UNSUPERVISED EMOTION DETECTION FROM TEXT USING SEMANTIC AND SYNTACTIC RELATIONS

AMEETA AGRAWAL

A THESIS SUBMITTED TO THE FACULTY OF GRADUATE STUDIES
IN PARTIAL FULFILLMENT OF THE REQUIREMENTS FOR THE DEGREE OF
MASTER OF SCIENCE

GRADUATE PROGRAMME IN COMPUTER SCIENCE AND ENGINEERING
YORK UNIVERSITY
TORONTO, ONTARIO

OCTOBER 2011

UNSUPERVISED EMOTION DETECTION FROM TEXT USING SEMANTIC AND SYNTACTIC RELATIONS

by **Ameeta Agrawal**

A thesis submitted to the Faculty of Graduate Studies of York University in partial fulfilment of the requirements for the degree of

MASTER OF SCIENCE

UNSUPERVISED EMOTION DETECTION FROM TEXT USING SEMANTIC AND SYNTACTIC RELATIONS

by **Ameeta Agrawal**

Examination Committee Members:

1. Aijun An
2. Nick Cercone
3. Jeff Edmonds
4. Gabriela Alboiu

ABSTRACT

Emotion detection from text is a relatively new classification task with applications ranging from marketing to social media to eLearning. The objective of this thesis is to propose a novel unsupervised context-based approach to emotion detection from textual data at the sentence level. Our methodology does not depend on any existing manually created affect lexicon such as WordNet-Affect, thereby rendering it flexible to classify sentences into not only Ekman's model of six basic emotions but any emotional model such as Plutchik’s or Izard’s. Our method computes an emotion vector for each potential affect-bearing word based on the semantic relatedness between words. The method is further augmented by using the syntactic dependency structure of a sentence transforming our technique into a context-sensitive approach. Extensive evaluation on various data sets shows that our framework is a more generic and practical solution to real-world sentence-level emotion classification problem and yields more accurate results than other unsupervised approaches.

ACKNOWLEDGEMENTS

There are several people who brought me here and stood by me through this journey of grad school. To begin with, I express my heartfelt gratitude to my supervisor and mentor, Professor Aijun An, whose encouragement, guidance and kindness from day one until today has enabled me to not only understand my subject but also work on it with such enthusiasm.

I also take this opportunity to thank my committee members, Professor Nick Cercone, Jeff and Professor Gabriela Alboiu for their invaluable feedback in making my thesis complete and sound.

There is one person without whom the grad school would not have happened. Damon, thank you so very much for the inspiring discussion we had over hot chocolate that day in Second Cup!

I also owe huge thanks to Penguin for always being there, nudging and cheering me along.

This is where words fall short and so I won't even try – mom, dad and *bhai* – you are the best!

And lastly, this experience would have been lacking without the wonderful moments spent with my awesome friends. *We've still got a chance!*

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1
Introduction

Emotions are complex, fuzzy and easily misunderstood entities. Emotions in text are even trickier. Consider the illustrations in Figure 1.1, where the statement “John is back in town” could be a reason of joy for someone, sadness for someone else or if spoken angrily, trouble for John. Accompanied with hand gestures, facial expressions and vocal pitch, it can be easy, to some extent, to detect the underlying emotional tone of the speaker when s/he utters this sentence. But imagine all these aforementioned features were stripped away and all we had were the words – “John is back in town”. This sentence could now mean any emotion or none at all. The process of detecting whether a piece of text contains any emotional sentiments attached to it is known as emotion recognition from textual data.

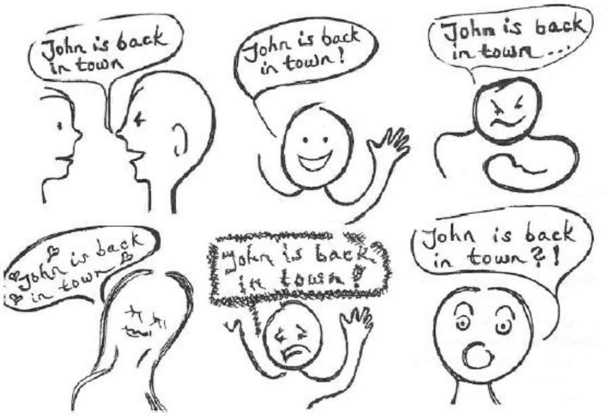

Figure 1.1 Illustrations from Alm-Arvius, C. (1998). Introduction to Semantics. Lund: Studentlitteratur. (p. 15).

