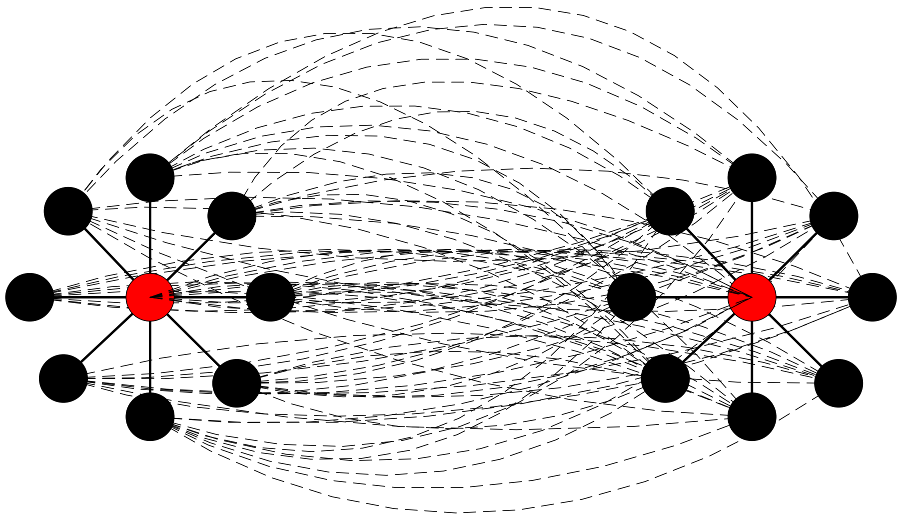

Figure 15: Cloud method

We try to find as many as possible shortest paths from the (NM) number of possible paths, in which N is the size of the starting cloud and M is the size of the goal cloud. We calculate the average path length and use this as the n-relatedness between the clouds. The variance of these paths will help to tell the quality of average calculated relatedness and of course in general about the quality of the method.

8.3.3 Pseudo-codes

Below the various algorithms will be explained by using pseudo-code. First of all a description is given of the recursive MPL function implemented by means of a recursive depth-first, breadth-first and by means of a bidirectional search algorithm. After this the pseudo-code is given of the shallow and deep cloude methods and how they use the recursive bidirectional MPL function.

```
new global list of MPLDepthFirstResults
new global list of lemmas: lemmasToLookFor
new global maximumDepth
new global onlyUseAdjectives

function FIND-SENSES(lemma) returns all senses of the lemma
   new list of senses

   if onlyUseAdjectives then
      add WORDNET.FINDSENSES(lemma, PartOfSpeech.Adjectives)
                 to list of senses
   else
      for each pos in the PartOfSpeech enumerator do
         add WORDNET.FINDSENSES(lemma, pos) to list of senses

   return list of senses

function GET-RELATED-LEMMAS(lemma) returns a list of related lemmas
   new list of lemmas

   for each sense in FIND-SENSES(lemma)
      add all lemmas of the synset of this sense to the list of lemmas

   return the list of lemmas

function RECURSIVE-DF-SEARCH(lemma, depth, walkedWords, walkedPath)
   if depth <= maximumDepth then
      relatedLemmas ← GET-RELATED-LEMMAS(lemma)

      remove lemmas from relatedLemmas that also occur in the walkedWords list to
      prevent circular references

      for each relatedLemma in relatedLemmas do
         add relatedLemma to walkedPath

         if relatedLemma ∈ lemmasToLookFor then
            new MPLDepthFirstResult
            MPLDepthFirstResult.foundAtDepth ← depth
            MPLDepthFirstResult.walkedPath ← walkedPath
            add MLPDepthFirstResult to MPLDepthFirstResults
         else
            add relatedLemma to walkedWords
            RECURSIVE-DF-SEARCH(relatedLemma, depth + 1, walkedWords,
                walkedPath)
```

```
function DEPTH-FIRST-MPL(startWord, goalWord)
   Initialize global variables

   if startWord ≠ goalWord then
      startingLemmas ← retrieve all senses of startWord from WordNet
      lemmasToLookFor ← retrieve all senses of goalWord from WordNet
      for each lemma in startingLemmas do
         new list of lemmas: walkedPath
         add lemma to walkedPath
      RECURSIVE-DF-SEARCH(lemma, 1, startingLemmas, walkedPath)
```

Figure 16: Recursive Depth-First MPL algorithm

```
new global list of lemmas: lemmasToLookFor
new global list of lemmas: walkedLemmas
new global maximumDepth
new global onlyUseAdjectives

function FIND-SENSES(lemma) returns all senses of the lemma
   new list of senses

   if onlyUseAdjectives then
      add WORDNET.FINDSENSES(lemma, PartOfSpeech.Adjectives)
                  to list of senses
   else
      for each pos in the PartOfSpeech enumerator do
         add WORDNET.FINDSENSES(lemma, pos) to list of senses

   return list of senses

function GET-RELATED-LEMMAS(lemma) returns a list of related lemmas
   new list of lemmas

   for each sense in FIND-SENSES(lemma)
      add all lemmas of the synset of this sense to the list of lemmas

   return the list of lemmas

function GET-RELATED-LEMMAS(lemmas) returns a list of related lemmas
   new list of lemmas

   for each lemma in lemmas
      add GET-RELATED-LEMMAS(lemma) to list of lemmas

   return the list of lemmas

function RECURSIVE-BF-SEARCH(lemmas, depth) returns depth if a path is found
   if depth <= maximumDepth then
      relatedLemmas ← GET-RELATED-LEMMAS(lemmas)

      for each lemma in relatedLemmas
         if lemma ∈ lemmasToLookFor then
            return depth
         else
            add lemma to walkedLemmas
            remove lemma from relatedLemmas

      return RECURSIVE-BF-SEARCH(relatedLemmas, depth + 1)

function BREADTH-FIRST-MPL(startWord, goalWord) returns value ∈ N, or failure
   Initialize global variables

   if startWord ≠ goalWord then
      startingLemmas ← retrieve all senses of startWord from WordNet
      lemmasToLookFor ← retrieve all senses of goalWord from WordNet

      return RECURSIVE-BF-SEARCH(startingLemmas, 1)
   else
      return 0
```

Figure 17: Recursive Breadth-First MPL algorithm

```
new global list of lemmas: lemmasToLookFor
new global list of lemmas: walkedLemmas
new global maximumDepth
new global onlyUseAdjectives

function FIND-SENSES(lemma) returns all senses of the lemma
   new list of senses

   if onlyUseAdjectives then
      add WORDNET.FINDSENSES(lemma, PartOfSpeech.Adjectives)
                  to list of senses
   else
      for each pos in the PartOfSpeech enumerator do
         add WORDNET.FINDSENSES(lemma, pos) to list of senses

   return list of senses

function GET-RELATED-LEMMAS(lemma) returns a list of related lemmas
   new list of lemmas

   for each sense in FIND-SENSES(lemma)
      add all lemmas of the synset of this sense to the list of lemmas

   return the list of lemmas

function GET-RELATED-LEMMAS(lemmas) returns a list of related lemmas
   new list of lemmas

   for each lemma in lemmas
      add GET-RELATED-LEMMAS(lemma) to list of lemmas

   return the list of lemmas

function RECURSIVE-BIDIRECTIONAL-SEARCH(startLemmas, goalLemmas, depth)
   new list of lemmas: newStartLemmas
   new list of lemmas: newGoalLemmas

   if depth <= maximumDepth then
      relatedStartLemmas = GET-RELATED-LEMMAS(startLemmas)
      relatedGoalLemmas = GET-RELATED-LEMMAS(goalLemmas)

      // forward search
      for each lemma in relatedStartLemmas do
         if lemma ∈ goalLemmas then
            return depth
         else
            if lemma not ∈ walkedStartLemmas then
               add lemma to walkedStartLemmas
               add lemma to newStartLemmas

      // backward search
      for each lemma in relatedGoalLemmas do
         if lemma ∈ startLemmas then
            return depth
         else
            if lemma not ∈ walkedGoalLemmas then
               add lemma to walkedGoalLemmas
               add lemma to newGoalLemmas
```

```
    // intermediate search
    if depth + 1 < maximumDepth then
      for each lemma in relatedStartLemmas do
        if lemma ∈ relatedGoalLemmas then
          return depth + 1

    return RECURSIVE-BIDIRECTIONAL-SEARCH(newStartLemmas,
                newGoalLemmas, depth + 2)

function BIDIRECTIONAL-MPL(startLemma, goalLemma)
  Initialize global variables

  new list of lemmas: startLemmas
  new list of lemmas: goalLemmas

  add startLemma to startLemmas
  add goalLemma to goalLemmas

return RECURSIVE-BIDIRECTIONAL-SEARCH(startLemmas, goalLemmas, 1)
```

Figure 18: Recursive Bidirectional MPL algorithm

```
new global cloudScale    // input words only, shallow or deep cloud

function FIND-CLOUD(word) returns the list of lemmas that form the cloud
   new list of lemmas: cloud

   for each sense in WORDNET.FINDSENSES(word) do
      if sense not ∈ cloud then
         add sense to cloud

      if cloudScale != onlyInputWords then

         for each lemmaRelation in sense.Relations do

            if cloudScale == shallowCloud then
               if lemmaRelation.Type == Derivation or
                        lemmaRelation.Type == SeeAlso then
                  if lemmaRelation.Lemma2 not ∈ cloud then
                     add lemmaRelation.Lemma2 to cloud

            if cloudScale == deepCloud then
               if lemmaRelation.Type == Derivation or
                        lemmaRelation.Type == Hypernym or
                        lemmaRelation.Type == InstanceHypernym or
                        lemmaRelation.Type == SeeAlso or
                        lemmaRelation.Type == AdjectiveSimilar or
                        lemmaRelation.Type == AdjectiveParticle or
                        lemmaRelation.Type == AdjectiveClusterMember or
                        lemmaRelation.Type == AdjectiveClusterHead then
                  if lemmaRelation.Lemma2 not ∈ cloud then
                     add lemmaRelation.Lemma2 to cloud

   return cloud

function CALCULATE-CLOUD-DISTANCE(cloud1, cloud2) returns the cloud distance
                                                                    data
   new CloudDistanceData: toReturn
   new DistanceItem: tmpDistanceData

   for each lemma1 in cloud1 do
      for each lemma2 in cloud2 do
         tmpDistanceData.lemma1 ← lemma1
         tmpDistanceData.lemma2 ← lemma2
         tmpDistanceData.distance ← BIDIRECTIONAL-MPL(lemma1, lemma2)

         add tmpDistanceData to toReturn.distances

   toReturn.averageDistance ← CALCULATE-AVERAGE-
                                       DISTANCE(toReturn.distances)

   return toReturn

function CLOUD-DISTANCE-CALCULATOR(word1, word2)
   new list of lemma: cloud1
   new list of lemma: cloud2

   cloud1 ← FIND-CLOUD(word1)
   cloud2 ← FIND-CLOUD(word2)

   return CALCULATE-CLOUD-DISTANCE(cloud1, cloud2)
```

Figure 19: Cloud Distance Calculator algorithm

Part III

Implementation and experiments

"A theory can be proved by experiment but no path leads from experiment to the birth of a theory."

- Albert Einstein

Implementation Chapter 9

The application created to research the proposed hypothesis and to support future research in this area is called “NLP Affect Toolbox”. This toolbox holds a set basic of natural language processing tools and implements various analysis tools. The toolbox also contains a number of corpora, namely “Dictionary of Affect in Language (DAL)” (Whissell, 1989) and ConceptNet (Liu & Singh, 2004) and WordNet (Fellbaum, 1998).

The natural language processing is mostly done by third party software, namely Proxem Antelope (Proxem, 2008). Antelope is an acronym for Advanced Natural Language Object-oriented Processing Environment. With this environment the application is able to do part of speech tagging, chunking, parsing, deep dependency parsing and some semantic parsing processes. Antelope also has an implementation of an object-oriented lexicon.

All other selected tools, methods, processes and corpora are designed and implemented without the use of any third party software.

The actual design and implementation of the application will be described in the following paragraphs. In this description diagrams are used to easily give insight into the properties, methods and values of the class, structure or enumerator. In these diagrams only the important properties and methods are included. Methods to access private properties and properties that are only used locally by the class are not included to maintain a better overview.

First the various corpora are described that are implemented in the NLP Affect Toolbox. After this the implementation of the natural language processes are described. In the third paragraph the textual affect sensing classes are described. The last paragraph describes the graphical user interface of the system.

9.1 Corpora

9.1.1 WordNet

The implementation of WordNet in the application is an adaptation from the Antelope Lexicon. This lexicon is used by the Antelope parser to correctly parse natural language as described in chapter 4. The lexicon is an object-oriented version of the data found in WordNet version 3.0. Around the lexicon a (static) class is build, to easily access the information of the lexicon from anywhere in the application.

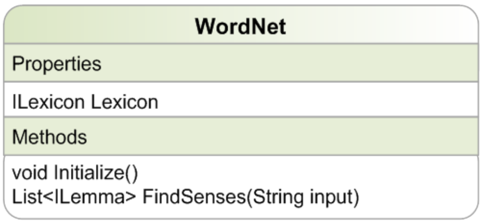

Figure 20: WordNet class

The WordNet class only has one property, the lexicon of the *ILexicon* type. The class has an Initialize method to load the lexicon into the memory for fast data access. The *FindSenses* method returns all senses of a particular input string (i.e. a word). This method returns a list of *ILemma*, which is a type used by Antelope. The structure of the lexicon and the types used will be explained below. Other important methods to access the data in the lexicon are available in the lexicon interface itself.

The lexicon application programming interface, as show in Figure 21, shows various interfaces. The main interface for the lexicon is the *ILexicon* interface. This interface has various methods to access the information stored in the lexicon. The data in the lexicon is structured as in WordNet in which words (i.e. lemmas) are grouped in synsets, as explained before.
The *ILemma* interface is used to access the properties of a single word. It has a method to find related lemmas. The possible relation types are the same as described in Table 10.

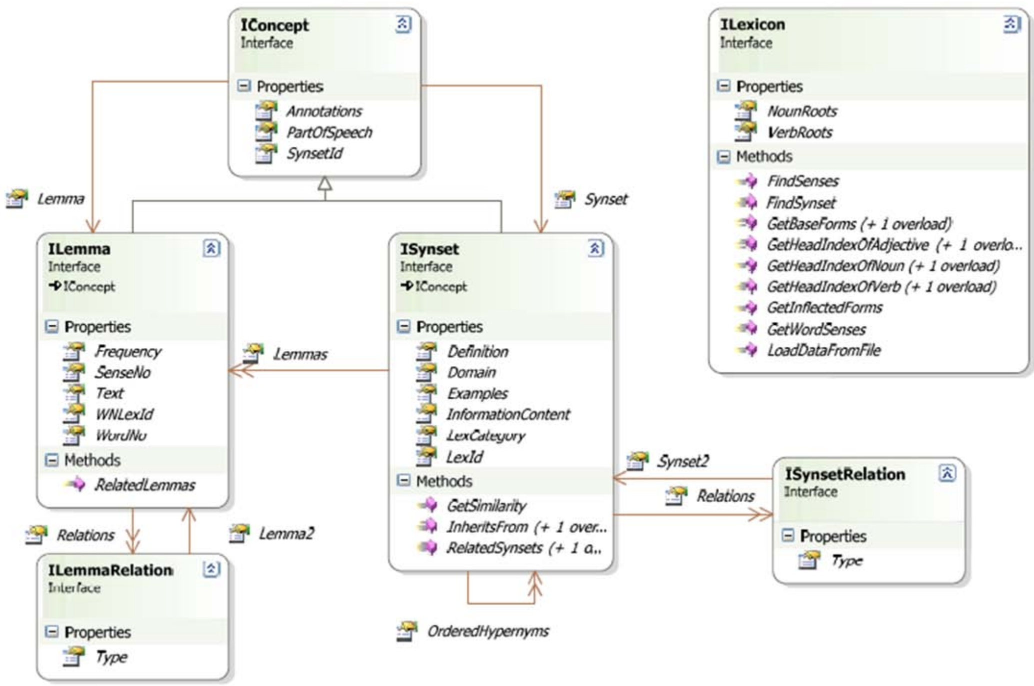

Figure 21: Lexicon API (Proxem, 2008)

9.1.2 Dictionary of Affect in Language

The Dictionary of Affect in Language (DAL) is also implemented in the toolbox. The following class implements all the methods to enable the application to load the DAL data into the memory and access it. The implementation of the DAL class is static, so it can be easily accessed throughout the application and is only instantiated once.

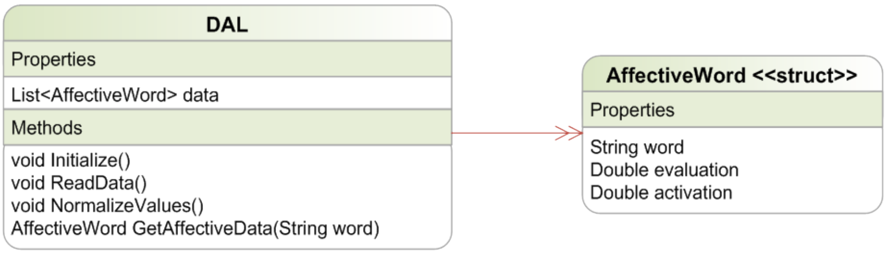

Figure 22: DAL class

As shown in the class diagram the data consists of a list of variables of the type "*AffectiveWord*". This type is implemented as a *structure*. The data is initialized by the *Initialize* method which uses the R*eadData* method, to read the data into the memory, and the *NormalizeValues* method to normalize the values on a scale of -1 to 1, respectively bad to good (evaluation) and passive to active (activation). There are two ways to access the data, one can use the *GetAffectiveData* method to access a particular *AffectiveWord*, or access the whole set of *AffectiveWords at once*.

The *AffectiveWord* structure as displayed above, has three variables. A *word* in the form of a string, a value for the evaluation (on a scale of -1 to 1) and a value for activation (also on a scale of -1 to 1).

9.1.3 ConceptNet

The implementation of ConceptNet, version 2.1, has been made by using the data used by the implementation of Liu & Singh in Python. The data is too extended to load into the memory of most machines, which has lead to the choice of keeping the data in text files as the implementation of Liu & Singh does. To speed up the process of accessing the data, the original data files are split and the data is grouped into different files according to the various possible relationships.

Table 11: ConceptNet relations

Relationship category	Relationship
Knowledge lines	Conceptually related to
	Thematic knowledge line
	Super thematic knowledge line
Thing	Is a
	Part of
	Property of
	Defined as
	Made of
Spatial	Location of
Events	Sub-event of
	Prerequisite event of
	First sub-event of
	Last sub-event of
Causal	Effect of
	Desirous effect of
Affective	Motivation of
	Desire of
Functional	Capable of receiving action
	Used for
Agents	Capable of

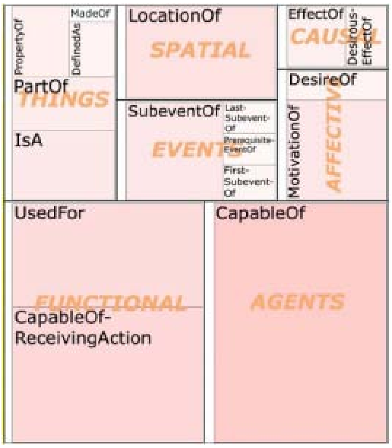

Figure 23: Grouped relationships of (Liu & Singh, 2004)

Every data file consists of lines that represent a relationship between two concepts. The form of a line from a data file is:

<concept1>;<concept2>;<frequency in original texts>;<frequency of inference>

So for example in the data file "C*apableOf.txt*" the following line occurs:

"accident;hurt;0;1"

This means that an *"accident"* is capable of *"hurt"*, this rule is not found in the original texts, but is inferred once.

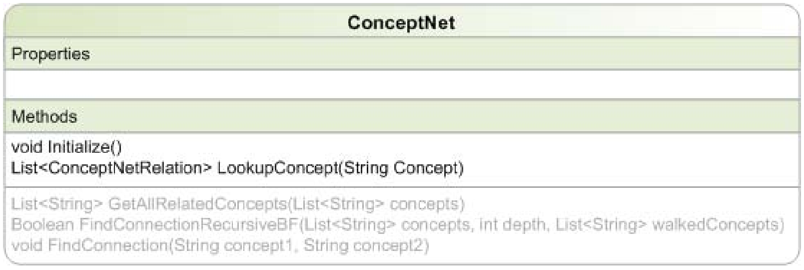

Figure 24: ConceptNet class

The ConceptNet class has only properties to specify in which files the search for concepts needs to be done and how the search needs to be done. Because of the extent of the relationships in ConceptNet and the ways of searching through them, these properties have been left out for clarity purposes.

The *Initialize* method sets up the class and opens the files that hold the data of the ConceptNet corpus. The *LookupConcept* method finds all related concepts of the input concept.
This class also implements a way of finding relations between two concepts by using the methods in gray. These methods are still very experimental and therefore drawn in gray.

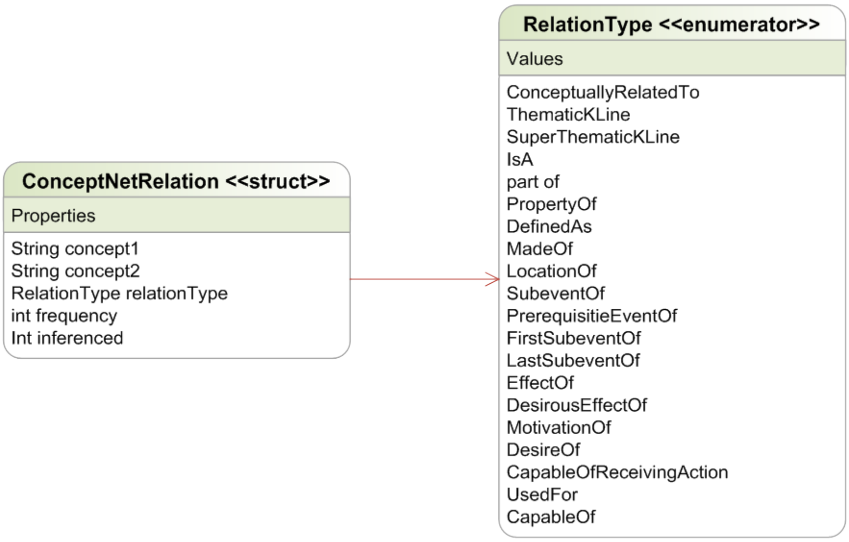

Figure 25: ConceptNetRelation

The *LookupConcept* method returns a list of *ConceptNetRelation*, this is a structure in which a single relation of ConceptNet can be hold. This structure consists of two strings (i.e. concepts), a *relationType*, an integer for the frequency and the number of times inferred as explained before.