Although there is no strict definition for *emotion*, most researchers agree that it is a particular feeling that characterizes a state of mind, such as joy, anger, love, fear and so on. On the other hand, *affect* is considered to be an emotion that influences behaviour or action. However, we noticed that the computer science literature from the field of affective sciences tends to use these two words interchangeably as they indicate some sort of an emotional state. In this thesis, emotion and affect refer to states such as joy, sadness, anger, guilt, surprise and so on.

1.1 Motivations

Most of the current approaches to emotion detection are based on supervised learning methods, in which a large set of annotated data (where text has been labelled with corresponding emotions) is needed to train the model. Although the supervised learning based methods can achieve good results compared to unsupervised methods, the availability of large annotated data sets is very low, and a model trained on the data set of one domain does not quite work well in another domain.

There are some methods that do not use supervised learning and thus do not rely on annotated data sets. However, most of these methods are based on manually designed dictionaries of emotion keywords. Those keywords (such as happy, delighted, and angry) bear obvious emotions and are often constructed using the synonyms of the most obvious emotional words for each emotion category. A problem with such an affect lexicon-based method is that the number of emotion categories is fixed and limited in the dictionary. Sometimes, new categories of emotion may be required for classification, which do not feature in the already created affect lexicons. Another problem of such affect lexicon-dependent approaches is that if a sentence expresses emotion using words that do not appear in the dictionary, then it would be considered to be unemotional by such methods. For example, the sentence “Izzy got lots of new toys for her first birthday yesterday!” conveys quite a happy feeling despite not containing any obvious happy keywords such as joy, glad, good, etc. Affect lexicon-dependent techniques fail to detect emotions from such sentences.

There are also methods that rely on manually-designed emotion detection rules. But designing such rules is not a trivial task. Once the rules have been constructed, updating them to incorporate newer categories is another time-consuming and effort-intensive process. Moreover, most of these rules have not been made publicly available.

Another problem with current emotion detection methods is that they look at individual words independently without considering the context a word is in. However, a word can bear different emotions in different contexts. By only looking at a word itself without considering its context, a sentence may be given a wrong emotion label.

In this thesis we propose a novel unsupervised context-based emotion detection method that does not rely on any emotion keyword dictionaries. The method does not need any training data with annotated emotion labels. It pre-computes a semantic relatedness score of a word to each emotion category based on a large collection of unlabeled documents such as Wikipedia, which is publicly available, and uses the semantic relatedness scores of the words in a sentence to detect the emotion(s) in the sentence. To consider the context a word is in, three types of word dependencies are identified within a sentence and are used to adjust the relatedness scores of some of the words in the sentence. The method computes a vector of emotion scores for a sentence, each corresponding to an emotion category, by combining the emotion scores of the affect bearing words in the sentence.

The emotion of the sentence can be identified by analyzing the score vector. In addition to not relying on any annotated data sets or emotion keyword dictionaries, our approach is not restricted to a fixed number of emotion categories. It can be extended to deal with other emotions easily. It can also detect neutral or mixed emotions.

1.2 Applications

New research in the emotion recognition field inspires new technologies that embrace these ideas and turn them into real-world applications. One of the most commonly known applications of affect detection is in the domain of market research. From early sentiment analysis (positive, negative or neutral classification) to more detailed emotion analysis (with classes such as happiness, sadness, anger and many more), businesses have always been eager to find out the consumer's reaction to their products. This enables the market moderators to develop better product designs and launches. Emotions are a huge influence on the way people decide to buy certain items. By gauging the general mood of the public, businesses can estimate how well their products and services are being received in the market.

Psychologists can better assist their patients by analyzing their transcripts for affective content. Similarly, teachers can better engage with their students by automatically identifying their current affective state in online classes where there is little face to face interaction. The only clues available at the teacher's disposal are contained in the text chat and emotion recognition tools can be very helpful in deciphering this emotional content. Likewise, automatic eLearning software can adapt itself to suit the student's current affective state. For example, if a student is getting frustrated by some difficult question, the software can pick up emotional hints and adjust the pace of learning.

Recently, researchers at the Cornell university have analyzed tweets from Twitter to chart a mood ring with findings such as "no matter how grumpy you might feel waking up, people are very positive during breakfast time" and "people tweet about drinking beer 7 hours before tweeting about being drunk" (Golder & Macy, 2011). Such affective social content is increasingly being integrated into social networks and smart phone devices to instantly provide users with emotional indicators. Online news media have also started gathering emotional feedback from the readers. For example, Yahoo! now provides six emotion smileys along with comments at the bottom of each news article to understand how the reader felt after reading the item. This can allow them to tailor and display their news content more interactively as well as interestingly.