9.2 Natural Language Processing

As said before most of the natural language processing has been implemented by using third-party software (Proxem). The NLP that will be done by Proxem Antelope will be described in this paragraph.

Proxem has developed an interface to process whole documents. This interface combines most available natural language processes as shown in the following figure.

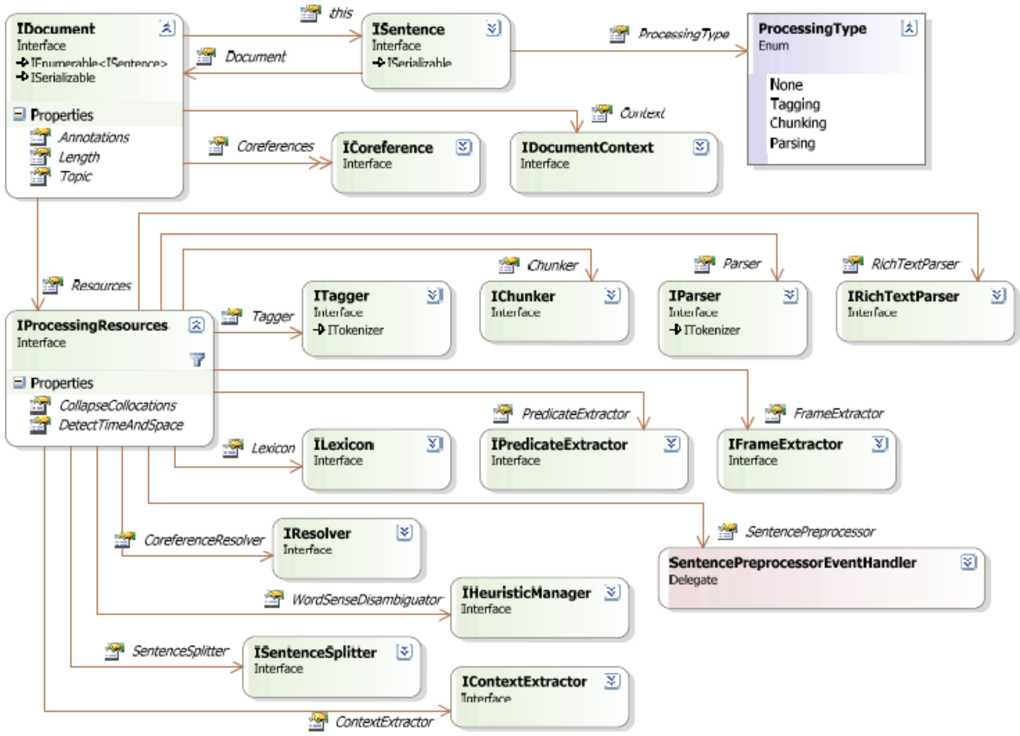

Figure 26: Document processing API (Proxem, 2008)

This interface is used to do most of the natural language processing available in the NLP Affect toolbox. The *IProcessingResources* interface is, as can be seen from the number of relations to the other interfaces, the most important interface. This interface holds the set of resources that are used to process the text of a document.

For the implementation of this API in the NLP Affect toolbox a class is created, named “Antelope” because it is an extension of the Proxem Antelope environment, which is responsible for all natural language parsing. This class makes use of the Proxem Antelope environment, but is **not** a part of it and is written to do the basic NLP processes as described before, in which it uses the API of Proxem.

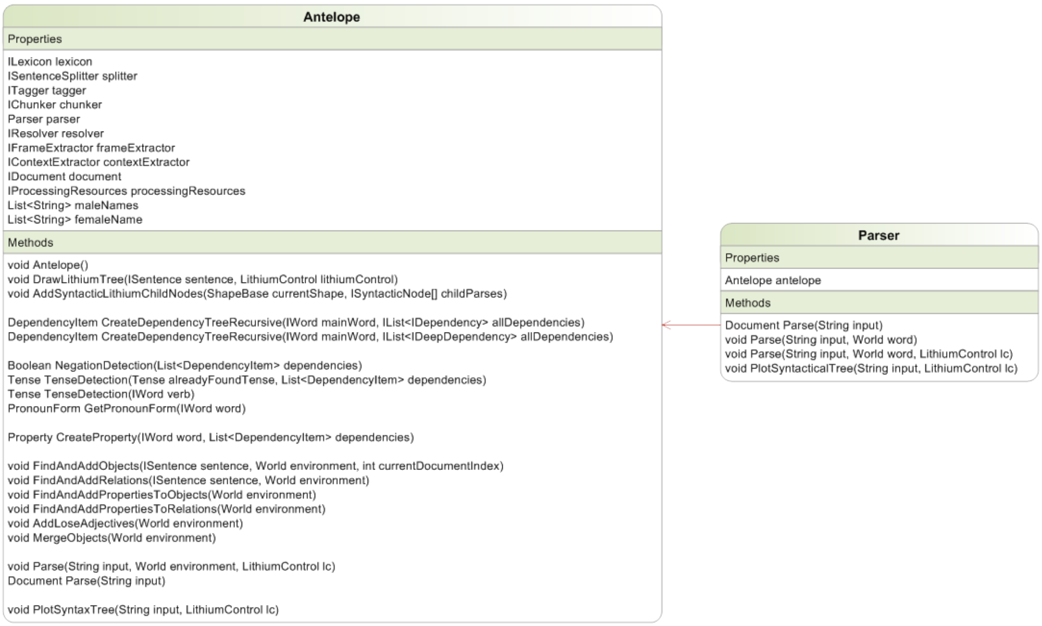

Figure 27: Antelope & Parser class

As shown in the figure above, the Antelope class holds a lot of properties and methods necessary for all natural language processing. A Parser class has been created as a static wrapper for the Antelope class, so the parser is only instantiated once and can be easily accessed throughout the application. The parser class is the “interface” for the NLP Affect toolbox. It provides three parse methods which can be used to parse a document; to parse a document and create a world model (explained in paragraph 9.3.3); or to parse a document, create a world model and create a syntax tree. The parser class also has implemented a method that only creates a syntax tree.

The instance of the Antelope class can be accessed through the Parser class. In this way, all natural language processing tools can be accessed and used separately.

Most methods in the Antelope class are used to parse a text into a world model. The concept of a world model and the methods used to create one will be explained in paragraph 9.3.3.

9.3 Textual affect sensing

9.3.1 WordNet lexical minimal path length

As suggested by (Kamps & Marx, 2001), we can use the lexical relations in WordNet to measure the evaluation and activation of a word. The MPL class implements this theory by using a breadth-first or a depth-first search approach as explained in paragraph 8.3.1. The reason to implement both search algorithms is because the breadth-first is a fast algorithm to calculate the evaluation and activation fast, and the depth-first algorithm is able to show us the actual path between two words.

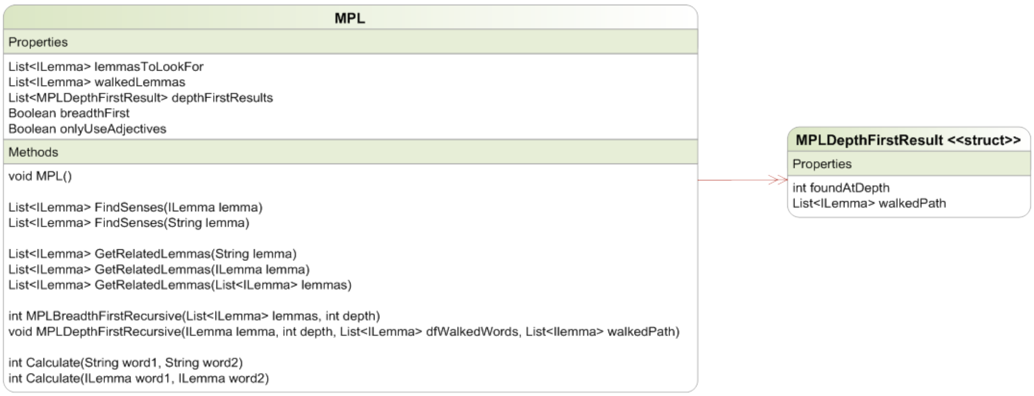

Figure 28: MPL class

There are two public methods in this class, which can be used to calculate the MPL namely, the *Calculate(String word1, String word2)* and the *Calculate(ILemma word1, ILemma word2)* method. The difference between the two is the type of input variables. The first method will find a minimal path between all senses of the input words and the second will find a minimal path between two senses of the input words. Because most of the time the application does not know which sense is meant, the first method is used more often. But if we know which sense is meant the outcome of the MPL can be different and might be more accurate.

Both methods use either the *MPLBreadthFirstRecursive* or the *MPLDepthFirstRecursive* method to calculate the MPL. As indicated by the methods names both the breadth-first as the depth-first algorithm are implemented recursively and thus will call themselves until the minimal path is found or all paths has been walked. Because WordNet contains a lot of circularity in its graph, the need to maintain a list of walked paths or walked lemmas is needed.

The *GetRelatedLemmas* methods are used to find out to which lemmas a lemma (or a list of lemmas, for the breadth-first algorithm) is related to. This can be seen as finding out which edges are connected to the node (i.e. lemma).

When the depth-first algorithm is used, a list of results is kept in the form of the *MPLDepthFirstResult* structure. This list contains all possible paths that exist between the two given words.

Because (Kamps & Marx, 2001) proposed in their theory to only use adjectives; the MPL can be calculated by only using adjective word senses or just use all possible senses. Experiments have been done by using both possibilities to see what the difference is between them, this will be put forward in chapter 10.

9.3.2 Cloud distance

To calculate the cloud distance of a particular word the following classes and structures have been set up.

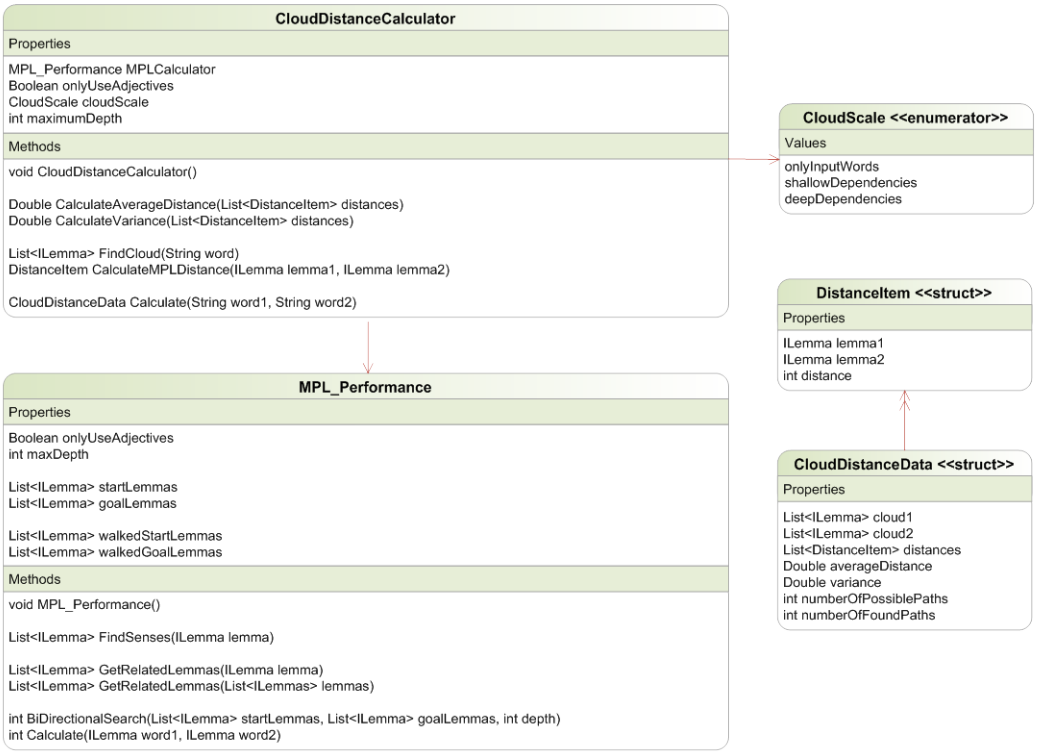

Figure 29: CloudDistanceCalculator class

The *CloudDistanceCalculator* class is the main class to calculate a cloud distance between two words. It uses the *MPL_Performance* class to calculate a minimal path length by using the bidirectional search algorithm, as explained in paragraph 8.3. Because a cloud distance is an average of NM number of normal minimal path lengths, in which N and M are respectively the number of words in the starting and goal cloud, we need to calculate these paths as fast as possible, thus by using the bidirectional search algorithm. The way of using the *MPL_Performance* class is the same as for the normal *MPL* class, it only applies a different algorithm.

Again the *CloudDistanceCalculator* can calculate the MPL by only using adjectives and by using all words. A second differentiation is used to support three kinds of clouds, namely: "*onlyInputWords*", "*shallowDependencies*", "*deepDependencies*". The "*onlyInputWords*" cloud calculates the average distance between all senses of both words. The difference between the "*shallowDependencies*" and the "*deepDependencies*" clouds are in the variety of relations they use to find related words, as specified in Table 10. These groups of related words are called respectively "shallow cloud" and "deep cloud".

To cope with the circularity in the WordNet graph, again a list of walked lemmas is maintained for both directions of the search in the *MPL_Performance* class.

The *Calculate* method returns a structure of the type *CloudDistanceData* which holds all information about the calculated cloud distance. The *CloudDistanceData* contains a list of words found for the starting cloud and for the goal cloud; a list of maximal NM number of distances of type *DistanceItem*; the average distance; the variance over the calculated distances; the number of possible paths and the number of found paths.

9.3.3 Deep semantic parsing

Deep semantic parsing can be seen as extracting all meanings or knowledge put forward in a text. This can be seen as sketching a picture of what is told in the text. In the NLP Affect toolbox we call this sketch the world model.
Extracting a world model that is described in a text is a somewhat new approach to textual affect sensing. As (Liu, Lieberman, & Selker, 2003) have proposed, it is necessary to understand the semantics of a text, so a reasoning can be made about the feelings or emotions the writer has. To be able to reason about the semantics, the semantics first need to be extracted. An experimental set up to do this has been made in the NLP Affect toolbox.
The object model by which the meaning or knowledge is described is the following.

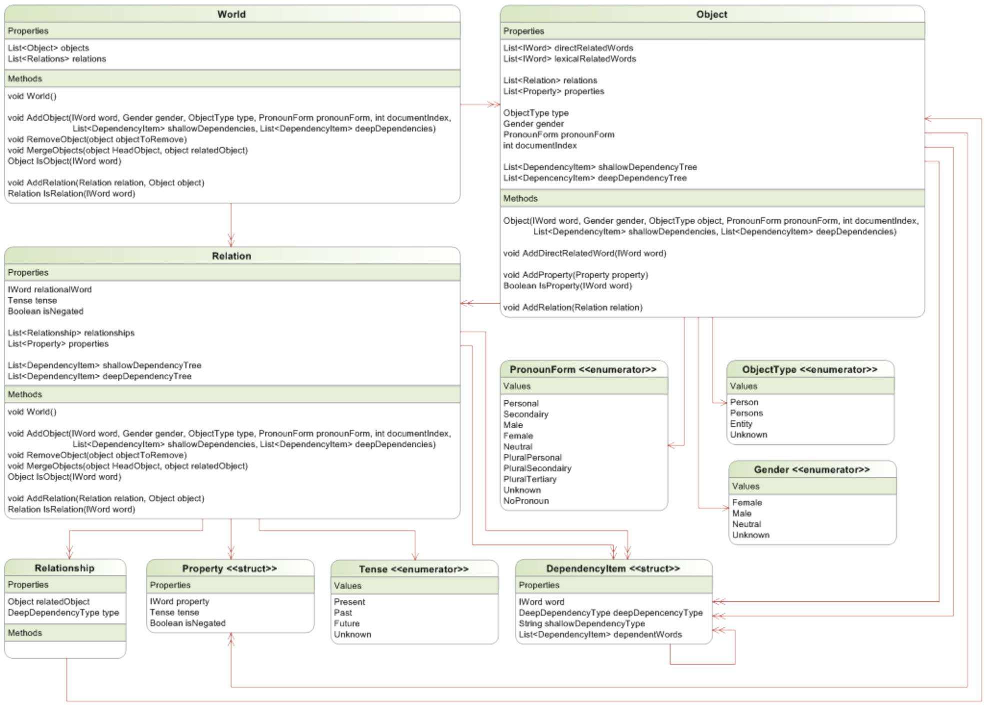

Figure 30: World model

This object model can be seen as the data of the world model, the methods used in the classes are only there to mutate the data which is hold by the classes. As said in paragraph 9.2, the Antelope class holds all methods to extract the semantics that are stored in the world model. These various methods are described below, these methods have all been newly written to extract semantics from text and to create the world model.
When the Antelope class begins to parse the document into a world model, it starts by finding and adding objects. The *FindAndAddObjects* method filters out the pronouns (e.g. I, he, she, it, we, etc.) and nouns. For each object the method tries to set the "fixed" properties (i.e. type, gender, pronounform). For pronouns it will first call the method *GetPronounForm* which is able to differentiate pronouns into the given *PronounForms* of the enumerator by using the following table.

Table 12: Pronoun forms

PronounForm	Pronouns	PronounForm	Pronouns
Personal	I, me, my, mine, myself	PluralPersonal	we, us, our, ours, ourselves
Secondary	you, your, yours, yourself	PluralSecondairy	you, your, yours, yourselves
Female	she, her, hers, herself	PluralTertiary	they, them, their, theirs, themselves
Male	he, him, his, himself		
Neutral	it, its, itself		

If the object found is a noun the *pronounForm* property will be set to “*NoPronoun*”. When the pronoun form has been set, the two other “fixed” properties for pronouns can be set easily by using the following rules.

Table 13: Fixed property rules for Object

If pronoun form is	Then
Personal	Gender is neutral ObjectType is person
Secondary	Gender is neutral ObjectType is person
Female	Gender is female ObjectType is person
Male	Gender is male ObjectType is person
Neutral	Gender is neutral ObjectType is entity
PluralPersonal	Gender is neutral ObjectType is persons
PluralSecondairy	Gender is neutral ObjectType is persons
PluralTertiary	Gender is neutral ObjectType is persons

If the object found is a noun the two other “fixed” properties are set in a different way. A noun can be a name of a person. The Antelope class has two lists of names, one for female names and one for male names. So we check if the noun is a female or male name. If so, the gender property can be set to either male or female and the type can be set to person. Otherwise the gender is set to neutral and the type is set to entity.
After this step a dependency tree is created using the Antelope shallow parsed dependencies and a dependency tree is created using the Antelope deep parsed dependencies. These dependencies can be seen as related words that say something about the object.
Also the *documentIndex* is set in this method, this is an integer value to the position of the word, that represents the object, in the document.

After the objects are found and added, the relations between the objects will be found and added by the *FindAndAddRelations* method. This method uses the deep syntax dependencies created by the Proxem Antelope parser to find relations between objects. The general idea behind this concept is that a verb acts as a relation between two objects. For example in the following sentence:

“The man is driving a new car”

This sentence has two objects, namely: “man” and “car”, these two objects are related by the verb phrase “is driving”. So a relation is created with the relational word “driving” between the head object, in this case “man”, and the dependent object, in this case “car”. This relation will get two relationships, the first is to the object “man” with type “subject” and the other is to the object “car” with type “*directObject*”. These relationship types are conform the Proxem *DeepDependencyType* type as described in appendix C.
A relation also has some “fixed” properties, namely: *tense* and *isNegated*. Because relations are actually verbs a tense can say a lot about the relation. The tense of a relation is extracted by using the *TenseDetection* method. This method is first being called with as input the relational word (i.e. the verb) of the relation. By checking the tag given to the word we can easily see in which tense that word is being used. Unfortunately a verb can be part of a verb phrase in which the tense can be modified by other verbs used in the verb phrase. If the *TenseDetection* method returns a tense other than “past” all dependencies of verb need to be checked for their tense. The main rule used here is, when one of the dependent verbs is using the past tense then the whole verb phrase is considered to be in the past tense.

A relation is also tested for negation, this is simply done by checking the dependencies of a relational word for the shallow dependency type or tag (as described in appendix C) “*neg*” recursively.

After all objects and relations are added, the properties of the objects and relations are added by calling the *FindAndAddPropertiesToObjects* and *FindAndAddPropertiesToRelations* methods. The first method find properties for the objects found before, by finding dependent words with the following tags:

Table 14: Tags for properties of objects

Tag	Description
JJ	Adjective or numeral, ordinal
JJR	Adjective, comparative
JJS	Adjective, superlative
IN	Subordinating

These four types of words say something about a noun or pronoun. The second method tries to find properties for the relations, by finding dependent words with the following tags:

Table 15: Tags for properties of relations

Tag	Description
RB	Adverb
RBR	Adverb, comparative
RBS	Adverb, superlative
IN	Subordinating
RP	Particle

These five types of words say something about (or modify the meaning of) verbs.

Sometimes adjectives (properties) in a sentence are not coupled to nouns (or objects), this is why the *AddLoseAdjectives* method is created. This will go through every word of the sentence and checks if every adjective is coupled to a object. If an adjective is found that has not been coupled to an object it will find out to which object it is related and add it as a property of that object.

The *MergeObjects* method is currently not used. This method should merge objects that refer to the same object, called co-references. Unfortunately the co-reference resolver of Proxem Antelope is still in experimental phase and has a poor quality in finding the actual co-references.

9.4 Graphical user interface

For users that do not have a technical background, and thus cannot use the NLP Affect toolbox as a programming library for research in the field of natural language processing or textual affect sensing, the toolbox has a graphical user interface. Which holds various tools that can be used for anyone interested in this field. The tools give the user a better insight in the structure and relations used in the corpora and theories discussed in this document. The various tools are described below on basis of their graphical user interface.

The general interface of the toolbox, as shown below, consists of a menu and a log. The log is used by several tools to show the user what happens in the processing.

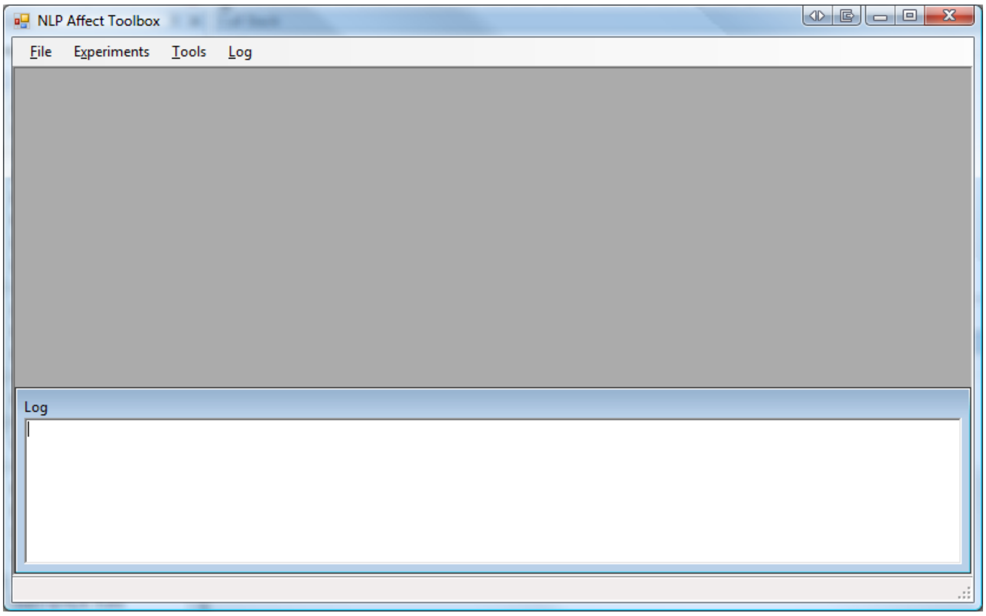

Figure 31: NLP Affect Toolbox - GUI overview

In the menu the user can chose which tool or experiment he wants to use. The menu has the following structure.

- File
    - Exit
- Experiments
    - World extraction
    - Cloud distance
- Tools
    - WordNet
        - MPL calculator
        - Cloud distance calculator
        - Browser
    - Dictionary of Affect in Language
        - Browser
    - ConceptNet
        - Browser
    - Sentence parsing
        - Syntactical tree plotter
        - Semantic parser
    - Activation Evaluation plotter
- Log
    - Clear log

This structure can change depending on the tool that is used. The various tools are described in the following paragraphs.

9.4.1 WordNet – MPL calculator

The MPL calculator can be used to calculate the minimal path length between two words, as proposed by (Kamps & Marx, 2001). A starting and goal word must be given. In the settings box the user can chose between two search algorithms. The depth-first search algorithm will walk every possible path between two words and will result in a list of all possible paths and of course a minimal path length. The breadth-first algorithm will only result in a minimal path length.

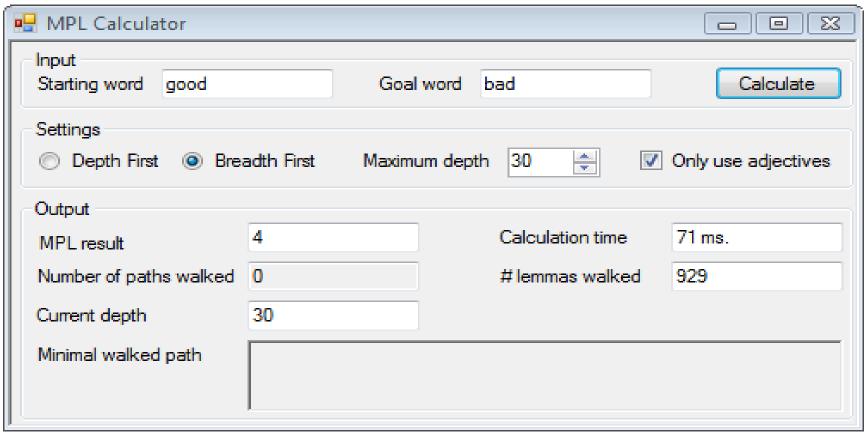

Figure 32: MPL Calculator GUI

9.4.2 WordNet – cloud distance calculator

The cloud distance calculator calculates the average cloud distance and its variance of the start and goal clouds. These clouds are created by using the given two words and the scale of the to be used clouds. The results is given as an average MPL, variance, number of possible paths and the number of found paths.

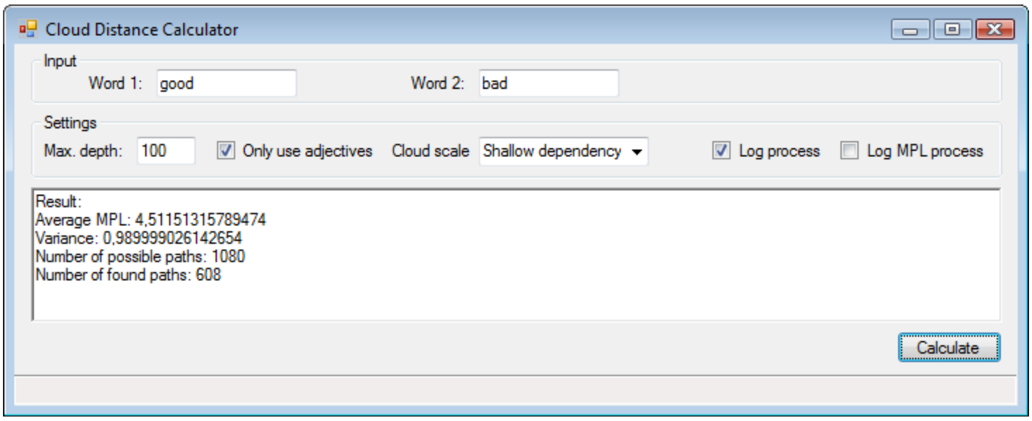

Figure 33: Cloud Distance Calculator GUI

9.4.3 WordNet – browser

The WordNet browser has been made to give a better insight in the structure held by WordNet. The user can input a word to lookup and differentiate which form of the word need to be shown.

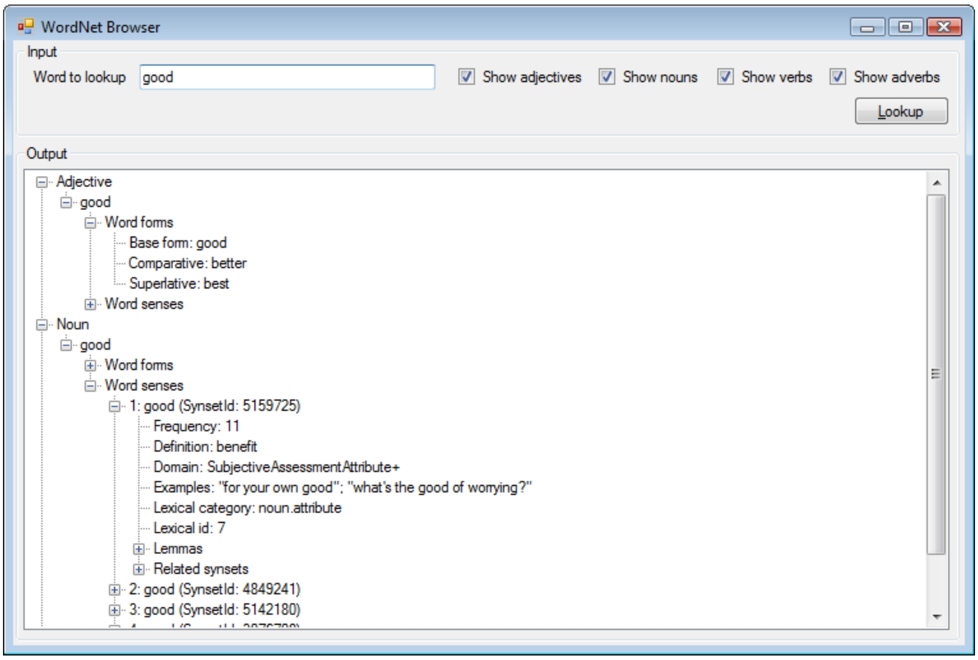

Figure 34: WordNet Browser GUI

The lookup results in a tree of the structure of WordNet.

9.4.4 DAL – browser

The DAL browser is a tool in which the evaluation and activation of a word can be looked up. If a word is not found in the DAL set the values are 0.

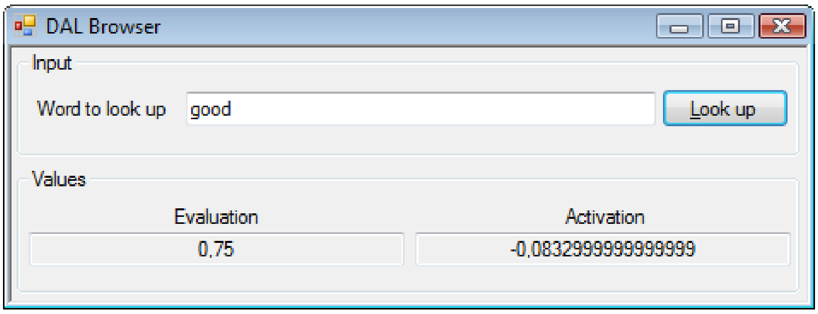

Figure 35: DAL Browser GUI

9.4.5 ConceptNet – browser

The ConceptNet browser enables users to lookup a concept and see all related concepts of a particular relation type.

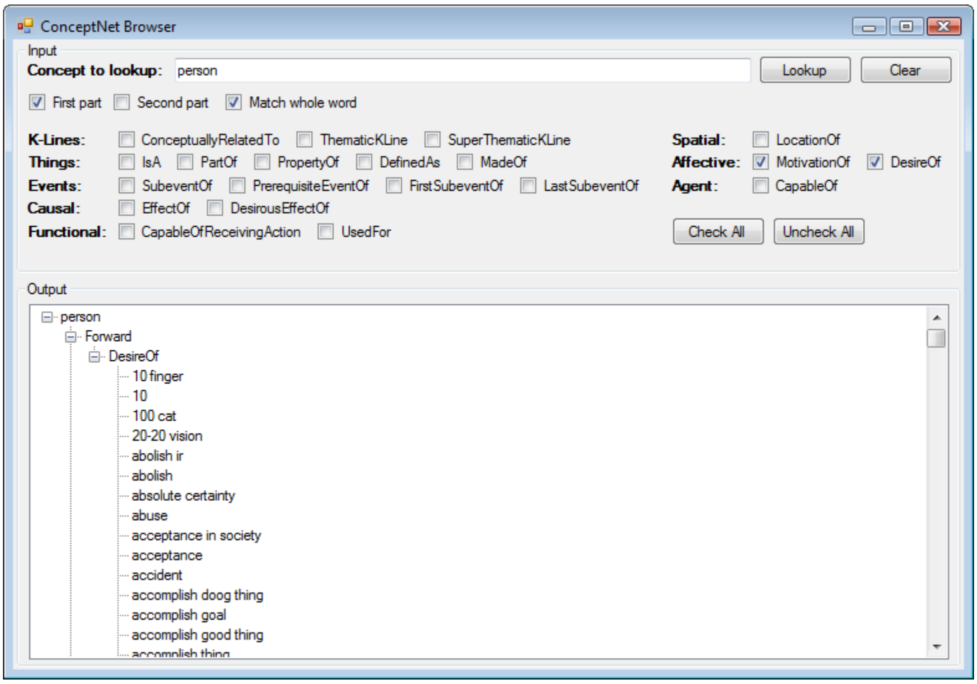

Figure 36: ConceptNet Browser GUI

9.4.6 Sentence parsing – Syntactical tree plotter

The syntactical tree plotter can plot a syntactical tree of any sentence inputted by the user.

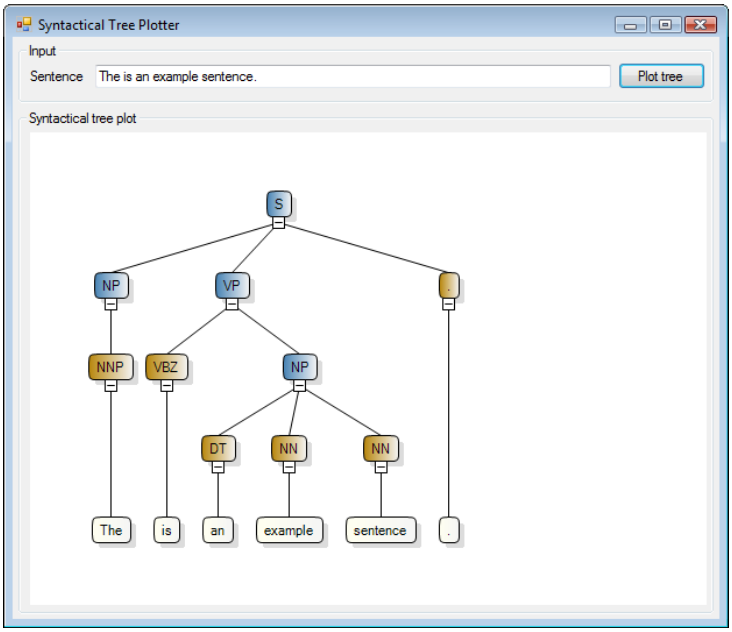

Figure 37: Syntactical Tree Plotter GUI

9.4.7 Sentence parsing – Semantic Parser

The semantic parser tool shows the abilities of the Proxem Antelope parser. The first three tab sheets show the syntactical parsed sentences in three different forms. The last three tab sheets show the semantic abilities.

In the screen below the document or input is split into sentences and chunked into words. These words have been tagged for their position of speech. The list of possible tags can be found in appendix A.

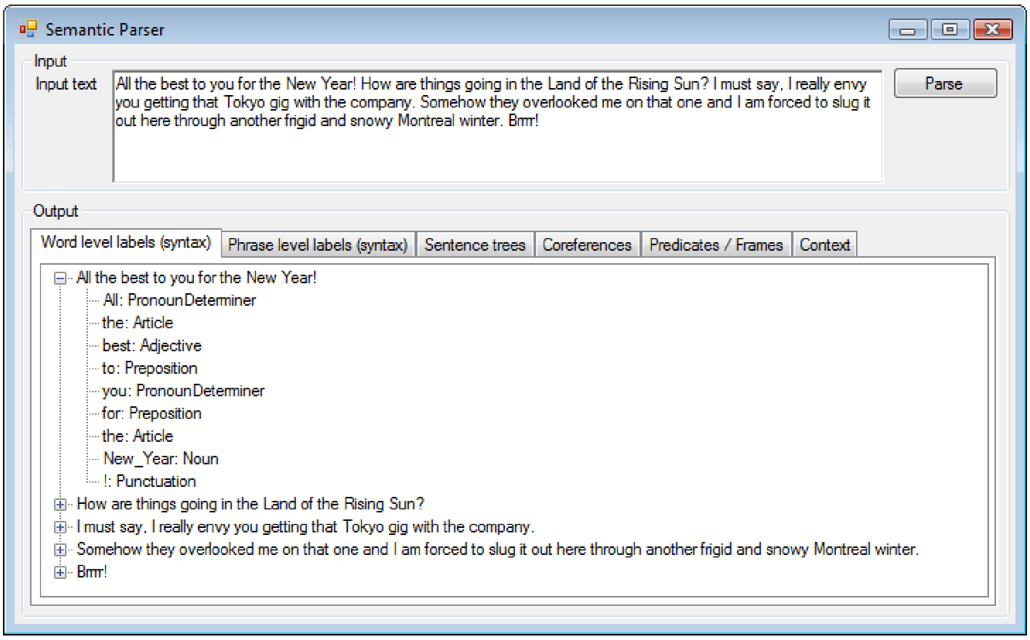

Figure 38: Semantic Parser GUI - Word level label

In the screen below the sentences are split into phrases and the phrases are tagged according to the type of phrase. The list of possible phrase types can be found in Appendix B.

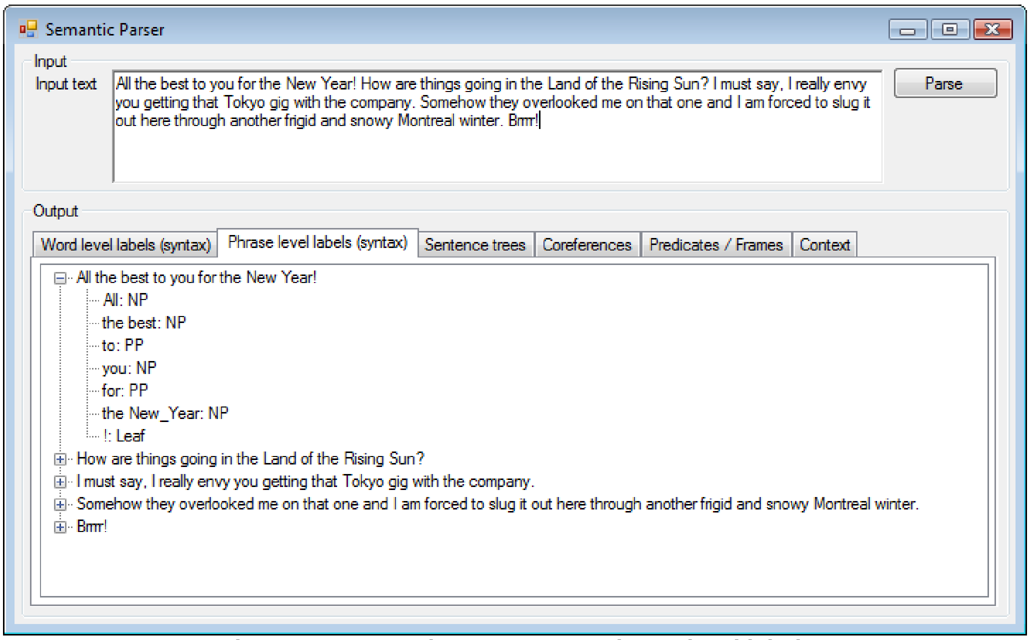

Figure 39: Semantic Parser GUI - Phrase level labels

The third tab sheet combines the information shown on the first two tab sheets by drawing a syntactical tree for each sentence. This tree shows the structure of the phrases in the sentences and the position of speech in the phrases.

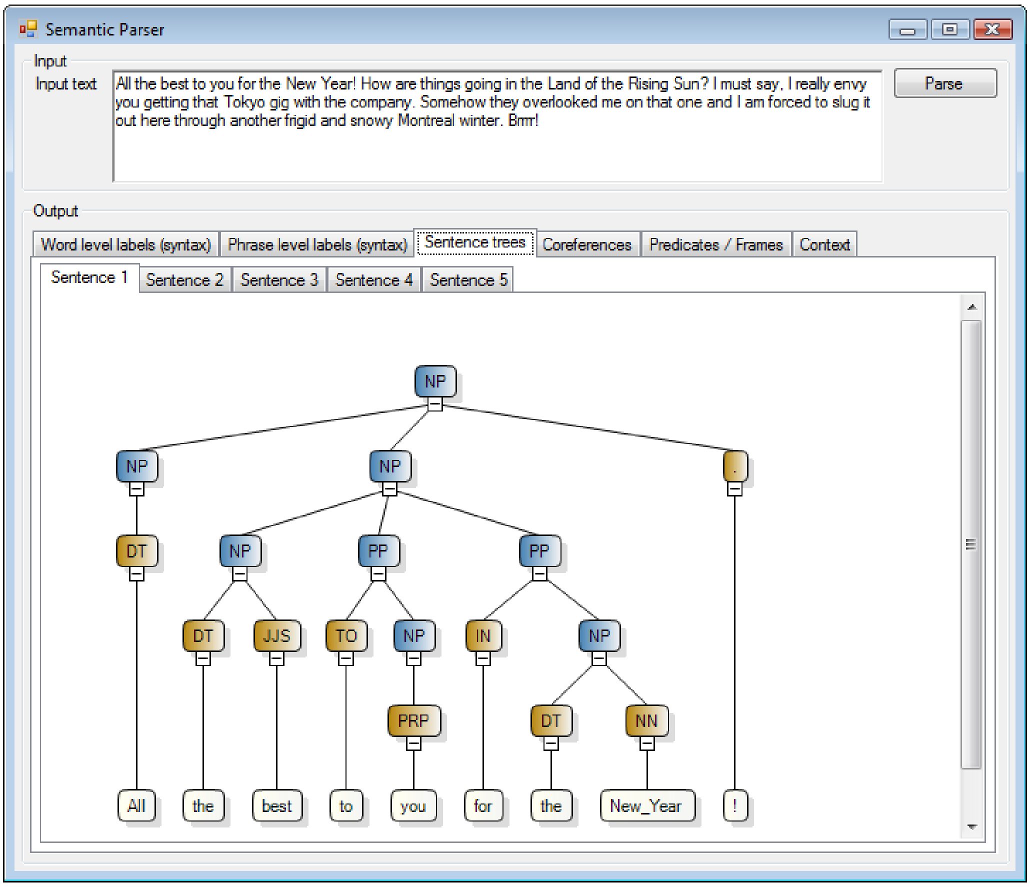

Figure 40: Semantic Parser GUI - Sentence trees

On this tab sheet the co-references are shown. As said before the co-references parsed by the Proxem Antelope environment is still experimental. The results as shown below are still of very poor quality. But it gives an insight in how Proxem handles the co-references.