The results from another research that also collected data from Twitter to analyze emotions revealed some interesting patterns (Kalev Leetaru, 2011). It was found that there were quite a lot of negative sentiments echoing in the online world *well before* the Egyptian revolution took place. Applications can be devised that in the future could predict such phenomenon before it happens.

Technology using emotion detection algorithms can automatically generate emoticons that are widely used in online instant messaging programs. One particular area that stands to benefit from reliable emotion detection systems is that of Artificial Intelligence. AI software equipped with efficient affect recognition can be used to develop powerful human-computer interaction devices i.e. smarter computers, smarter phones, where these machines can understand the emotions of their human user and respond accordingly. For example, when dealing with speech controlled devices, if a user is getting increasingly annoyed, then the machine can switch to a calmer voice or easier instructions.

With a rise in the number of personal textual outlets such as blogs, microblogs, discussion forums and social networks, people have become increasingly vocal about their feelings and opinions in the virtual world. Deep emotional analysis of such publicly available data could reveal interesting insights about human nature.

1.3 Our contributions

Supervised text classification approaches usually have high accuracies when applied to emotion detection process. However, large annotated data sets are required to build classification models for these techniques to be effective and obtaining such annotated data sets is not a trivial task. Hence unsupervised techniques are preferred in the field of emotion classification.

The problem with existing unsupervised approaches is that although they do not need any annotated data sets, they do rely on affect dictionaries. These affect lexicons are lists of emotion keywords that help in recognizing the emotional affinity of words. But such dictionaries are usually restricted to a fixed number of emotion categories and therefore the unsupervised approaches that employ such dictionaries have a disadvantage that they can classify sentences only into a fixed number of categories that are present in these dictionaries. Our approach does not necessitate the use of any such affect dictionary as emotional tendencies of words are computed using a statistical measure.

The disadvantage of using a typical statistical machine learning approach is that it uses what is called a bag-of-words technique, where words are looked at individually. But words are part of a larger schema such as a phrase or a sentence and it is only natural that the context in which the words appear should be considered instead of treating them in isolation. We incorporate this context-based methodology by measuring the emotional tendency of a word according to the emotional tendencies of its neighbouring words before classifying the sentence.

Therefore, an unsupervised, manually-created affect-lexicon-free approach that does not require any annotated data sets or affect dictionaries before recognizing emotions from text is a more practical solution as compared to a strategy that relies on annotated training examples and a limited set of emotion synonyms listed in an affect dictionary. In addition to this, we also consider the surrounding background of words to better judge their emotional predispositions.

A summary of the contributions of this thesis is as follows:

- We propose a novel unsupervised approach for detecting emotions primarily at the sentence level, which can be extended to document level as well. Unlike supervised methods, our unsupervised technique does not require any annotated data sets for training a model before classification.
- We also do not employ any existing manual affect lexicon, thus allowing our technique to be more generic which can be adopted to fit into any emotional model. We start with a small set of five representative/ seed words which are used to compute relatedness between an affect-bearing word and an emotion concept. This lets our algorithm classify sentences into any number of emotion categories and not just into those listed in the affect dictionary.
- By taking into account the context of a word, our context-based algorithm helps reduce errors that result from the use of pure statistical machine learning approaches.
- Extensive evaluation of our classification framework on various data sets shows promising results when compared to other unsupervised techniques. Moreover, since our approach is independent of any affect dictionary, it can classify sentences from data sets such as ISEAR which have *shame* and *guilt*

categories which, to the best of our knowledge, are not found in any state-of-the-art affect dictionaries.

1.4 Thesis outline

The rest of the thesis is organized as follows. In Chapter 2 we look at a variety of related work and present a literature survey in the field of emotion detection from textual data, including both the supervised as well as unsupervised techniques. Chapter 3 formally introduces the task of emotion detection and describes the details of our proposed algorithm. An extensive set of experiments that thoroughly evaluate the performance of our approach is presented in Chapter 4. Finally, we conclude the thesis and discuss future avenues of research work in Chapter 5.

Chapter 2
Related Work

Researchers in the field of affective computing (Picard, 1995) have been aspiring to develop computational systems that recognize, interpret and represent various affective states (moods and emotions) of the user. Emotions can be expressed through various modalities including face, voice, body language, physiology, brain imaging and text. Recently, textual affect has been gaining increased interest from researchers in the field of natural language processing attempting to recognize sentiment, subjectivity and emotions.

Textual data comprise various composition levels such as word, phrase, clause, sentence, paragraph and document. Research into emotion recognition has been done at all levels, most commonly at the word and sentence levels.