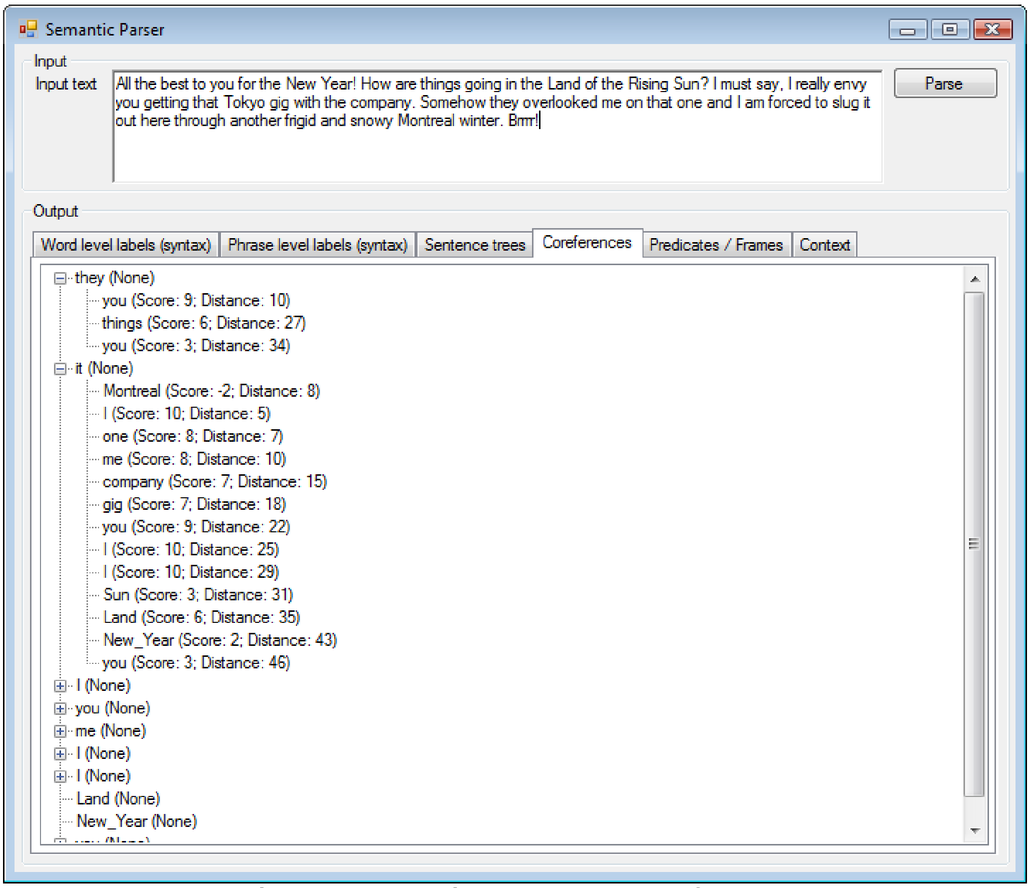

Figure 41: Semantic Parser GUI – Co-references

The tab sheet predicates and frames shows which predicates and frames are extracted by using the Proxem Antelope environment. The roles used in the predicates are also shown here. The various types of roles is equal to the table “Deep semantic dependency types” as described in appendix C.

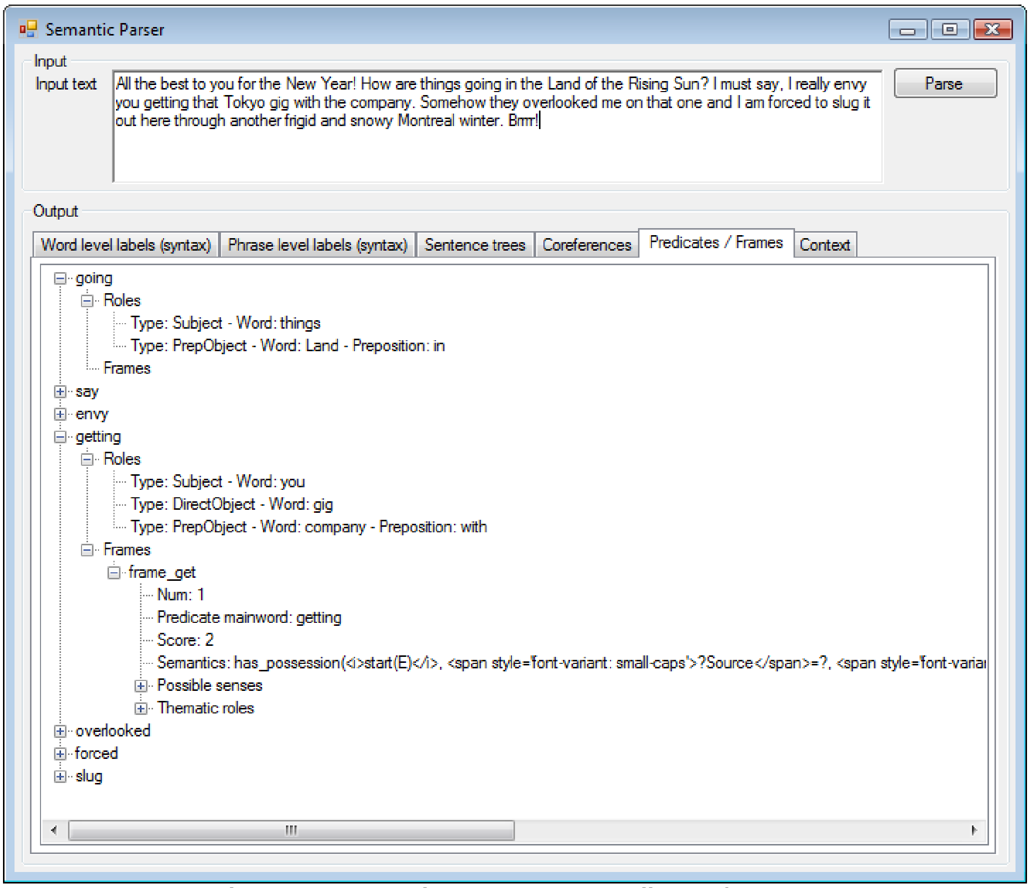

Figure 42: Semantic Parser GUI - Predicates / Frames

The last tab sheet shows the results of the context extractor of the Proxem Antelope environment. As can be seen from the results below, this also still is very experimental.

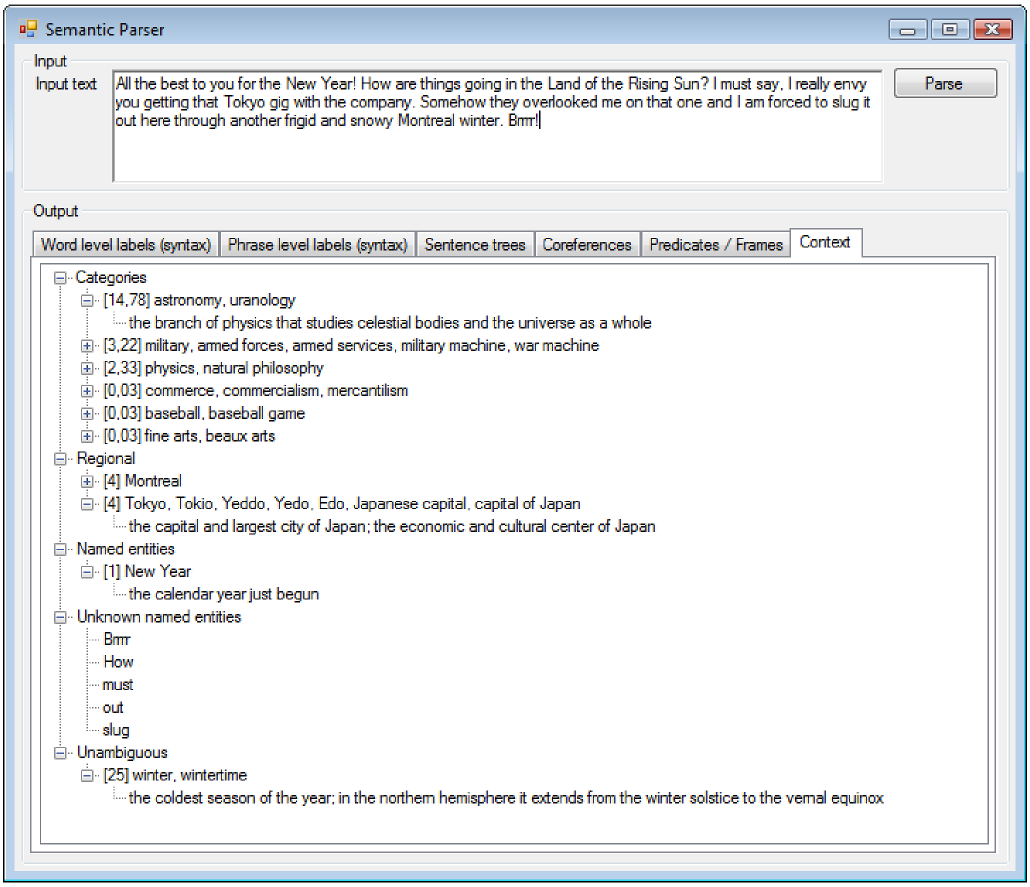

Figure 43: Semantic Parser GUI – Context

9.4.8 Activation – evaluation plotter

The activation – evaluation plotter is a very useful tool to quickly get insight in the quality of calculated activation and evaluation values. This data can be easily imported via a semicolon separated values list. The tool has three kinds of backgrounds to easily compare the values with commonly used circumplexes of affect. These backgrounds can be selected from the menu.

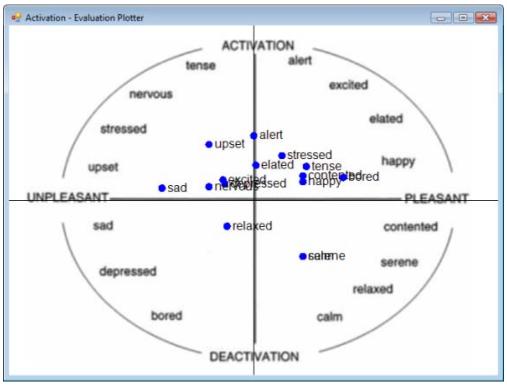

Figure 44: Activation - Evaluation Plotter using Russell's circumplex

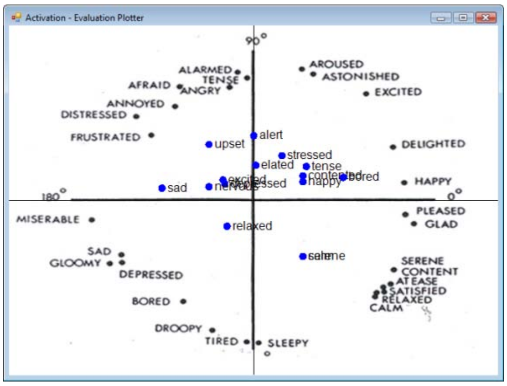

Figure 45: Activation - Evaluation Plotter GUI using Altarriba's plot

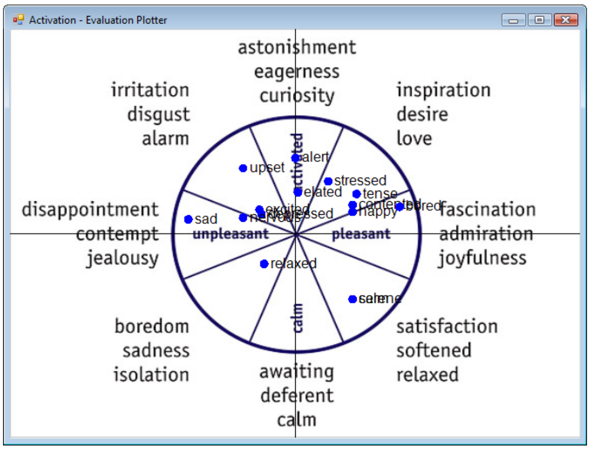

Figure 46: Activation - Evaluation Plotter GUI using Desmet's circumplex

9.4.9 Experiments – World extraction

The world extraction experiment has a graphical user interface as show below. In this interface the user can input one or more sentences, which will be parsed into a world model. The world model will then be shown in the form of a tree. The tree contains all objects and relation with their fixed and variable properties. For relations also the related objects are shown.

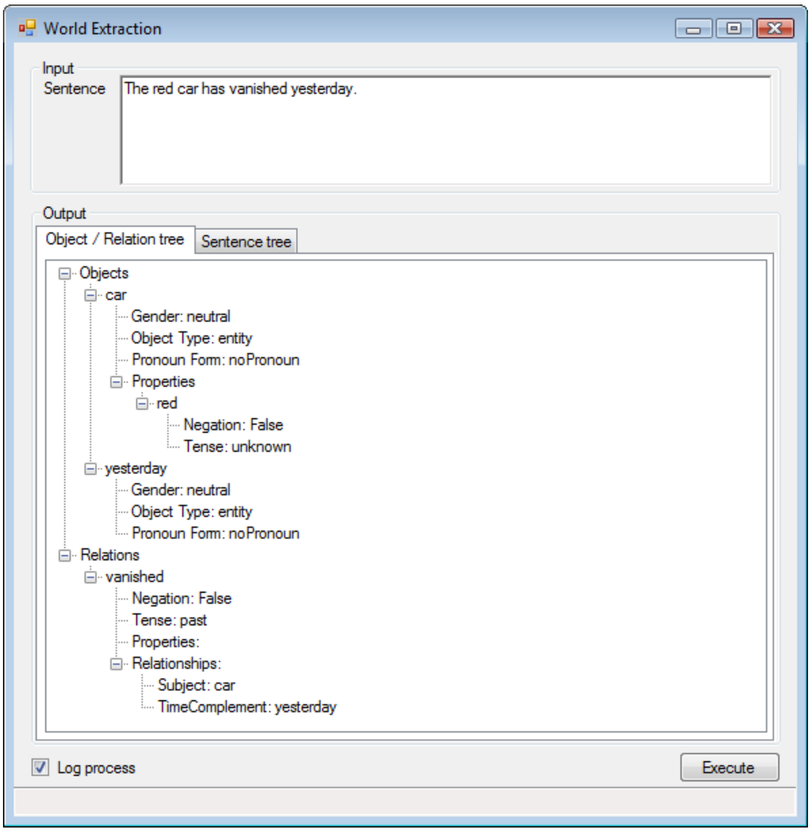

Figure 47: World Extraction GUI

9.4.10 Experiments – Cloud distance (multi-threaded)

For the cloud distance experiments a lot of calculations needed to be done as fast as possible. The implementation already is using the fastest algorithm (i.e. bidirectional search). To speed up the process even more, a multi threaded version of the cloud distance calculator has been made. Because modern computers have multiple cores in their processors, they can do multiple tasks at the same time. By creating multiple threads, the operating system balances the load evenly over all cores. In this way the process can be (at the current state-of-art) four times faster.

The cloud distance experiment uses a semicolon separated values file to queue up all calculations that are needed to be done. The same settings can be applied, as in the normal cloud calculator, plus the number of threads to be used.

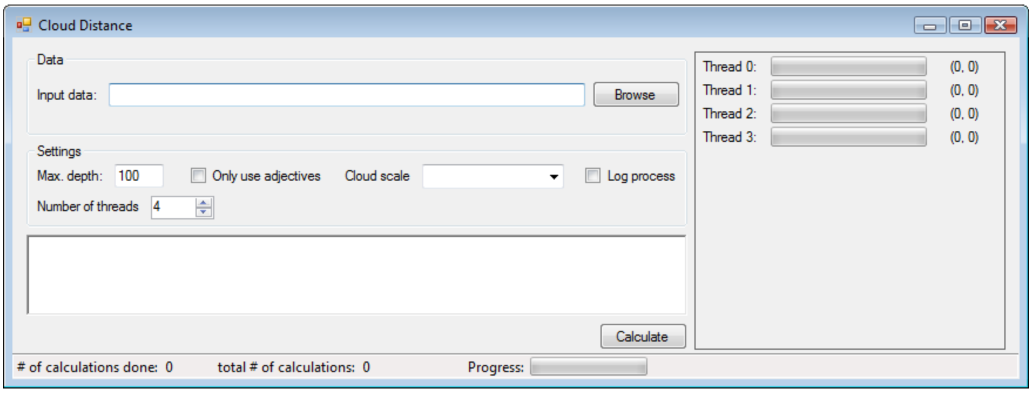

Figure 48: Multi-threaded Cloud distance GUI

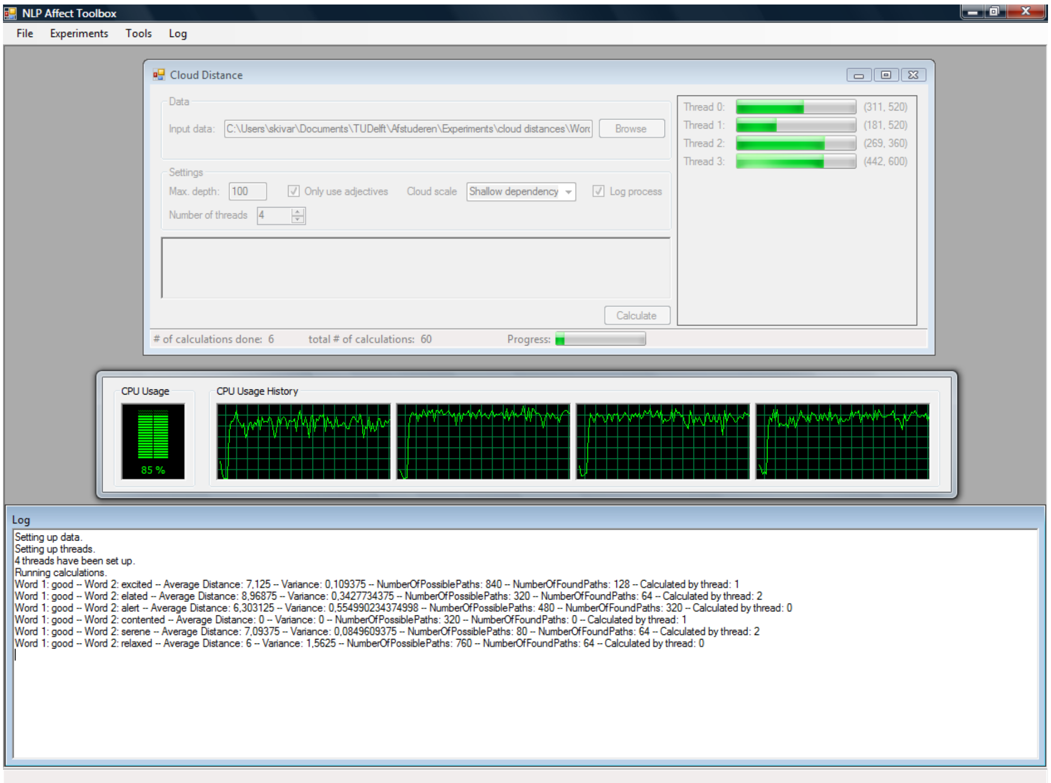

Figure 49: Multi-threaded Cloud Distance GUI in action

Experiments Chapter 10

In this chapter the experiments and the results of these experiments are described. The experiments done are based on the defined questions, described in paragraph 8.2. They are done to investigate the quality, from a numerical and classification point of view, of the MPL method as proposed by (Kamps & Marx, 2001) and the novel shallow and deep cloud methods. These new methods have been proposed in the first place to increase the number of words for which the activation and evaluation can be calculated.

The experiments are in the form of calculating the values for words using the different methods, and comparing them with hand annotated sets. To know in which degree the methods may differ from the hand annotated sets, we first need to investigate in which degree the hand annotated sets differ from each other. After this the proposed methods will be compared with the existing annotated sets to answer the proposed research questions described in paragraph 8.2. All word sets used can be found in appendix D.

10.1 Deviation of hand annotated sets

Hand annotated sets of words are scarcely found. At this time the only corpus of hand annotated activation and evaluation values for words, for us available, is the Dictionary of Affect in Language (DAL) (Whissell, 1989). To overcome this problem a selection of figures of circumplexes of affect have been gathered from which the values for activation and evaluation can be derived.

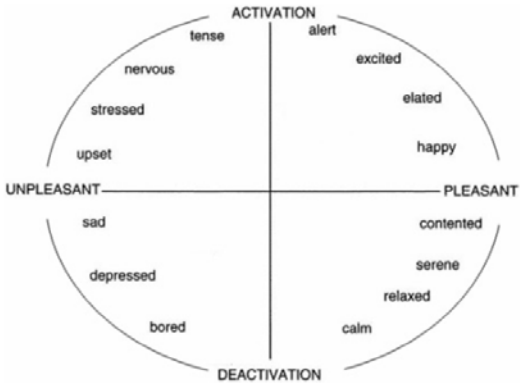

Figure 50: Circumplex of affect (Russell & Barrett, 1999)

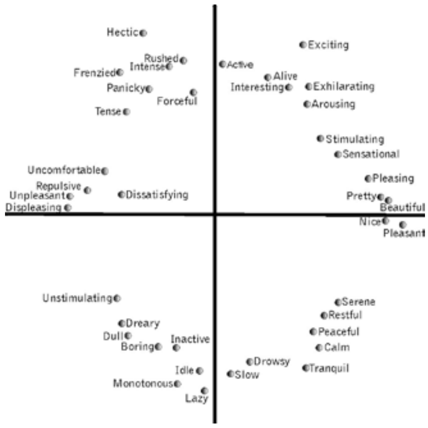

Figure 51: Circumplex of affect (Russell & Lanius, 1984)

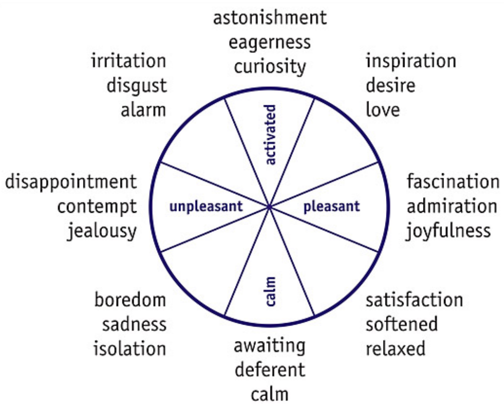

Figure 52: Circumplex of affect (Desmet, 2002)

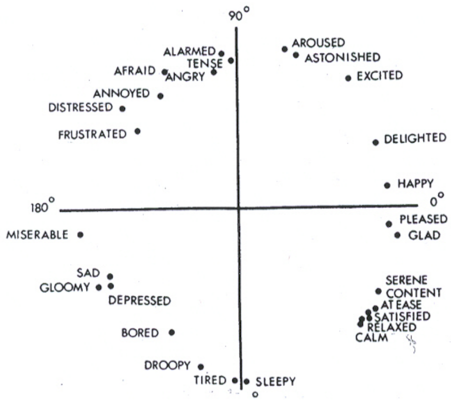

Figure 53: Circumplex of affect (Altarriba, Basnight, & Canary, 2003)

Because, as can be seen from the figures above, the nature of these figures differ from each other, they cannot easily be compared. The figure of (Russell & Barrett, 1999) contains words normalized to lay directly on the circumplex. (Desmet, 2002) depicts the words normalized on the circumplex and only shows a range in which these words lie. And the figures of (Russell & Lanius, 1984) and (Altarriba, Basnight, & Canary, 2003) show us actual values for the words, that creates a circular pattern on which the theory of the circumplex of affect is based.

To compare all word sets, the angle (in radians) for each word was extracted and compared to the angles of the words found in the DAL set. For all but the figure of (Desmet, 2002) this process is trivial. For the figure of (Desmet, 2002) the center angle of each range was used for all words in that range, and thus is less accurate.

To give more insight in the differences of the angles, a comparison has also been made between the normalized values for evaluation and activation. This normalization is done by calculating the values from the radian angle by using the sinus and cosinus functions. Because the theory of the circumplex of affect (Russell & Lanius, 1984) states that the words tend to form a circular pattern (as can be seen in the figures above), the assumption can be made that the difference between the values for activation and evaluation and the normalized values are minimal.

Because of the lag in overlap of the used words in the different figures, a comparison can only be made between the four word lists and the DAL list. All together 93 unique words, gathered from the four figures, are used to make the comparison. This is a somewhat small amount of words to generalize the quality of the DAL set (8700 words) with, but these words were chosen for their distinctiveness in the set of affective words by leading scholars and therefore can be seen as a good representation of the underlying classes of words.

Table 16: Numerical comparison of annotated word sets

	Normalized DAL evaluation deviation	**Normalized DAL activation deviation**	**DAL angle deviation**
Set 1: (Russell & Barrett, 1999)	12.03 %	12.73 %	11.87 %
Set 2: (Russell & Lanius, 1984)	17.26 %	19.48 %	19.13 %
Set 3: (Desmet, 2002)	8.76 %	14.77 %	11.87 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	18.70 %	18.87 %	18.43 %
Weighted average percentages	**15.25 %**	**17.19 %**	**16.28 %**

The table above depicts the percentages in which the word sets differ from the DAL set. The weighted average percentages show in what degree all word sets differ with the DAL set. The average in which the angles deviate is 16.28 %. This deviation is almost equally divided among the values for activation and evaluation, but the percentages for the activation are slightly higher.

To get more insight in what these percentages mean for the classification of words, we will compare the different sets on basis of the quadrant in which the words lay. Because the theories differ from time to time about which structure of classes should be used, we will compare all words on basis of three different structures, as seen in the figures below.

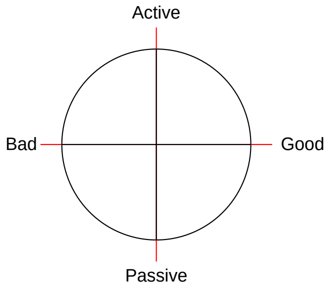

Figure 54: Normal activation - evaluation quadrants

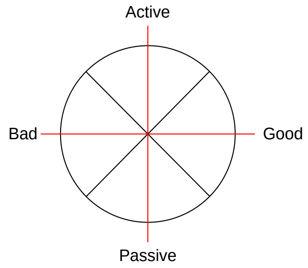

Figure 55: Quadrants rotated 45 degrees

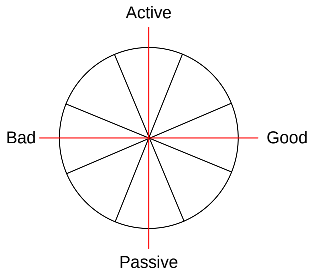

Figure 56: Octants as used by (Desmet, 2002)

In the table below the percentages of words, that both occur in the DAL set and in the designated set, that differ in quadrant or octant are shown. Sometimes words occur precisely on the boundary of a quadrant or octant. In this case nothing can be said about the quadrant or octant in which the word should be lying, these cases are omitted from the comparison.

Table 17: Classification comparison of annotated word sets

	% in diff. quadrant	**% in diff. 45° rotated quadrant**	**% in diff. octant**
Set 1: (Russell & Barrett, 1999)	7.69 %	33.33 %	36.36 %
Set 2: (Russell & Lanius, 1984)	45.45 %	40.91 %	68.18 %
Set 3: (Desmet, 2002)	14.29 %	0.00 %	58.33 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	18,18 %	45.45 %	54.55 %
Weighted average percentages	**25.00 %**	**37.70 %**	**56.72 %**

Even though DAL is a large corpus, not every word used in the four other sets could be found in this corpus. In the table below the percentages of words that could be found in the DAL set are shown.

Table 18: Number of words compared

	Total number of words	Number of words occurring in DAL	%
Set 1: (Russell & Barrett, 1999)	15	13	86.67 %
Set 2: (Russell & Lanius, 1984)	40	22	55.00 %
Set 3: (Desmet, 2002)	24	12	50.00 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	28	22	78.57 %
Total	**107**	**69**	**64.49 %**

The conclusion from the comparison made in the tables above, is that from the 69 key affective words compared the average difference found is about 16%. In classification terms there is a minimal difference of 25% (normal quadrants) and a maximum difference of 57% (octants). Of course nothing can be said about the correctness of any hand annotated word set from these percentages. But a clear margin of error has been set by this comparison and will be used as the margin of error for the proposed methods.

10.2 Minimal Path Length method using adjectives only

To examine the quality of the minimal path length method as proposed by (Kamps & Marx, 2001), a comparison will be made between the calculated and normalized MPL values and the 5 other sets of normalized values. Because this comparison is only between normalized values, it could give a slightly different result than when these values would not be normalized. For an even finer comparison, a comparison will also be made between the actual values of the MPL with the actual values of the DAL set.

To calculate all MPL values for all words in the DAL set is to extensive, thus all unique words of the four other lists that can be found in the DAL set are used to compare with the MPL values. Of the 93 unique words found in the four lists 55 can also be found in the DAL set.

Table 19: Numerical comparison of MPL with annotated word sets

	MPL evaluation deviation	MPL activation deviation	MPL angle deviation
Set 1: (Russell & Barrett, 1999)	31.43 %	21.72 %	29.65 %
Set 2: (Russell & Lanius, 1984)	30.56 %	15.05 %	25.93 %
Set 3: (Desmet, 2002)	27.75 %	9.44 %	19.10 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	36.19 %	24.44 %	33.70 %
Set 5: DAL (Whissell, 1989)	28.24 %	20.14 %	26.60 %
Weighted average percentages	**30.98 %**	**19.55 %**	**28.12 %**
Set 5: DAL (Whissell, 1989) not normalized	24.58 %	20.64 %	26.60 %

Table 20: Classification comparison of MPL with annotated word sets

	% in diff. quadrant	% in diff. 45° rotated quadrant	% in diff. octant
Set 1: (Russell & Barrett, 1999)	72.73 %	60.00 %	55.56 %
Set 2: (Russell & Lanius, 1984)	76.92 %	31.58 %	68.42 %
Set 3: (Desmet, 2002)	0.00 %	0.00 %	100.00 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	84.62 %	57.14 %	85.71 %
Set 5: DAL (Whissell, 1989)	70.59 %	34.78 %	69.57 %
Weighted average percentages	**75.93 %**	**43.28 %**	**71.64 %**

Table 21: Number of words found by using the MPL method

	Total # of words	Words found by MPL	Words also found in set
Set 1: (Russell & Barrett, 1999)	15	73.33 %	73.33 %
Set 2: (Russell & Lanius, 1984)	40	47.50 %	47.50 %
Set 3: (Desmet, 2002)	24	8.33 %	8.33 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	28	50.00 %	50.00 %
Set 5: DAL (Whissell, 1989)	93	39.78 %	24.73 %
Total	**200**	**41.50 %**	**34.50 %**

As can be seen from the tables above the MPL method has an average deviation for the angle of 28%. Compared to the set error margin in the previous paragraph of 16% this is 12% larger. The difference in a classification perspective is much worse. Compared to the set error margin there is a difference of more than 50% for the normal quadrants. Of course this difference is also due to the fact that only 42% of the words

could be calculated by the MPL method. From a numerical point of view the average differences of the values for evaluation differ 10% and the average differences of the values for activation only differ 3% with the set error margin, which could indicate a certain correlation between them.
Because (Kamps & Marx, 2001) suggest that the MPL method might work better for extreme valued words (i.e. words with a very high or very low value for activation or evaluation), more experiments have been done.

Lists of words with an extreme high value for evaluation, an extreme high value for activation, an extreme low value for evaluation and an extreme low value for activation have been extracted from the DAL set. For these lists the MPL values were calculated. These values were compared to the values of the DAL set. The following tables show the comparison between these values.

Table 22: Numerical comparison of MPL and DAL for extreme valued words

	MPL evaluation deviation	**MPL activation deviation**
Set 1: Extreme high evaluation	46.43 %	-
Set 2: Extreme low evaluation	40.00 %	-
Set 3: Extreme high activation	-	50.00 %
Set 4: Extreme low activation	-	55.00 %
Weighted average percentages	**42.76 %**	**54.33 %**

Table 23: Classification comparison of MPL and DAL for extreme valued words

	Percentage different sign
Set 1: Extreme high evaluation	46.67 %
Set 2: Extreme low evaluation	15.00 %
Set 3: Extreme high activation	0.00 %
Set 4: Extreme low activation	53.85 %
Weighted average percentages	**34.00 %**

Table 24: Number of words found by MPL for extreme valued word sets

	Total number of words	**Number of words found by MPL**	**%**
Set 1: Extreme high evaluation	64	15	23.44 %
Set 2: Extreme low evaluation	82	20	24.39 %
Set 3: Extreme high activation	30	2	6.67 %
Set 4: Extreme low activation	72	13	18.06 %
Total	**248**	**50**	**20.16 %**

From the percentages in the tables above we can only conclude that for extreme values words the quality of the MPL method only decreases. The quality in calculating the right value for the activation or evaluation is much worse for these kinds of words, but also the quality in the number of words that the MPL could calculated is drastically decreased.

10.3 Shallow cloud method using adjectives only

Next the shallow cloud method (using adjectives only) will be evaluated. The same structure of tables will be used to compare the different word sets with this method.

Table 25: Numerical comparison of SCM and annotated word sets

	SCM eva. deviation	SCM act. deviation	SCM angle deviation
Set 1: (Russell & Barrett, 1999)	31.43 %	21.72 %	29.65 %
Set 2: (Russell & Lanius, 1984)	30.56 %	15.05 %	25.93 %
Set 3: (Desmet, 2002)	28.26 %	25.54 %	28.00 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	36.19 %	24.44 %	33.70 %
Set 5: DAL (Whissell, 1989)	27.76 %	21.56 %	27.18 %
Weighted average percentages	**30.62 %**	**20.79 %**	**28.51 %**
Set 5: DAL (Whissell, 1989) not normalized	23.35 %	20.45 %	27.18 %

Table 26: Classification comparison of SCM

	% in diff. quadrant	% in diff. 45° rotated quadrant	% in diff. octant
Set 1: (Russell & Barrett, 1999)	72.73 %	60.00 %	56.56 %
Set 2: (Russell & Lanius, 1984)	76.92 %	31.58 %	68.42 %
Set 3: (Desmet, 2002)	100.00 %	50.00 %	83.33 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	84.62 %	57.14 %	85.71 %
Set 5: DAL (Whissell, 1989)	72.22 %	40.00 %	72.00 %
Weighted average percentages	**76.78 %**	**44.65 %**	**72.29 %**

Table 27: Number of words found by using SCM

	Total # of words	Words found by MPL	Words also found in set
Set 1: (Russell & Barrett, 1999)	15	73.33 %	73.33 %
Set 2: (Russell & Lanius, 1984)	40	47.50 %	47.50 %
Set 3: (Desmet, 2002)	24	25.00 %	25.00 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	28	50.00 %	50.00 %
Set 5: DAL (Whissell, 1989)	93	44.09 %	26.88 %
Total	**200**	**45.50 %**	**37.50 %**

By comparing the average weighted percentages of the first table with the values corresponding to the MPL method the following can be seen. In the percentage of the deviation of the evaluation a small decrease can be seen. For the percentage of the deviation of the activation a small increase can be seen. In total this comes down to an increase of the deviation of the angle of about 0.4%. From a classification point of view an increase of about 1% can be seen by comparing both methods.

Because of these differences are small and could just be because of natural bias, this is a relative good result. The shallow cloud method does not decrease the correlation between the lexical relations in WordNet and the measure of activation or evaluation.
Another question to investigate was if this method would be able to find more paths between the words, and thus is able to calculate the activation and evaluation for more words. By comparison of the last table with the last table of the MPL method, in the previous paragraph, a small increase in the number of found paths can be seen, about 4%. So this method does increase the number of words that can be calculated in a small degree.

10.4 Deep cloud method by using only adjectives

To evaluate the deep cloud method (using adjectives only) the same experiments have been done as before, only now by using the deep cloud method. Again the same structure of tables will be used to compare the different word sets with this method.

Table 28: Numerical comparison of DCM and annotated word sets

	DCM eva. deviation	DCM act. deviation	DCM angle deviation
Set 1: (Russell & Barrett, 1999)	31.19 %	19.09 %	27.92 %
Set 2: (Russell & Lanius, 1984)	32.68 %	20.24 %	29.44 %
Set 3: (Desmet, 2002)	26.23 %	25.15 %	26.81 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	40.52 %	30.42 %	40.79 %
Set 5: DAL (Whissell, 1989)	30.88 %	25.80 %	32.24 %
Weighted average percentages	**32.90 %**	**24.30 %**	**32.18 %**
Set 5: DAL (Whissell, 1989) not normalized	26.77 %	19.57 %	32.24 %

Table 29: Classification comparison of DCM

	% in diff. quadrant	% in diff. 45° rotated quadrant	% in diff. octant
Set 1: (Russell & Barrett, 1999)	81.82 %	50.00 %	55.56 %
Set 2: (Russell & Lanius, 1984)	78.95 %	40.91 %	72.73 %
Set 3: (Desmet, 2002)	50.00 %	50.00 %	66.67 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	85.71 %	75.00 %	87.50 %
Set 5: DAL (Whissell, 1989)	80.00 %	48.28 %	75.86 %
Weighted average percentages	**80.28 %**	**51.83 %**	**73.94 %**

Table 30: Number of words found by using DCM

	Total # of words	Words found by MPL	Words also found in set
Set 1: (Russell & Barrett, 1999)	15	73.33 %	73.33 %
Set 2: (Russell & Lanius, 1984)	40	55.00 %	55.00 %
Set 3: (Desmet, 2002)	24	25.00 %	25.00 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	28	57.14 %	57.14 %
Set 5: DAL (Whissell, 1989)	93	49.46 %	39.73 %
Total	**200**	**50.50 %**	**42.00 %**

Again we compare the tables above with the tables found for the MPL method. Here, as expected, a larger difference can be seen. For the average percentage of the evaluation deviation an increase of 2% can be seen. And for the measurement of activation also an increase of 4% is found. This results in an overall increase of the angle of 4%. The percentages of the words laying in a different quadrant or octant, thus seen from a classification perspective, also increases with an average of 4%.

By comparing the last table to the last table found for the MPL method, an overall increase of 9% was found. So this method can calculate the activation and evaluation for even more words than the shallow cloud method.

10.5 MPL, SCM and DCM using all words

Another idea was to investigate what happened when not only the lexical relations between adjectives in WordNet, but between all words would be used to calculate the evaluation and activation. The findings for these experiments are shown below.

Table 31: Numerical comparison of methods by using all words

	MPL angle deviation	SCM angle deviation	DCM angle deviation
Set 1: (Russell & Barrett, 1999)	28.78 %	29.32 %	30.32 %
Set 2: (Russell & Lanius, 1984)	21.50 %	26.96 %	30.82 %
Set 3: (Desmet, 2002)	37.73 %	42.40 %	41.74 %
Set 4: (Altarriba, Basnight, & Canary, 2003)	28.74 %	28.18 %	29.90 %
Set 5: DAL (Whissell, 1989)	32.40 %	33.72 %	35.18 %
Weighted average percentages	**29.60 %**	**32.07 %**	**33.74 %**

Table 32: Classification comparison of methods by using all words

	% in diff. quadrant			% in diff. 45° rotated quadrant			% in diff. octant		
	MPL	**SCM**	**DCM**	**MPL**	**SCM**	**DCM**	**MPL**	**SCM**	**DCM**
Set 1	61.54%	60.00%	60.00%	58.33%	57.14%	50.00%	76.92%	76.92%	76.92%
Set 2	37.50%	58.06%	62.50%	37.04%	39.39%	44.12%	65.52%	75.76%	79.41%
Set 3	50.00%	63.64%	63.64%	66.67%	81.82%	81.82%	100%	100.00%	95.45%
Set 4	61.90%	61.54%	61.54%	45.45%	53.85%	50.00%	76.00%	69.23%	76.92%
Set 5	50.00%	56.52%	57.45%	45.45%	58.82%	57.69%	70.21%	68.63%	71.15%
Weight. avg. perc.	**51.94%**	**58.91%**	**60.31%**	**48.18%**	**54.81%**	**54.01%**	**74.81%**	**75.86%**	**78.23%**

Table 33: Number of words found by using all words

	Total # of words	% of words found by			% also found in set		
		MPL	**SCM**	**DCM**	**MPL**	**SCM**	**DCM**
Set 1	15	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %	100.00 %
Set 2	40	72.50 %	82.50 %	85.00 %	72.50 %	82.50 %	85.00 %
Set 3	24	70.83 %	91.67 %	91.67 %	70.83 %	91.67 %	91.67 %
Set 4	28	89.29 %	92.86 %	92.86 %	89.29 %	92.86 %	92.86 %
Set 5	93	77.42 %	88.17 %	89.25 %	50.54 %	54.84 %	55.91 %
Total	**200**	**79.00 %**	**89.00 %**	**90.00 %**	**66.50 %**	**73.50 %**	**74.50 %**

By comparison with the tables from the previous paragraphs, only a small increase in the deviation of the angles occurs. The percentages for the number of words that are in a different quadrant decrease by more than 20%, the percentages for the differences in the rotated quadrant increases and the percentages for the differences in octant also increases a little. The decrease of difference of quadrant of 20% is very promising, but still from a classification point of view the results are rather dissapointing.

The number of found paths does increase a lot, in general almost twice as many paths can be found, while the increase of deviation of the angles only changes by a few percentages.

10.6 Comparison of all methods and conclusion

In the previous paragraphs all methods have been compared in great detail necessary to answer the question set up to investigate the proposed method of (Kamps & Marx, 2001) and the proposed novel methods. In this paragraph a brief overview of the results of the different methods is given.

First a numerical overview is given of the difference of the various methods. The abbreviation AO stands for “Adjectives Only” and AW stands for “All Words”.

Table 34: Methods compared on basis of their accuracy

Hand annotated	MPLAO	SCMAO	DCMAO	MPLAW	SCMAW	DCMAW
16.28 %	28.12 %	28.51 %	32.18 %	29.61 %	32.08 %	33.74 %

The methods were initially developed to increase the number of paths that could be found (i.e. the number of words that could be calculated), without modifying the calculated activation and evaluation too much.

Table 35: Methods compared on basis of the number of calculated words

MPLAO	SCMAO	DCMAO	MPLAW	SCMAW	DCMAW
41.50 %	45.50 %	50.50 %	79.00 %	89.00 %	90.00 %

As can be seen from the two tables above, the initial method as proposed by (Kamps & Marx, 2001) of calculating the activation and evaluation of a word has an average deviation which is much higher than the deviation between the hand annotated sets. Therefore the only conclusion is that this method cannot be used to calculate these values.
The methods to improve this initial method by enabling it to calculate more words, without compromising the calculated values do work, as can be seen from the second table. An increase of almost 50% can be obtained when using the lexical relations of WordNet between all words.

Now a classification point of view is given, by comparing the methods on basis of the difference in quadrants or octants.

Table 36: Classification comparison of methods in quadrants

Hand annotated	MPLAO	SCMAO	DCMAO	MPLAW	SCMAW	DCMAW
25.00 %	75.93 %	76.79 %	80.28 %	51.94 %	58.91 %	60.31 %

Table 37: Classification comparison of methods in rotated quadrants

Hand annotated	MPLAO	SCMAO	DCMAO	MPLAW	SCMAW	DCMAW
37.70 %	43.28 %	44.66 %	51.83 %	48.18 %	54.81 %	54.01 %

Table 38: Classification comparison of methods in octants

Hand annotated	MPLAO	SCMAO	DCMAO	MPLAW	SCMAW	DCMAW
56.72 %	71.64 %	72.29 %	73.94 %	74.81 %	75.86 %	78.23 %

The only conclusion that can be made from a classification point of view is that the proposed methods do not even come near a good classification. Even the percentages for the hand annotated sets are very high, which shows how subjective this data is.
There is a surprisingly better classification for the normal quadrants when all words are used, but still this differs about 25% with the hand annotated sets.

Part IV

Final results

"The most beautiful thing we can experience is the mysterious.
It is the source of all true art and all science.
He to whom this emotion is a stranger,
who can no longer pause to wonder and stand rapt in awe,
is as good as dead: his eyes are closed."

- Albert Einstein

Conclusion and future work Chapter 11

In this chapter a discussion is put forward, that combines an overall conclusion of the work done for this thesis, the way this helps the scientific field in which it belongs and future recommendations that arose while doing the thesis research.

The first paragraph discusses, in respect to the setted up research questions, which results are made and how the research questions are answered by the results. The second paragraph describes the contribution that has been made to the field of research of this thesis. The last paragraph will give recommendations for future research.

11.1 Goals and objectives

In the introduction the following goals were determined to demarcate the research done for this thesis:

- Creation of a natural language processing platform, for further research in the field of NLP and textual affect sensing, which is easy to use, expandable, easy to access and powerful.