In this section, we begin by presenting an outline of the various affect lexical resources that have been either manually compiled or automatically generated over the years to support emotion recognition applications. We then discuss a variety of methodologies and techniques that have been proposed to identify emotions from textual data. Broadly, most techniques can be grouped into those using machine learning approaches and those relying on linguistic rules. Furthermore, we separate the machine learning approaches into supervised and unsupervised categories. A detailed discussion of the strengths and weaknesses of each approach follows.

2.1 Lexical Resources

To support applications relying on the recognition of textual subjectivity, semantic orientation and affective language, researchers have created different lexical resources.

One of the first such resources was a list of 1,336 adjectives manually labelled as either positive or negative created by Hatzivassiloglou and McKeown (1997). They explored the idea that adjectives that tend to appear together usually have similar semantic orientation. For example, the adjectives “fair” and “legitimate” co-occur very often. Table 2.1 lists some randomly selected positive and negative adjectives from their lexicon.

Table 2.1 Randomly selected adjectives with positive and negative orientations

Positive	Negative
adequate central clever famous intelligent remarkable reputed sensitive slender thriving	contagious drunken ignorant lanky listless primitive strident troublesome unresolved unsuspecting

Aimed at introducing a hierarchy of affective domain labels, Strapparava and Valitutti (2004) created WordNet-Affect, an affective lexical repository of words referring to Ekman’s six basic emotion states (Ekman, Apr 1993). WordNet-Affect extends WordNet (Miller, Beckwith, Fellbaum, Gross, & Miller, 1990) by assigning a variety of affect labels for the concepts related to emotional states, moods, traits, situations evoking emotions or emotional responses, to a subset of synsets[1] representing affective concepts in WordNet as shown in Table 2.2.

Table 2.2 Randomly selected affect concepts from WordNet-Affect

Joy	Anger	Disgust	Fear	Sadness	Surprise
worship satisfaction rejoicing pride lovingness love liking kindhearted joy	wrath umbrage infuriation fury frustration annoyance anger aggravation displeasing	Repugnance nausea disgust nauseating nauseated hideous abhorrent sicken repel	timidity suspense panic intimidation hysteria horror cruelty creeps alarm	woe weepiness sorrow sadness repentance misery melancholy joylessness guilt	admiration amaze astonish awe awful baffle bewilder dazed dumbfound

The subjectivity lexicon developed by Wilson, Wiebe, and Hoffmann (2005) is comprised of over 8,000 single-word subjectivity clues annotated by polarity (positive, negative, both and neutral). Subjectivity clues are words and phrases that may be used to express mostly subjective usages and were derived from (Riloff & Wiebe, 2003) where they were already marked as strongly subjective or weakly subjective. Wilson et al. (2005) expanded this list using a dictionary and a thesaurus and also added potentially subjective words from the General Inquirer positive and negative word lists (The General-Inquirer, 2000). They annotated the polarity of subjective expressions with tags such as positive, negative, both or neutral as shown in Table 2.3. The "positive" tag is for

[1] A synset is a set of synonymous terms as defined in WordNet.

positive emotions (*I'm happy*), evaluations (*Great idea!*), and stances (*She supports the bill*). The "negative" tag is for negative emotions (*I'm sad*), evaluations (*Bad idea!*), and stances (*She's against the bill*). The "both" tag is applied to sentiment expressions that have both positive and negative polarity. The "neutral" tag is used for all other subjective expressions: those that express a different type of subjectivity such as speculation, and those that do not have positive or negative polarity.

Table 2.3 Examples of contextual polarity annotations

Positive	**Negative**	**Both**	**Neutral**
Thousands of coup supporters celebrated overnight, waving flags, blowing whistles…	The criteria set by Rice are the following: the three countries in question are repressive and grave human rights violators…	Besides, politicians refer to good and evil only for purposes of intimidation and exaggeration.	Jerome says the hospital feels no different than a hospital in the states.

Motivated by the assumption that different senses of the same term may have different opinion-related properties, Esuli & Sebastiani (2006) developed SentiWordNet, a lexicon based on WordNet synsets. Three numerical scores (Obj(s), Pos(s) and Neg(s), which range from 0.0 to 1.0 and in sum equal to 1.0), characterizing to what degree the terms included in a synset are Objective, Positive and Negative, were automatically determined based on the proportion of eight ternary classifiers assigning the corresponding label to the synset. For example, the synset [estimable(3)][2], corresponding to the sense "*may be computed or estimated*" of the adjective *estimable*, has an Obj score of 1.0 (and Pos and Neg scores of 0.0), while the synset [estimable(1)] corresponding to the sense "*deserving of respect or high regard*" has a Pos scores of 0.75, a Neg score of 0.0 and an Obj score of 0.25.