- Investigation of the proposed theory of (Kamps & Marx, 2001). Is this theory a way to calculate activation and evaluation for (emotional) words?; Can there be any improvements?; Can this theory support keyword spotting techniques?

- Investigation of the proposed theory of (Liu, Lieberman, & Selker, 2003) and creation of the basis of this method for further research.

A detailed overview will now be given of the results of each of the goals.

11.1.1 NLP Affect Toolbox

The idea was to create a platform for general natural language processing and specifically a platform which supports textual affect sensing research. Therefore an extensive literature study has been conducted prior to the work of this thesis. From this literature study various profound methods, tools and corpora were selected, which are implemented in the so called "NLP Affect Toolbox".
The toolbox is implemented in C# in an object oriented manner. It can be used from a programmer's point of view and from a researcher's point of view. From a programmer's point of view the toolbox can be used as an programming library in which most basic textual affect sensing necessaries are available. From a researchers point of view the toolbox's interface can be used to investigate existing theories and it can support research for new methods and theories.

The selected corpora: "WordNet", "Dictionary of Affect in Language" and "ConceptNet" are implemented in an object oriented manner. For the implementation of WordNet, the object oriented version of this data was used as implemented by Proxem. For the corpora of DAL and ConceptNet only the data was extracted and placed in an object oriented programming structure to pursue speed and accessibility.
Browsers were created to enable users without programming experience to investigate the contents of the various corpora.

The basic natural language processing has been implemented by using the ANTELOPE ("Advanced Natural Language Object-oriented Processing Environment") environment created by Proxem. This environment handles the basic natural language processing (e.g. sentence splitting, chunking, part-of-speech tagging) that forms the basis for many methods. All basic natural language processing can be used through the library and through interfaces created to show the abilities of ANTELOPE. All natural language processing that ANTELOPE supports is described in chapter 4.

The basic natural language processes were used as a basis for some experiments of the extraction of the semantics for a given text. This basic extraction of the meaning in text is necessary for many methods of textual affect sensing. Processes as negation detection, tense detection, object extraction, relation finding between objects, extraction of properties of objects have all been designed and implemented.

The NLP Affect Toolbox also facilitates the visualization of parsed sentence tree. The parsing of these trees is done by the Proxem ANTELOPE environment and is being displayed by the Lithium component.

Furthermore, the algorithms, that were needed for the experiments of the MPL, shallow cloud and deep cloud methods to investigate the lexical relations to measure the activation and evaluation of words, were implemented in the toolbox. Also a tool that was used to calculate the results of the experiments was implemented in the toolbox.

Altogether the toolbox consists of 11.000 lines of C# code that bring together all tools, corpora and experiments and sets a solid base for textual affect sensing research for the future.

11.1.2 Lexical relations to measure activation and evaluation

The investigation of the proposed research by (Kamps & Marx, 2001) has been done in great detail. First of all the Minimal Path Length algorithm has been implemented by using three different search algorithms (i.e. depth-first, breadth-first and bidirectional search). The bidirectional search algorithm has also been implemented by using the multithreaded paradigm to speed up the overall time to process up to four times in the current state-of-art of single processor hardware. These algorithms are used to verify and validate the proposed research and to support the proposed ACT and EVA functions of (Kamps & Marx, 2001) to measure the activation and evaluation of words.

The initial MPL method appeared to be rather disappointing, by comparison to manually annotated sets of words it had a numerical difference of ± 28% and from a classification point of view (with the classes set as the quadrants of the circumplex of affect) ± 77% of the words lay in the wrong quadrant.

The main idea of improvement was to enable the method to calculate the activation and evaluation for more words. To improve the quantity of words for which the values could be calculated the shallow and deep cloud methods were invented. These methods improved this by a maximum of ± 10%, so for 50% of the selected words these values could be calculated (while only using adjectives), while the decrease of quality of was only around 4%.

Furthermore the capabilities of the MPL, shallow cloud and deep cloud method were investigated by seeing what happened when all word forms of WordNet where used, instead of the adjectives only. This turned out to be very positive. From a numerical point of view almost no change could be seen in the percentage of deviation between the manually annotated sets and the calculated values, but from a classification point of view (i.e. classes defined as the quadrant of the circumplex of affect) there was an improvement of ± 25% for all methods. By using all word forms of WordNet the number of words that the methods could calculate the values for was also increased by ± 40%, so for a maximum of ± 90% of the selected words the values could be calculated.

Altogether significant improvements were made, by the introduction of the various new methods and the use of all word forms, to the initial method of (Kamps & Marx, 2001). Also the implementation of the multithreaded bidirectional MPL functions improved the speed of calculating the activation and evaluation for words drastically, so more in depth research for this method could be done to evaluate the quality of the theory better. For all of these experiments graphical user interfaces were created in the NLP Affect Toolbox, enabling users of the toolbox to investigate the capabilities of these functions for different word sets.

11.1.3 Commonsense knowledge approach

The commonsense knowledge approach as proposed by (Liu, Lieberman, & Selker, 2003) for textual affect sensing has been studied and a basis for this theory has been created. The commonsense knowledge base ConceptNet has been implemented in the toolbox and a basis for the extraction of the semantics has been made, which are the primary necessities for this theory.

Unfortunately the limitation of time did not allow for the completion of the general implementation of the proposed theory. This theory does solve most of the problems encountered with other proposed methods, as sensing affect from text where no “emotional” words are used.

11.2 Contribution to the field

The NLP Affect Toolbox created for this thesis forms a solid object oriented base for general natural language processing and textual affect sensing research. The toolbox can be used by researchers with programming knowledge to easily set up and test theories. The toolbox can also be used, in a more limited way, by researchers that do not have programming experience through the graphical user interface. The creation of such a toolbox saves researchers a lot of time and brings new capabilities and ideas to the field of research.

The research done for the use of the lexical relations to measure the activation and evaluation of words had been investigated thoroughly and has shown us some degree of correctness, but by comparison to manually annotated sets of word it showed us from a classification point of view to much deviation. This theory can support textual affect sensing in some degree, but not as a single keyword spotting method.

A basis has been laid out for the implementation of the proposed theory to solve the problem by reasoning over the semantics of the text by means of commonsense knowledge, by the implementation of ConceptNet and the semantic parser.

11.3 Future work

The NLP Affect Toolbox can support all research in the field of natural language processing and should therefore be used to maintain efficiency. It can also be easily expanded due to its set up and the use of the object oriented paradigm and programming language.

In the field of textual affect sensing many theories have been tried and failed, this due to the ever changing form of language and the way cultures change the semantics of language. This problem can only be tackled by using commonsense knowledge in the broadest sense of the term. The way in which (Liu, Lieberman, & Selker, 2003) propose this to do is very fascinating and needs more investigation. First of all an implementation should be made of their proposed theory, which still is open for a lot of improvements. The way of inferencing over the commonsense rules can be improved and the way in which the language (i.e. a piece of text) is dissected to inference upon can also change a lot for textual affect sensing.
This novel way of tackling this problem gives us new insight in the way affect can be sensed, but can also give us the solution to extracting all semantic information from text.

For the matter of the proposed work of (Kamps & Marx, 2001), this theory has some potential, but in the end will give us a generalized form or value of the emotion contained by words. This emotional value is probably very culture or group depending and a generalized value would always give a slightly different picture of the containing emotion, therefore people call this kind of information subjective. Altough this value is slightly different it can point out a direction of emotional content over a larger piece of text (i.e. paragraph or chapter). And therefore could be usable to combine with other techniques to create a better end result.

The semantic parser as described in this thesis has been set up, but is still very experimental. It does give good performance on simple sentences, but should be investigated and improved in future research. Such a semantic parser forms the basic for all natural language processing, based upon commonsense knowledge reasoning.

Bibliography

Altarriba, J., Basnight, D., & Canary, T. (2003). Emotion representation and perception acress cultures. *In W. J. Lonner, D. L. Dinnel, S. A. Hayes, & D. N. Sattler (Eds.), Online Readings in Psychology and Culture, Unit 4* , Chapter 5.

Arnold, M. (1960). Emotion and personality. *New York: Columbia University Press* .

Averill, J. R. (1980). A constructionist view of emotion. In R. Plutchik, & H. Kellerman, *Emotion: Theory, research and experience (vol. 1)* (pp. 305-339). New York: Academic Press.

Carver, C. (2001). Affect and the functional bases of behavior: On the dimensional structure of affective experience. *Personality and Social Psychology Review, 5* , 345-356.

Cowie, R., Douglas-Cowie, E., Tsapatsoulis, N., Votsis, G., Kollias, S., Fellenz, W., et al. (2001). Emotion recognition in human-computer interaction. *IEEE Signal Process. Mag. 18* , 32-80.

Damasio, A. (1994). *Descartes' Error: Emotion, Reason, and the Human Brain.* New York, NY: Gosset/Putnam Press.

Desmet, P. (2002). *Designing Emotion.* Delft: Delft University of Technology.

Dyer, M. (1987). Emotions and Their Computations: Three Computer Models. *Cognition and Emotion, Vol. 1* , 323-347.

Eckman, P., Friesen, W., & Ellsworth, P. (1982). What emotion categories or dimensions can observers judge from facial behavior? In P. Eckman, *Emotion in the human face* (pp. 39-55). New York: Cambridge University Press.

Ekman, P. (1994). All emotions are basic. In P. Ekman, & R. Davidson, *The nature of emotion: Fundamental questions* (pp. 7-19). New York: Oxford University Press.

Ekman, P. (1993). Facial Expression and Emotion. In P. Ekman, *American Psychologist* (pp. 384-392).

Ekman, P. (1994). Strong evidence for universals in facial expressions: A reply to Russell's mistaken critique. *Psychological Bulletin, 115* , 268-287.

Ekman, P., Friesen, W., & Ellsworth, P. (1972). *Emotion in the human face: Guidelines for research and an integration of findings.* New York: Pergamon Press.

Elliott, C. (1992). *The Affective Reasoner: A Process Model of Emotions in a Multi-agent System.* Northwestern University.

Ellsworth, P. (1994). Some reasons to expect universal antecedents of emotion. In P. Ekman, & R. Davidson, *The nature of emotion: Fundamental questions.* New York: Oxford University Press.

Esuli, A., & Sebastiani, F. (2005). Determining the semantic orientation of terms through gloss analysis. *CIKM*, (pp. 617-624).

Fellbaum, C. (1998). *WordNet: An Electronic Lexical Database.* Massachusetts: MIT Press.

Fitrianie, S., & Rothkrantz, L. (2006). Constructing Knowledge for Automated Text-Based Emotion Expressions. *International Conference on Computer Systems and Technologies - CompSysTech'06.*

Fragopanagos, N., & Taylor, J. (2005). Emotion recognition in human–computer interaction. *Elsevier Ltd* .

Frijda, N. (1986). *The Emotions.* New York: Cambridge University Press.

Goertzel, B., Silverman, K., Hartley, C., Bugaj, S., & Ross, M. (2000). The Baby Webmind Project. *Proceedings of AISB 2000, the annual conference of The Society for the Study of Artificial Intelligence and the Simulation of Behaviour* .

Gray, J. (1982). *The neuropsychology of anxiety.* Oxford: Oxford University Press.

Hu, M., & Liu, B. (2004). Mining and summarizing customer reviews. *In: Proc. KDD (2004)* .

Izard, C. (1992). Basic emotions, relations among emotions, and emotion–cognition relations. *Psychology Review, 99* , 561-565.

Izard, C. (1971). *The face of emotions.* New York: Appleton-Century-Crofts.

James, W. (1884). What is an emotion? In *Mind* (pp. 188-205).

Kamps, J., & Marx, M. (2001). Words with attitude. *CCSOM Working Paper*, (pp. 01-194).

Kim, S., & Hovy, E. (2005). Automatic Detection of Opinion Bearing Words and Sentences. *Companion Volume to the Proceedings of the 2nd IJCNLP.*

Kim, S., & Hovy, E. (2006). Identifying and Analyzing Judgment Opinions. *HLTNAACL 2006, ACL*, (pp. 200-207).

Lazarus, R. (1968). *Emotions and adaptation: Conceptual and empirical relations.* Nebraska: University of Nebraska Press.

Lenat, D. (1995). CYC: A large-scale investment in knowledge infrastructure. *Communications of the ACM, 38(11)* , 33-38.

Liu, H., & Singh, P. (2004). *Commonsense Reasoning in and Over Natural Language.* Boston: MIT press.

Liu, H., & Singh, P. (2004). ConceptNet: A Practical Commonsense Reasoning Toolkit. *BT Technology Journal, vol. 22(4)* , 211–226.

Liu, H., Lieberman, H., & Selker, T. (2003). A Model of Textual Affect Sensing using Real-World Knowledge. *Technical report, MIT Media Laboratory* .

Mayer, J. D., & Gaschke, Y. N. (1988). The experience and metaexperience of mood. *Journal of Personality and Social Psychology, 55* , 102-111.

McDougall, W. (1926). *An introduction to social psychology.* Boston: Luce.

Miller, G. (1990). An on-line lexical database. *International Journal of Lexicography, 13(4)* , 235-312.

Minsky, M. (2006). *The Emotion Machine: Commonsense Thinking, Artificial Intelligence, and the Future of the Human Mind.* Simon & Schuster .

Mowrer, O. (1960). *Learning theory and behavior.* New York: Wiley.

Nasukawa, T., & Yi, J. (2003). Sentiment Analysis: Capturing Favorability Using Natural Language Processing. *In Proc. K-CAP* (pp. 70-77). New York: ACM Press.

Neese, R. M. (1990). Evolutionary explanations of emotions. *Human Nature, 1* , 261-289.

Oatley, K., & Johnson-Laird, P. (1987). Towards a cognitive theory of emotions. In *Cognition & Emotion* (pp. 29-50).

Orthony, A., & Turner, T. (1990). What's basic about basic emotions? *American Psychological Association, Inc.* , 17.

Ortony, A., Clore, G. L., & Collins, A. (1988). *The Cognitive Structure of Emotions.* Cambridge: Cambridge University Press.

Osgood, C. (1957). *The Measurement of Meaning.* Urbana, IL.: University of Illinois Press.

Panksepp, J. (1982). Toward a general psychobiological theory of emotions. In *The Behavioral and Brain Sciences* (pp. 407-467).

Pennebaker, J. W., Mehl, M. R., & Niederhoffer, K. G. (2003). Psychological aspects of natural language use: our words, our selves. *Annu. Rev. Psychol. 54* , 547-577.

Plutchik, R. (1980). A general psychocvolutionary theory of emotion. In R. Plutchik, & H. Kellerman, *Emotion: Theory, research, and experience." Vol. 1. Theories of emotion* (pp. 3-31). New York: Academic Press.

Polanyi, L., & Zaenen, A. (2004). Contextual valence shifters. In J. Shanahan, Y. Qu, & J. Wiebe, *Computing Attitude and Affect in Text: Theory and Applications, The Information Retrieval Series, vol. 20* (pp. 1-10).

Proxem. (2008, April). *www.proxem.com.* Retrieved 2008, from Proxem.

Riloff, E., Wiebe, J., & Wilson, T. (2003). Learning Subjective Nouns Using Extraction Pattern Bootstrapping. *In Proc. CoNLL-2003.*

Rosengren, K. E. (1999). *Communication - an introduction.* Sage Publications Ltd.

Russell, J. (1994). Is there universal recognition of emotion from facial expression? A review of the cross-cultural studies. *Psychological Bulletin, 115* , 102–141.

Russell, J., & Barrett, L. (1999). Core affect, prototypical emotional episodes, and other things called emotion: Dissecting the elephant. *Journal of Personality and Social Psychology, 76* , 805-819.

Russell, J., & Lanius, U. (1984). Adaptation Level and the Affective Appraisal of Environments. *Journal of Environmental Psychology, 4* , 119-135.

Russell, S., & Norvig, P. (2003). *Artificial Intelligence, A Modern Approach.* New Jersey: Pearson Education, Inc.

Shaikh, M., Prendinger, H., & Ishizuka, M. (2007). Assessing Sentiment of Text by Semantic Dependency and Contextual Valence Analysis. *A. Paiva, R. Prada, and R.W. Picard (Eds.): ACII 2007, LNCS 4738, pp. 191–202, 2007. © Springer-Verlag Berlin Heidelberg* , 191-202.

Shaikh, M., Prendinger, H., & Ishizuka, M. (2007). SenseNet: A Linguistic Tool to Visualize Numerical-Valance Based Sentiment of Textual Data. *In Proc. ICON* , 147-152.

Singh, P. (2002). The public acquisition of commonsense knowledge. *In Proceedings of AAAI Spring Symposium.* Palo Alto, CA: AAAI.

Solomon, R. (2003). *What is an emotion?: Classic and contemporary readings.* New York: New York: Oxford University Press.

Subasic, P., & Huettner, A. (2001). Affect analysis of text using fuzzy semantic typing. *IEEE Transactions on Fuzzy Systems 9* , 483-496.

Tomkins, S. (1984). Affect theory. In K. Scherer, & P. Eckman, *Approaches to emotion* (pp. 163-195). Hillsdale, NJ: Erlbaum.

Tomkins, S. (1962). *Affect, imagery, consciousness.* New York: Springer.

Tooby, J., & Cosmides, L. (1990). The past explains the present: Emotional adaptations and the structure of ancestral environments. *Ethology and Sociobiology, 11* , 407-424.

Turney, P. (2002). Conference on Empirical Methods in Natural Language Processing. *Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews*, (pp. 79-86).

Turney, P. (2002). Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. *In Proceedings of the 2002 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, (pp. 79-86).

Valitutti, A., Strapparava, C., & Stock, O. (2004). Developing Affective Lexical Resources. *Psychology Journal, 2004, Volume 2, Number 1* , 61-83.

Watson, D., & Tellegen, A. (1985). Toward a consensual structure of mood. *Psychological Bulletin, 98* , 219-235.

Watson, J. (1930). *Behaviorism.* Chicago: University of Chicago Press.

Weiner, B., & Graham, S. (1984). An attributional approach to emotional development. In C. Izard, J. Kagan, & R. Zajonc, *Emotions, cognition, and behavior* (pp. 167-191). New York: Cambridge University Press.

Whissell, C. (1989). Whissell's Dictionary of Affect in Language. *The measurement of emotions (vol. 4)* , 113–131.

Wiebe, J. (2000). Learning subjective adjectives from corpora. *In Proc. AAAI.*

Wiebe, J., & Mihalcea, R. (2006). Word Sense and Subjectivity. *In Proc. ACL-06*, (pp. 1065-1072).

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

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing Contextual Polarity in Phrase-Level Sentiment Analysis. *In: Proc. HLT/EMNLP. ACL*, (pp. 347-354).

Part V

Appendices

List of possible parts of speech

Appendix A

Tag	Description	Examples
$	dollar	$ -$ --$ A$ C$ HK$ M$ NZ$ S$ U.S.$ US$
``	opening quotation mark	` ``
"	closing quotation mark	' "
(	opening parenthesis	([{
)	closing parenthesis	)] }
,	comma	,
--	dash	--
.	sentence terminator	. ! ?
:	colon or ellipsis	: ; ...
CC	conjunction, coordinating	& 'n and both but either et for less minus neither nor or plus so therefore times v. versus vs. whether yet
CD	numeral, cardinal	mid-1890 nine-thirty forty-two one-tenth ten million 0.5 one forty-seven 1987 twenty '79 zero two 78-degrees eighty-four IX '60s .025 fifteen 271,124 dozen quintillion DM2,000 ...
DT	determiner	all an another any both del each either every half la many much nary neither no some such that the them these this those
EX	existential there	there
FW	foreign word	gemeinschaft hund ich jeux habeas Haementeria Herr K'ang-si vous lutihaw alai je jour objets salutaris fille quibusdam pas trop Monte terram fiche oui corporis ...
IN	preposition or conjunction, subordinating	astride among uppon whether out inside pro despite on by throughout below within for towards near behind atop around if like until below next into if beside ...
JJ	adjective or numeral, ordinal	third ill-mannered pre-war regrettable oiled calamitous first separable ectoplasmic battery-powered participatory fourth still-to-be-named multilingual multi-disciplinary ...
JJR	adjective, comparative	bleaker braver breezier briefer brighter brisker broader bumper busier calmer cheaper choosier cleaner clearer closer colder commoner costlier cozier creamier crunchier cuter ...

JJS	adjective, superlative	calmest cheapest choicest classiest cleanest clearest closest commonest corniest costliest crassest creepiest crudest cutest darkest deadliest dearest deepest densest dinkiest ...
LS	list item marker	A A. B B. C C. D E F First G H I J K One SP-44001 SP-44002 SP-44005 SP-44007 Second Third Three Two * a b c d first five four one six three two
MD	modal auxiliary	can cannot could couldn't dare may might must need ought shall should shouldn't will would
NN	noun, common, singular or mass	common-carrier cabbage knuckle-duster Casino afghan shed thermostat investment slide humour falloff slick wind hyena override subhumanity machinist ...
NNP	noun, proper, singular	Motown Venneboerger Czestochwa Ranzer Conchita Trumplane Christos Oceanside Escobar Kreisler Sawyer Cougar Yvette Ervin ODI Darryl CTCA Shannon A.K.C. Meltex Liverpool ...
NNPS	noun, proper, plural	Americans Americas Amharas Amityvilles Amusements Anarcho-Syndicalists Andalusians Andes Andruses Angels Animals Anthony Antilles Antiques Apache Apaches Apocrypha ...
NNS	noun, common, plural	undergraduates scotches bric-a-brac products bodyguards facets coasts divestitures storehouses designs clubs fragrances averages subjectivists apprehensions muses factory-jobs ...
PDT	pre-determiner	all both half many quite such sure this
POS	genitive marker	' 's
PRP	pronoun, personal	hers herself him himself hisself it itself me myself one oneself ours ourselves ownself self she thee theirs them themselves they thou thy us
PRP$	pronoun, possessive	her his mine my our ours their thy your
RB	adverb	occasionally unabatingly maddeningly adventurously professedly stirringly prominently technologically magisterially predominately swiftly fiscally pitilessly ...
RBR	adverb, comparative	further gloomier grander graver greater grimmer harder harsher healthier heavier higher however larger later leaner lengthier less-perfectly lesser lonelier longer louder lower more ...
RBS	adverb, superlative	best biggest bluntest earliest farthest first furthest hardest heartiest highest largest least less most nearest second tightest worst
RP	particle	aboard about across along apart around aside at away back before behind by crop down ever fast for forth from go high i.e. in into just later low more off on open out over per pie raising start teeth that through under unto up up-pp upon whole with you
SYM	symbol	% & ' " ".)). * + ,. < = > @ A[fj] U.S U.S.S.R * ** ***
TO	"to" as preposition or infinitive marker	to
UH	interjection	Goodbye Goody Gosh Wow Jeepers Jee-sus Hubba Hey Kee-reist Oops amen huh howdy uh dammit whammo shucks heck anyways whodunnit honey golly man baby diddle hush sonuvabitch ...
VB	verb, base form	ask assemble assess assign assume atone attention avoid bake balkanize bank begin behold believe bend benefit bevel beware bless boil bomb boost brace

		break bring broil brush build ...
VBD	verb, past tense	dipped pleaded swiped regummed soaked tidied convened halted registered cushioned exacted snubbed strode aimed adopted belied figgered speculated wore appreciated contemplated ...
VBG	verb, present participle or gerund	telegraphing stirring focusing angering judging stalling lactating hankerin' alleging veering capping approaching traveling besieging encrypting interrupting erasing wincing ...
VBN	verb, past participle	multihulled dilapidated aerosolized chaired languished panelized used experimented flourished imitated reunifed factored condensed sheared unsettled primed dubbed desired ...
VBP	verb, present tense, not 3rd person singular	predominate wrap resort sue twist spill cure lengthen brush terminate appear tend stray glisten obtain comprise detest tease attract emphasize mold postpone sever return wag ...
VBZ	verb, present tense, 3rd person singular	bases reconstructs marks mixes displeases seals carps weaves snatches slumps stretches authorizes smolders pictures emerges stockpiles seduces fizzes uses bolsters slaps speaks pleads ...
WDT	WH-determiner	that what whatever which whichever
WP	WH-pronoun	that what whatever whatsoever which who whom whosoever
WP$	WH-pronoun, possessive	whose
WRB	Wh-adverb	how however whence whenever where whereby whereever wherein whereof why

List of possible phrases

Appendix B

Abbreviation	Full name	Description
ADJP	Adjective phrase	The phrase is headed by an adjective
ADVP	Adverbial phrase	This phrase acts in the position of an adverb
CONJP	Conjunctive phrase	This phrase is used to indicate several multi-word conjunctions. ("as well as")
FRAG	Fragment	
INTJ	Interjection	Used instead of the POS tag UH
Leaf	Leaf node	A word
LST	List marker	Is used to include the surrounding punctuation
NAC	Not a constituent	This is used to show the scope of certain pre-nominal modifiers within a noun phrase
NP	Noun phrase	
NX	To mark the head of the noun phrase	This is used with the complex noun phrase
PP	Prepositional phrase	
PRN	Parenthetical	
PRT	Particle	Same as the 'RP' tag in the POS tag set
QP	Quantifier phrase	Used within the noun phrase
RRC	Reduced relative clause	
S	Simple declarative clause	This clause is not introduced by a subordinating conjunction or wh-word. This clause also does not show the subject verb inversion
SBAR	Subordinate clause	Clause that is introduced by a subordinating conjunction
SBARQ	Direct question introduced by a wh-word or wh-phrase	
SINV	Inverted declarative sentence, subject is inverted	In other words, subject follows the tensed verb or modal
SQ	Sub-constituent of SBARQ which does not	

	include wh-question	
UCP	Unlike coordinated phrase	
VP	Verb phrase	
WHADJP	Wh-adjective phrase	An adjectival phrase containing a wh-adverb
WHADVP	Wh-adverb phrase	Contains a wh-adverb such as “how”
WHNP	Wh-noun phrase	Contains some wh-words such as, “who” and “which”
WHPP	Wh-prepositional phrase	This is a prepositional phrase containing a wh-noun phrase
X	Unknown constituent	

List of syntax dependencies

Appendix C

Shallow syntax dependency types

Abbri-viation	**Description**	**Example**
abbrev	The "abbreviation appositional modifier" grammatical relation. An abbreviation modifier of an NP is an NP that serves to abbreviate the NP.	Examples: "The Australian Broadcasting Corporation (ABC)" → abbrev(Corporation, ABC)
advcl		
advmod	The "adverbial modifier" grammatical relation. An adverbial modifier of a word is an RB or ADVP that serves to modify the meaning of the word.	Examples: "genetically modified food" → advmod(modified, genetically)
amod	The "adjectival modifier" grammatical relation. An adjectival modifier of an NP is any adjectival phrase that serves to modify the meaning of the NP.	Examples: "Sam eats red meat" → amod(meat, red)
appos	The "appositional modifier" grammatical relation. An appositional modifier of an NP is an NP that serves to modify the meaning of the NP.	Examples: "Sam, my brother, eats red meat" → appos(Sam, brother)
infmod	The "infinitval modifier" grammatical relation. A participial modifier of an NP is an S/VP that serves to modify the meaning of the NP.	Examples: "points to establish are ..." → infmod(points, establish)
mod	The "modifier" grammatical relation. A modifier of a VP is any constituent that serves to modify the meaning of the VP (but is not an ARGUMENT of that VP); a modifier of a clause is an modifier of the VP which is the predicate of that clause.	Examples: "I swam in the pool last night" → mod(swam, in the pool), mod(swam, last night)
neg		
nn	The "noun compound modifier" grammatical relation. A noun compound modifier of an NP is any noun that serves to modify the head noun. Note that this has all nouns modify the rightmost a la Penn headship rules. There is no intelligent noun compound analysis.	Example: "Oil price futures" nn(futures, oil) nn(futures, price)

num	The "numeric modifier" grammatical relation. A numeric modifier of an NP is any number phrase that serves to modify the meaning of the NP.	Examples: "Sam eats 3 sheep" → num(sheep, 3)
partmod	The "participial modifier" grammatical relation. A participial modifier of an NP is a VP that serves to modify the meaning of the NP.	Examples: "truffles picked during the spring are tasty" → partmod(truffles, picked)
poss		
Possessive		
prep	The "prepositional modifier" grammatical relation. A prepositional modifier of an NP is any prepositional phrase that serves to modify the meaning of the NP.	Examples: "I saw a cat in a hat" → prep(cat, in)
purpcl		
rcmod		
tmod	The "temporal modifier" grammatical relation. A temporal modifier of a VP is any constituent that serves to modify the meaning of the VP by specifying a time; a temporal modifier of a clause is an temporal modifier of the VP which is the predicate of that clause.	Examples: "I swam in the pool last night" → tmod(swam, last night)
aux	The auxiliary grammatical relation. An auxiliary of a clause is a non-main verb of the clause.	Example: "Reagan has died" → aux(died, has)
Arg	The "argument" grammatical relation. An argument of a VP is a subject or complement of that VP; an argument of a clause is an argument of the VP which is the predicate of that clause.	Example: "Clinton defeated Dole" → arg(defeated, Clinton), arg(defeated, Dole)
subj	The "subject" grammatical relation. The subject of a VP is the noun or clause that performs or experiences the VP; the subject of a clause is the subject of the VP which is the predicate of that clause.	Examples: "Clinton defeated Dole" → subj(defeated, Clinton), "What she said is untrue" → subj(is, What she said)
nsubj	The "nominal subject" grammatical relation. A nominal subject is a subject which is an noun phrase.	Example: "Clinton defeated Dole" → nsubj(defeated, Clinton),
Nsubjpass	The "nominal passive subject" grammatical relation. A nominal passive subject is a subject of a passive which is an noun phrase.	Example: "Dole was defeated by Clinton" → nsubjpass(defeated, Dole)
csubj	The "clausal subject" grammatical relation. A clausal subject is a subject which is a clause.	Example: "What she said is untrue" → csubj(is, What she said)
comp	The "complement" grammatical relation. A complement of a VP is any object (direct or indirect) of that VP, or a clause or adjectival phrase which functions like an object; a complement of a clause is an complement of the VP which is the predicate of that clause.	Examples: "She gave me a raise" → comp(gave, me), comp(gave, a raise); "I like to swim" → comp(like, to swim)

obj	The "object" grammatical relation. An object of a VP is any direct object or indirect object of that VP; an object of a clause is an object of the VP which is the predicate of that clause.	Examples: "She gave me a raise" → obj(gave, me), obj(gave, raise)
dobj	The "direct object" grammatical relation. The direct object of a VP is the noun phrase which is the (accusative) object of the verb; the direct object of a clause is the direct object of the VP which is the predicate of that clause.	Example: "She gave me a raise" → dobj(gave, raise)
iobj	The "indirect object" grammatical relation. The indirect object of a VP is the noun phrase which is the (dative) object of the verb; the indirect object of a clause is the indirect object of the VP which is the predicate of that clause.	Example: "She gave me a raise" → iobj(gave, me)
ccomp	The "clausal complement" grammatical relation. A clausal complement of a VP is a clause which functions like an object of the verb; a clausal complement of a clause is the clausal complement of the VP which is the predicate of that clause.	Example: "I like to swim" → ccomp(like, swim)
acomp	The "adjectival complement" grammatical relation. An adjectival complement of a VP is a adjectival phrase which functions like an object of the verb; an adjectival complement of a clause is the adjectival complement of the VP which is the predicate of that clause.	Example: "She looks very beautiful" → acomp(looks, very beautiful)
det	The "determiner" grammatical relation. We're treating these as a sort of degenerate NP modifier, for now.	
sdep	The "semantic dependent" grammatical relation has been introduced as a supertype for the controlling subject relation.	
xsubj	The "controlling subject" grammatical relation.	Example: "Tom likes to eat fish" → xsubj(eat, Tom)
agent		
attr	- attributive	
auxpass	- passive auxiliary	
compl	- complementizer	
conj	- conjunct	
cop	- copula	
expl	- expletive (expletive there)	
mark	- marker (word introducing an advcl)	
number	- element of compound number	
pobj	- object of preposition	
prt	- phrasal verb particle	

Ref	- referent	
Rel	- relative (word introducing a rcmod)	
xcomp	- clausal complement with external subject	

Deep syntax dependency types

Dependency type	Description
Subject	a dependency between a subject and a verb. In a passive construction, a middle word "by" should be present
DirectObject	a dependency between a verb and a direct object
IndirectObject	a dependency between a verb and an indirect object
PrepObject	a dependency between a verb and a prepositional object. A middle word containing the preposition should be present
AdjectiveNoun	a dependency between an adjective and a noun modified by this adjective
NounOfNoun	a dependency between a noun (owner) and another noun (part of the owner)
TimeComplement	a dependency between a verb and a time complement
SpaceComplement	a dependency between a verb and a space complement

Word sets used

Appendix D

Table 39: Word set (Russell & Barrett, 1999)

Word	Normalized evaluation	Normalized activation	Angle in radians
alert	0.309016994	0.951056516	1.256637061
excited	0.587785252	0.809016994	0.942477796
elated	0.809016994	0.587785252	0.628318531
happy	0.951056516	0.309016994	0.314159265
contented	0.951056516	-0.309016994	5.969026042
serene	0.809016994	-0.587785252	5.654866776
relaxed	0.587785252	-0.809016994	5.340707511
calm	0.309016994	-0.951056516	5.026548246
bored	-0.382683432	-0.923879533	4.319689899
depressed	-0.707106781	-0.707106781	3.926990817
sad	-0.923879533	-0.382683432	3.534291735
upset	-0.951056516	0.309016994	2.827433388
stressed	-0.809016994	0.587785252	2.513274123
nervous	-0.587785252	0.809016994	2.199114858
tense	-0.309016994	0.951056516	1.884955592

Table 40: Word set (Russell & Barrett, 1999) calculated radian angles

Word	MPLAO	SCMAO	DCMAO	MPLAW	SCMAW	DCMAW
alert	1,922241	1,922241	1,895499	1,570796	1,576261	1,557267
excited	2,714965	2,714965	2,614925	2,356194	2,401715	2,297482
elated	1,985303	1,985303	2,062258	1,570796	1,536032	1,792598
happy	-	-	-	0,463648	0,49004	0,519419
contented	-	-	-	0,463648	0,613499	0,622634
serene	4,297882	4,297882	4,543763	5,300392	5,270899	4,947381
relaxed	-	-	-	3,60524	4,064701	4,19362
calm	4,297882	4,297882	4,443273	5,300392	5,270899	5,041089
bored	2,714965	2,714965	2,975421	0,244979	0,345872	0,401176
depressed	2,918116	2,918116	2,892256	2,446854	2,47464	2,409608
sad	2,991221	2,991221	2,966031	2,976444	2,957891	2,936169
upset	2,991221	2,991221	2,929231	2,158799	2,097204	1,98937
stressed	1,985303	1,985303	1,477317	1,405648	1,15119	0,965802
nervous	2,714965	2,714965	2,321779	2,677945	2,743406	2,647503
tense	-	-	-	0,785398	0,737358	0,739946

Table 41: Word set (Desmet, 2002)

Word	Normalized evaluation	Normalized activation	Angle in radians
astonishment	0	1	1.570796327
eagerness	0	1	1.570796327
curiosity	0	1	1.570796327
inspiration	0.707106781	0.707106781	0.785398163
desire	0.707106781	0.707106781	0.785398163
love	0.707106781	0.707106781	0.785398163
fascination	1	0	0
admiration	1	0	0
joyfulness	1	0	0
satisfaction	0.707106781	-0.707106781	5.497787144
softened	0.707106781	-0.707106781	5.497787144
relaxed	0.707106781	-0.707106781	5.497787144
awaiting	0	-1	4.71238898
deferent	0	-1	4.71238898
calm	0	-1	4.71238898
boredom	-0.707106781	-0.707106781	3.926990817
sadness	-0.707106781	-0.707106781	3.926990817
isolation	-0.707106781	-0.707106781	3.926990817
disappointment	-1	0	3.141592654
contempt	-1	0	3.141592654
jealousy	-1	0	3.141592654
irritation	-0.707106781	0.707106781	2.35619449
disgust	-0.707106781	0.707106781	2.35619449
alarm	-0.707106781	0.707106781	2.35619449

Table 42: Word set (Desmet, 2002) calculated radian angles

Word	MPLAO	SCMAO	DCMAO	MPLAW	SCMAW
astonishment	-	-	-	-	0,427896
eagerness	-	-	-	0,463648	0,500661
curiosity	-	1,570796	1,554032	0,165149	0,333821
inspiration	-	-	-	0,244979	0,31132
desire	-	-	-	0,244979	0,349908
love	-	-	-	0,244979	0,196276
fascination	-	-	-	0,463648	0,449281
admiration	-	-	-	0,785398	0,817744
joyfulness	-	1,985303	2,00484	0,785398	1,767714
satisfaction	-	-	-	-	0,488988
softened	4,712389	4,712389	4,831762	0,106736	6,263038
relaxed	-	-	-	3,60524	4,064701
awaiting	-	-	-	0,244979	0,278816
deferent	-	-	-	-	0,30012
calm	4,297882	4,297882	4,443273	5,300392	5,270899
boredom	-	-	-	-	-
sadness	-	2,991221	2,966031	3,141593	2,957891
isolation	-	-	-	-	6,100638
disappointment	-	-	-	-	6,114114
contempt	-	1,985303	2,00484	0,165149	0,298746
jealousy	-	-	-	-	-
irritation	-	-	-	0	0,477033
disgust	-	-	-	0,463648	0,427896
alarm	-	-	-	1,570796	1,551487

Table 43: Word set (Russell, 1984)

Word	Normalized evaluation	Normalized activation	Angle in radians
active	0.052137826	0.998639899	1.51863485
alive	0.361454984	0.932389562	1.200968415
interesting	0.507200323	0.861828192	1.038863186
exciting	0.462349542	0.886697751	1.090153187
exhilarating	0.594498208	0.804096935	0.934154867
arousing	0.645563874	0.763706282	0.869034935
stimulating	0.811534341	0.584304726	0.624023053
sensational	0.892670007	0.450710837	0.467561483
pleasing	0.973869595	0.227107929	0.22910698
pretty	0.99448562	0.104873029	0.105066225
beautiful	0.996521771	0.083332831	0.083429583
nice	0.999371851	-0.035438718	6.247739167
pleasant	0.998670709	-0.051544295	6.231618161
serene	0.816967863	-0.576683198	5.668522355
restful	0.734959101	-0.678111436	5.537995347
peaceful	0.647187289	-0.762331039	5.416277993
calm	0.618137103	-0.786070304	5.378759582
tranquil	0.530312513	-0.847802241	5.271358118
drowsy	0.223452414	-0.97471484	4.937743995
slow	0.10212484	-0.99477159	4.814692177
lazy	-0.058319812	-0.998297951	4.654036059
idle	-0.092980661	-0.995667915	4.619273819
monotonous	-0.213592452	-0.976922855	4.497138175
inactive	-0.275298224	-0.961358876	4.433489081
boring	-0.393919299	-0.91914503	4.307497194
dull	-0.581622873	-0.813458563	4.091666681
dreary	-0.64764842	-0.761939318	4.007894916
unstimulating	-0.759256602	-0.650791373	3.850218926
displeasing	-0.99896854	0.045407661	3.096169374
unpleasant	-0.992149079	0.125060808	3.016203533
dissatisfying	-0.975883982	0.218289838	2.921530951
repulsive	-0.982006447	0.188847394	2.951604366
uncomfortable	-0.928476691	0.371390676	2.761086276
tense	-0.648946606	0.760833952	2.276995419
forceful	-0.170723583	0.985318963	1.742360313
panicky	-0.462566007	0.886584846	2.051683607
frenzied	-0.553062843	0.833139539	2.156832365
intense	-0.290225091	0.956958409	1.865258371
rushed	-0.193143792	0.981170462	1.765161593
hectic	-0.366036235	0.930600599	1.945542399

Table 44: Word set (Russell, 1984) calculated radian angles

Word	MPLAO	SCMAO	DCMAO	MPLAW	SCMAW	DCMAW
active	1,78800	1,78800	1,770691	1,152572	1,042003	1,075802
alive	1,83312	1,83312	1,812484	1,063301	0,943289	0,979912
interesting	-	-	-	0,785398	0,817744	0,837504
exciting	-	-	-	1,570796	1,991342	1,914173
exhilarating	-	-	-	0	1,415716	1,188081
arousing	-	-	-	0,643501	0,503611	0,511657
stimulating	-	-	-	0,463648	0,419404	0,403112
sensational	2,71497	2,71497	3,141593	0	1,368414	3,222904
pleasing	-	-	-	0,463648	0,497413	0,526670
pretty	-	-	-	0,643501	0,505762	0,528653
beautiful	-	-	2,155898	-	-	2,070945
nice	0,15037	0,15037	0,156685	0,165149	0,271207	0,317249
pleasant	-	-	1,027034	-	0,626864	0,667177
serene	4,29788	4,29788	4,543763	5,300392	5,270899	4,947381
restful	-	-	-	-	-	-
peaceful	4,71239	4,71239	4,626209	0,571470	4,812501	4,837079
calm	4,29788	4,29788	4,443273	5,300392	5,270899	5,041089
tranquil	4,29788	4,29788	4,505277	4,712389	4,732454	4,798641
drowsy	-	-	-	2,896613	2,864753	2,824896
slow	4,71239	4,71239	4,737668	4,712389	4,072985	4,334413
lazy	0	0	0	4,712388	4,892798	4,929085
idle	2,71497	2,71497	2,645918	3,141593	3,106186	3,102387
monotonous	5,34514	5,34514	5,307581	5,819538	5,669686	5,733756
inactive	4,71239	4,71239	4,736552	4,712389	4,722211	4,723454
boring	4,71239	4,71239	4,861210	0,083141	0,247871	0,265926
dull	4,71239	4,71239	4,821885	4,712389	4,996521	5,313374
dreary	2,91812	2,91812	2,889156	2,896614	2,867184	2,850190
unstimulating	-	-	-	-	-	-
displeasing	-	-	-	-	-	-
unpleasant	-	-	-	-	-	-
dissatisfying	-	-	-	-	-	-
repulsive	2,71497	2,71497	2,609498	2,677945	1,170841	1,092373
uncomfortable	-	-	-	-	0,383735	0,400473
tense	-	-	-	0,785398	0,737358	0,739946
forceful	-	-	0	-	0,960414	0,342459
panicky	-	-	-	-	1,474534	1,539726
frenzied	2,71497	2,71497	2,614925	3,141593	3,141593	3,141593
intense	2,20355	2,20355	1,977540	1,570796	1,131607	1,061406
rushed	-	-	-	0	0	0
hectic	-	-	-	-	-	-

Table 45: Word set (Altarriba, Basnight, & Canary, 2003)

Word	Normalized evaluation	Normalized activation	Angle in radians
aroused	0.295867396	0.955229022	1.270432885
astonished	0.368923957	0.929459582	1.192945281
Excited	0.661803264	0.749677557	0.847574725
delighted	0.910366477	0.413802944	0.426627493
Happy	0.991146815	0.132770446	0.133163652
pleased	0.992419036	-0.12290019	6.159973605
Glad	0.981036743	-0.193821849	6.088128924
Serene	0.852148855	-0.523299464	5.732467005
content	0.797691494	-0.603065734	5.635846498
at_ease	0.769590848	-0.638537334	5.59058912
satisfied	0.755599396	-0.655034009	5.568957679
Relaxed	0.733293625	-0.679912097	5.535542554
Calm	0.712609306	-0.7015611	5.505599485
Sleepy	0.032241294	-0.999480114	4.744635863
Tired	-0.036462224	-0.999335032	4.675918672
Droopy	-0.245898025	-0.969295704	4.463942925
Bored	-0.489005764	-0.872280553	4.201439404
depressed	-0.859683775	-0.510826591	3.677738676
gloomy	-0.876338412	-0.481695949	3.644181606
Sad	-0.889250629	-0.457420287	3.616684684
miserable	-0.988936353	-0.148340453	3.290482601
frustrated	-0.786573359	0.617496843	2.476036298
distressed	-0.749926812	0.661520806	2.418747763
annoyed	-0.560999409	0.827816201	2.166388917
Afraid	-0.461637717	0.887068553	2.050636852
Angry	-0.162749272	0.986667459	1.734272765
Tense	-0.03320952	0.999448412	1.604011954
alarmed	-0.09622809	0.995359309	1.667173549

Table 46: Word set (Altarriba, Basnight, & Canary, 2003) calculated radian angles

Word	MPLAO	SCMAO	DCMAO	MPLAW	SCMAW	DCMAW
aroused	2,714965	2,714965	2,614925	0,919720	0,674531	0,634088
astonished	2,991221	2,991221	2,962048	0,463648	0,427895	0,456470
excited	2,714965	2,714965	2,614925	2,356194	2,401715	2,297482
delighted	-	-	-	0,463648	0,497413	0,526670
happy	-	-	-	0,463648	0,490040	0,519419
pleased	-	-	2,183463	0,463648	0,497413	0,555193
glad	-	-	-	0,463648	0,490040	0,584367
serene	4,297882	4,297882	4,543763	5,300392	5,270899	4,947381
content	-	-	-	0,463648	0,564392	0,573149
at_ease	-	-	-	-	-	-
satisfied	-	-	3,141593	0,244979	0,488988	0,512648
relaxed	-	-	-	3,605240	4,064701	4,193620
calm	4,297882	4,297882	4,443273	5,300392	5,270899	5,041089
sleepy	-	-	-	-	4,762091	4,940509
tired	5,345138	5,345138	2,195446	0,463648	0,462073	0,618412
droopy	-	-	-	0,463648	1,505229	1,576080
bored	2,714965	2,714965	2,975421	0,244979	0,345872	0,401176
depressed	2,918116	2,918116	2,892256	2,446854	2,474640	2,409608
gloomy	3,141593	3,141593	3,141593	3,141593	3,021320	3,007663
sad	2,991221	2,991221	2,966031	2,976444	2,957891	2,936169
miserable	2,991221	2,991221	2,968149	3,141593	3,081976	3,060086
frustrated	-	-	-	1,570796	2,278323	2,145235
distressed	2,543174	2,543174	2,282391	2,356194	2,287727	2,089913
annoyed	0,426627	0,426627	0,440697	0,463648	0,616477	0,622665
afraid	-	-	-	-	-	-
angry	2,596729	2,596729	2,544012	2,677945	2,493326	2,514991
tense	-	-	-	0,785398	0,737358	0,739946
alarmed	-	-	-	1,570796	1,551487	1,502526

Table 47: Word set of extreme high activation in DAL (Whissell, 1989)

Word	Normalized evaluation	Normalized activation	Angle in radians	DAL eva value	DAL act value
playing	0.485672619	0.874140782	1.063663861	0.5556	1.0000
running	-0.316199305	0.948692785	1.892516881	-0.3333	1.0000
energy	0	1	1.570796327	0.0000	1.0000
arrested	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
adventure	0.640169287	0.768233873	0.876077723	0.8333	1.0000
weapons	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
song	0.640169287	0.768233873	0.876077723	0.8333	1.0000
violence	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
travel	0.496120237	0.868253828	1.051671751	0.5714	1.0000
escape	0	1	1.570796327	0.0000	1.0000
speed	-0.164430978	0.986388591	1.735977436	-0.1667	1.0000
nightmare	-0.613951543	0.789343717	2.231853342	-0.7778	1.0000
argue	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
examination	0	1	1.570796327	0.0000	1.0000
lover	0.707106781	0.707106781	0.785398163	1.0000	1.0000
sport	0.624695048	0.780868809	0.896055385	0.8000	1.0000
stimulation	0.588186123	0.808725593	0.941982203	0.7273	1.0000
energetic	0.242535625	0.9701425	1.325817664	0.2500	1.0000
exercise	0.52999894	0.847998304	1.012197011	0.6250	1.0000
squirm	-0.554719397	0.832037494	2.158822007	-0.6667	1.0000
spontaneous	0.640169287	0.768233873	0.876077723	0.8333	1.0000
vigorously	0.447213595	0.894427191	1.107148718	0.5000	1.0000
explosion	-0.613951543	0.789343717	2.231853342	-0.7778	1.0000
urgent	-0.52999894	0.847998304	2.129395642	-0.6250	1.0000
victor	0.447213595	0.894427191	1.107148718	0.5000	1.0000
escaped	0.099503719	0.99503719	1.471127674	0.1000	1.0000
fret	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
offends	-0.650772615	0.759272681	2.279397893	-0.8571	1.0000
violently	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
vigorous	0.216909785	0.976191654	1.352148557	0.2222	1.0000

Table 48: Word set of extreme low activation in DAL (Whissell, 1989)

Word	Normalized evaluation	Normalized activation	Angle in radians	DAL eva value	DAL act value
Than	0	-1	4.71238898	0.0000	-1.0000
Here	-0.316199305	-0.948692785	4.390668426	-0.3333	-1.0000
isn't	-0.496120237	-0.868253828	4.193264405	-0.5714	-1.0000
Thou	-0.124034735	-0.992277877	4.588033986	-0.1250	-1.0000
won't	-0.496120237	-0.868253828	4.193264405	-0.5714	-1.0000
Wall	-0.316199305	-0.948692785	4.390668426	-0.3333	-1.0000
History	0	-1	4.71238898	0.0000	-1.0000
Evening	0	-1	4.71238898	0.0000	-1.0000
Tired	-0.668964732	-0.743294146	3.979573879	-0.9000	-1.0000
Concern	-0.554719397	-0.832037494	4.1243633	-0.6667	-1.0000
Sent	0.196116135	-0.980580676	4.90978454	0.2000	-1.0000
Mountain	0	-1	4.71238898	0.0000	-1.0000
Moments	-0.141462934	-0.989943553	4.570449926	-0.1429	-1.0000
Empty	-0.573462344	-0.819231921	4.101663016	-0.7000	-1.0000
elementary	0.196116135	-0.980580676	4.90978454	0.2000	-1.0000
Allow	0.316199305	-0.948692785	5.034109534	0.3333	-1.0000
False	-0.554719397	-0.832037494	4.1243633	-0.6667	-1.0000
Supper	0.514495755	-0.857492926	5.252808481	0.6000	-1.0000
Generally	-0.514495755	-0.857492926	4.17196948	-0.6000	-1.0000
Honey	0.514495755	-0.857492926	5.252808481	0.6000	-1.0000
Flat	-0.52999894	-0.847998304	4.153789665	-0.6250	-1.0000
Senior	-0.242535625	-0.9701425	4.467410317	-0.2500	-1.0000
Bench	-0.242535625	-0.9701425	4.467410317	-0.2500	-1.0000
Homes	0	-1	4.71238898	0.0000	-1.0000
Ending	0	-1	4.71238898	0.0000	-1.0000
Fuel	-0.447213595	-0.894427191	4.248741371	-0.5000	-1.0000
comment	-0.447213595	-0.894427191	4.248741371	-0.5000	-1.0000
conservative	-0.514495755	-0.857492926	4.17196948	-0.6000	-1.0000
Existed	0.196116135	-0.980580676	4.90978454	0.2000	-1.0000
unconscious	-0.52999894	-0.847998304	4.153789665	-0.6250	-1.0000
random	0	-1	4.71238898	0.0000	-1.0000
ideal	0.196116135	-0.980580676	4.90978454	0.2000	-1.0000
invented	0.351123442	-0.936329178	5.071159651	0.3750	-1.0000
string	-0.554719397	-0.832037494	4.1243633	-0.6667	-1.0000
scanned	0	-1	4.71238898	0.0000	-1.0000
interim	0	-1	4.71238898	0.0000	-1.0000
hoots	0	-1	4.71238898	0.0000	-1.0000
ho-ho	0.196116135	-0.980580676	4.90978454	0.2000	-1.0000
eighty	-0.316199305	-0.948692785	4.390668426	-0.3333	-1.0000
mummy	-0.196116135	-0.980580676	4.514993421	-0.2000	-1.0000
hmm	-0.447213595	-0.894427191	4.248741371	-0.5000	-1.0000
hermit	0	-1	4.71238898	0.0000	-1.0000
insulation	0	-1	4.71238898	0.0000	-1.0000
solemn	-0.141462934	-0.989943553	4.570449926	-0.1429	-1.0000
vacated	-0.447213595	-0.894427191	4.248741371	-0.5000	-1.0000
foggy	-0.554719397	-0.832037494	4.1243633	-0.6667	-1.0000
segments	-0.371390676	-0.928476691	4.331882603	-0.4000	-1.0000

moonlight	0.6	-0.8	5.355890089	0.7500	-1.0000
seldom	-0.351123442	-0.936329178	4.35361831	-0.3750	-1.0000
sufficiently	-0.316199305	-0.948692785	4.390668426	-0.3333	-1.0000
continental	-0.447213595	-0.894427191	4.248741371	-0.5000	-1.0000
inadequate	-0.554719397	-0.832037494	4.1243633	-0.6667	-1.0000
suite	0.141462934	-0.989943553	4.854328035	0.1429	-1.0000
housed	0	-1	4.71238898	0.0000	-1.0000
rows	0	-1	4.71238898	0.0000	-1.0000
shade	0.141462934	-0.989943553	4.854328035	0.1429	-1.0000
cone	0	-1	4.71238898	0.0000	-1.0000
institute	-0.164430978	-0.986388591	4.547207871	-0.1667	-1.0000
shadows	0	-1	4.71238898	0.0000	-1.0000
shades	0.164430978	-0.986388591	4.87757009	0.1667	-1.0000
scar	-0.624695048	-0.780868809	4.037648038	-0.8000	-1.0000
supervision	-0.514495755	-0.857492926	4.17196948	-0.6000	-1.0000
mellow	0.573462344	-0.819231921	5.323114945	0.7000	-1.0000
vases	0.371390676	-0.928476691	5.092895357	0.4000	-1.0000
communion	0	-1	4.71238898	0.0000	-1.0000
envelope	0	-1	4.71238898	0.0000	-1.0000
avocado	0	-1	4.71238898	0.0000	-1.0000
sermon	-0.640169287	-0.768233873	4.017670377	-0.8333	-1.0000
supplement	-0.316199305	-0.948692785	4.390668426	-0.3333	-1.0000
mm	-0.196116135	-0.980580676	4.514993421	-0.2000	-1.0000
seas	0.164430978	-0.986388591	4.87757009	0.1667	-1.0000
continuously	-0.447213595	-0.894427191	4.248741371	-0.5000	-1.0000

Table 49: Word set of extreme high evaluation in DAL (Whissell, 1989)

Word	Normalized evaluation	Normalized activation	Angle in radians	DAL eva value	DAL act value
graduate	0.813727973	0.581245891	0.620258945	1.0000	0.7143
Love	0.843647592	0.536897328	0.566755099	1.0000	0.6364
anniversary	1	0	0	1.0000	0.0000
socialize	0.780868809	0.624695048	0.674740942	1.0000	0.8000
Soft	0.9701425	-0.242535625	6.038206644	1.0000	-0.2500
Softly	0.9701425	0.242535625	0.244978663	1.0000	0.2500
woods	0.780868809	-0.624695048	5.608444365	1.0000	-0.8000
amusing	0.847998304	0.52999894	0.558599315	1.0000	0.6250
special	1	0	0	1.0000	0.0000
Hope	0.9647705	-0.263092915	6.016958643	1.0000	-0.2727
reflected	0.980580676	-0.196116135	6.085789747	1.0000	-0.2000
truelove	0.948692785	0.316199305	0.321720554	1.0000	0.3333
wisdom	0.936329178	0.351123442	0.35877067	1.0000	0.3750
Toys	0.8	0.6	0.643501109	1.0000	0.7500
relatives	0.9701425	-0.242535625	6.038206644	1.0000	-0.2500
Lottery	0.989943553	0.141462934	0.141939054	1.0000	0.1429
wedding	0.976191654	0.216909785	0.218647769	1.0000	0.2222
Treats	0.948692785	0.316199305	0.321720554	1.0000	0.3333
Beach	0.961527576	-0.274708428	6.004898856	1.0000	-0.2857
beautiful	0.941754695	-0.336300602	5.940199376	1.0000	-0.3571
beautifully	0.832037494	-0.554719397	5.695159627	1.0000	-0.6667
Sky	0.928476691	-0.371390676	5.90267893	1.0000	-0.4000
Silk	0.992277877	-0.124034735	6.158830313	1.0000	-0.1250
Heroic	0.928476691	0.371390676	0.380506377	1.0000	0.4000
sociable	0.868253828	0.496120237	0.519124576	1.0000	0.5714
Quick	0.843647592	0.536897328	0.566755099	1.0000	0.6364
Team	0.894427191	0.447213595	0.463647609	1.0000	0.5000
healthy	0.910382066	0.413768649	0.426589821	1.0000	0.4545
Happy	0.8	0.6	0.643501109	1.0000	0.7500
Ready	1	0	0	1.0000	0.0000
Real	1	0	0	1.0000	0.0000
happiness	0.780868809	0.624695048	0.674740942	1.0000	0.8000
reasonable	0.992277877	-0.124034735	6.158830313	1.0000	-0.1250
quests	0.9701425	-0.242535625	6.038206644	1.0000	-0.2500
mommy	0.894427191	-0.447213595	5.819537698	1.0000	-0.5000
comfortable	0.832037494	-0.554719397	5.695159627	1.0000	-0.6667
young	0.986388591	-0.164430978	6.118004198	1.0000	-0.1667
ace	0.813727973	-0.581245891	5.662926362	1.0000	-0.7143
lovey	0.894427191	-0.447213595	5.819537698	1.0000	-0.5000
win	0.874140782	0.485672619	0.507132466	1.0000	0.5556
loving	0.9701425	0.242535625	0.244978663	1.0000	0.2500
companion	0.928476691	-0.371390676	5.90267893	1.0000	-0.4000
comfort	0.832037494	-0.554719397	5.695159627	1.0000	-0.6667
company	0.894427191	-0.447213595	5.819537698	1.0000	-0.5000
comfortably	0.863791432	-0.503849542	5.75513573	1.0000	-0.5833
miracle	0.948692785	0.316199305	0.321720554	1.0000	0.3333
successes	1	0	0	1.0000	0.0000

millionaire	1	0	0	1.0000	0.0000
summer	0.928476691	-0.371390676	5.90267893	1.0000	-0.4000
summers	0.9701425	-0.242535625	6.038206644	1.0000	-0.2500
melodies	0.986388591	0.164430978	0.16518111	1.0000	0.1667
marvellous	0.863791432	0.503849542	0.528049578	1.0000	0.5833
vacation	0.989943553	0.141462934	0.141939054	1.0000	0.1429
lovely	0.993884947	-0.110420618	6.172539062	1.0000	-0.1111
scholarship	0.894427191	-0.447213595	5.819537698	1.0000	-0.5000
loves	0.913826621	0.406104551	0.418187216	1.0000	0.4444
giving	0.780868809	0.624695048	0.674740942	1.0000	0.8000
lovers	0.868253828	0.496120237	0.519124576	1.0000	0.5714
loved	0.923065999	0.384641602	0.394819522	1.0000	0.4167
winning	0.894427191	0.447213595	0.463647609	1.0000	0.5000
reliable	0.936329178	-0.351123442	5.924414637	1.0000	-0.3750
lover	0.707106781	0.707106781	0.785398163	1.0000	1.0000
comedy	0.768233873	0.640169287	0.694718604	1.0000	0.8333
relief	0.9701425	-0.242535625	6.038206644	1.0000	-0.2500

Table 50: Word set of extreme low evaluation in DAL (Whissell, 1989)

Word	Normalized evaluation	Normalized activation	Angle in radians	DAL eva value	DAL act value
angry	-0.894427191	0.447213595	2.677945045	-1.0000	0.5000
alone	-0.877878798	-0.478882884	3.640974406	-1.0000	-0.5455
break	-0.986388591	0.164430978	2.976411544	-1.0000	0.1667
less	-0.843647592	-0.536897328	3.708347753	-1.0000	-0.6364
war	-0.819231921	0.573462344	2.530866689	-1.0000	0.7000
wait	-0.948692785	-0.316199305	3.463313208	-1.0000	-0.3333
fear	-0.939804399	0.341712879	2.792853767	-1.0000	0.3636
waiting	-0.857492926	-0.514495755	3.682012154	-1.0000	-0.6000
mad	-0.980580676	0.196116135	2.944197094	-1.0000	0.2000
cannot	-0.877878798	-0.478882884	3.640974406	-1.0000	-0.5455
shot	-0.986388591	-0.164430978	3.306773763	-1.0000	-0.1667
missing	-0.752576695	-0.658504608	3.860422653	-1.0000	-0.8750
kill	-1	0	3.141592654	-1.0000	0.0000
nobody	-0.894427191	-0.447213595	3.605240263	-1.0000	-0.5000
murder	-0.980580676	0.196116135	2.944197094	-1.0000	0.2000
accident	-0.936329178	0.351123442	2.782821983	-1.0000	0.3750
anger	-0.747405195	0.664368478	2.414944106	-1.0000	0.8889
afraid	-0.936329178	0.351123442	2.782821983	-1.0000	0.3750
knows	-0.813727973	-0.581245891	3.761851599	-1.0000	-0.7143
lonely	-0.808725593	-0.588186123	3.770406778	-1.0000	-0.7273
pain	-0.868253828	0.496120237	2.622468078	-1.0000	0.5714
sick	-0.948692785	-0.316199305	3.463313208	-1.0000	-0.3333
shut	-0.759272681	-0.650772615	3.850194219	-1.0000	-0.8571
attack	-0.868253828	0.496120237	2.622468078	-1.0000	0.5714
losing	-0.991232229	0.132131256	3.009073873	-1.0000	0.1333
lose	-0.894427191	-0.447213595	3.605240263	-1.0000	-0.5000
arrested	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
weak	-0.961527576	-0.274708428	3.419879105	-1.0000	-0.2857
Argument	-0.847998304	0.52999894	2.582993338	-1.0000	0.6250
Pressure	-0.894427191	0.447213595	2.677945045	-1.0000	0.5000
Lack	-0.877878798	-0.478882884	3.640974406	-1.0000	-0.5455
Stupid	-0.894427191	-0.447213595	3.605240263	-1.0000	-0.5000
Refused	-0.948692785	-0.316199305	3.463313208	-1.0000	-0.3333
Weapons	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
Taxes	-0.928476691	-0.371390676	3.522099031	-1.0000	-0.4000
Confused	-0.99503719	0.099503719	3.041924001	-1.0000	0.1000
Lying	-0.923065999	-0.384641602	3.536412175	-1.0000	-0.4167
Politics	-1	0	3.141592654	-1.0000	0.0000
Loss	-0.983873058	-0.178868122	3.321428553	-1.0000	-0.1818
Limited	-0.868253828	-0.496120237	3.660717229	-1.0000	-0.5714
Shooting	-0.8	0.6	2.498091545	-1.0000	0.7500
Violence	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
Prevent	-0.992277877	0.124034735	3.017237659	-1.0000	0.1250
Failed	-0.894427191	-0.447213595	3.605240263	-1.0000	-0.5000
Burning	-0.743294146	0.668964732	2.408777552	-1.0000	0.9000
Weapon	-0.9701425	0.242535625	2.89661399	-1.0000	0.2500
Anti	-1	0	3.141592654	-1.0000	0.0000
Orders	-0.980580676	0.196116135	2.944197094	-1.0000	0.2000

Victim	-0.9701425	0.242535625	2.89661399	-1.0000	0.2500
mess	-0.936329178	-0.351123442	3.500363324	-1.0000	-0.3750
terrible	-0.986388591	0.164430978	2.976411544	-1.0000	0.1667
mistake	-0.874140782	-0.485672619	3.64872512	-1.0000	-0.5556
alarm	-0.9701425	0.242535625	2.89661399	-1.0000	0.2500
waste	-0.980580676	0.196116135	2.944197094	-1.0000	0.2000
confusion	-0.976191654	0.216909785	2.922944884	-1.0000	0.2222
unable	-0.928476691	-0.371390676	3.522099031	-1.0000	-0.4000
victims	-1	0	3.141592654	-1.0000	0.0000
blame	-0.980580676	0.196116135	2.944197094	-1.0000	0.2000
politicians	-1	0	3.141592654	-1.0000	0.0000
shame	-0.9701425	-0.242535625	3.386571317	-1.0000	-0.2500
prison	-1	0	3.141592654	-1.0000	0.0000
shots	-0.894427191	0.447213595	2.677945045	-1.0000	0.5000
tumor	-0.989943553	0.141462934	2.999653599	-1.0000	0.1429
conflict	-0.923065999	0.384641602	2.746773132	-1.0000	0.4167
parliament	-0.919135521	-0.393941484	3.546508578	-1.0000	-0.4286
awful	-0.976191654	-0.216909785	3.360240423	-1.0000	-0.2222
smoke	-0.919135521	-0.393941484	3.546508578	-1.0000	-0.4286
violent	-0.832037494	0.554719397	2.553566973	-1.0000	0.6667
burned	-0.874140782	0.485672619	2.634460188	-1.0000	0.5556
capture	-0.948692785	0.316199305	2.819872099	-1.0000	0.3333
accused	-0.961527576	-0.274708428	3.419879105	-1.0000	-0.2857
suffered	-0.9701425	0.242535625	2.89661399	-1.0000	0.2500
burns	-0.847998304	0.52999894	2.582993338	-1.0000	0.6250
murders	-0.928476691	0.371390676	2.761086276	-1.0000	0.4000
argue	-0.707106781	0.707106781	2.35619449	-1.0000	1.0000
attacked	-0.759272681	0.650772615	2.432991088	-1.0000	0.8571
anxious	-0.894427191	0.447213595	2.677945045	-1.0000	0.5000
missile	-0.948692785	0.316199305	2.819872099	-1.0000	0.3333
tension	-0.9701425	0.242535625	2.89661399	-1.0000	0.2500
blind	-0.868253828	-0.496120237	3.660717229	-1.0000	-0.5714
argued	-0.752576695	0.658504608	2.422762654	-1.0000	0.8750
burden	-0.9647705	-0.263092915	3.407819318	-1.0000	-0.2727