The Appraisal lexicon (Argamon, Bloom, Esuli, & Sebastiani, 2007) contains a large number of evaluative adjectives and adverbs annotated by orientation, force and attitude type (affect, judgment, appreciation). It was built by starting with words and phrases derived from (Martin & White, 2005) and finding more candidate terms using WordNet and two online thesauruses. Orientation determines the polarity i.e. whether the term is positive or negative. Force describes the intensity of the term being expressed. Force may be realized via modifiers such as *very* (increased force) or *slightly* (decreased force). Attitude type specifies the type of term being expressed as one of affect, appreciation, or judgment. Affect refers to a personal emotional state (e.g., happy, angry), and is the most explicitly subjective type of appraisal. The two other options differentiate between the appreciation of 'intrinsic' object properties (e.g., slender, ugly) and social judgment (e.g., heroic, idiotic). Figure 2.1 gives a detailed view of the attitude type taxonomy, together with illustrative adjectives (Argamon et al., 2007).

[2] According to the standard convention, the term enclosed in square bracket denotes a synset; thus [poor(7)] refers not just to the term poor but to the synset consisting of {inadequate(2), poor(7), short(4)}. Recall that a synset is a set of synonymous terms which have been grouped together in WordNet.

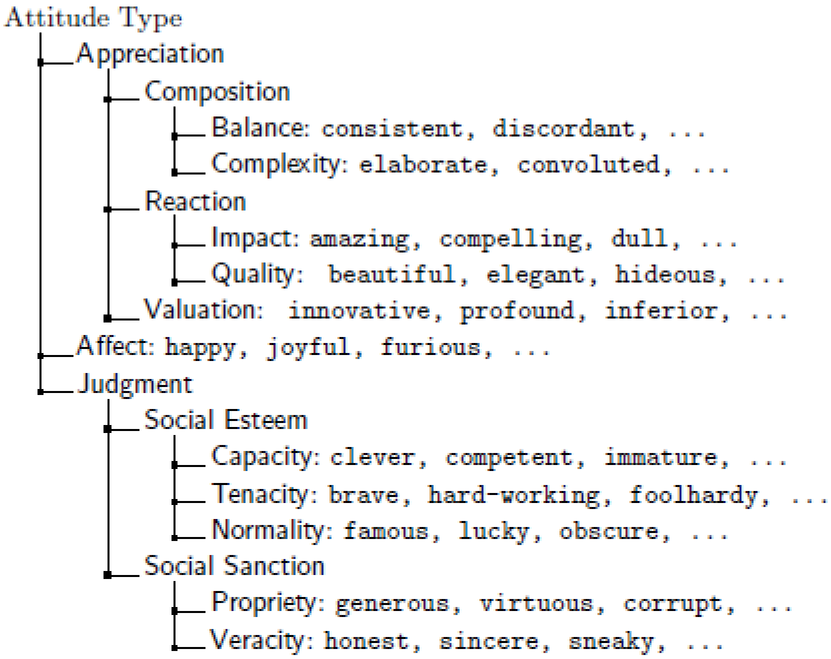
Attitude Type
Appreciation
Composition
Balance: consistent, discordant, ...
Complexity: elaborate, convoluted, ...
Reaction
Impact: amazing, compelling, dull, ...
Quality: beautiful, elegant, hideous, ...
Valuation: innovative, profound, inferior, ...
Affect: happy, joyful, furious, ...
Judgment
Social Esteem
Capacity: clever, competent, immature, ...
Tenacity: brave, hard-working, foolhardy, ...
Normality: famous, lucky, obscure, ...
Social Sanction
Propriety: generous, virtuous, corrupt, ...
Veracity: honest, sincere, sneaky, ...

Figure 2.1 Options in the attitude type taxonomy, with examples of appraisal adjectives, taken from

(Argamon et al., 2007)

An automatically generated polarity sentiment lexicon called SentiFul database, which contains an extensive list of sentiment-conveying adjectives, adverbs, nouns and verbs annotated by sentiment polarity, polarity scores/intensities and weights, was introduced in (A. Neviarouskaya, Prendinger, & Ishizuka, 2009). They employ the Affect database (A. Neviarouskaya, Prendinger, & Ishizuka, 2007) that contains 2438 direct and indirect emotion-related entries which are encoded using nine emotions (*anger, disgust, fear, guilt, interest, joy, sadness, shame and surprise*) and are represented as a vector of emotional state intensities that range from 0.0 to 1.0. They consider three emotions (*interest, joy* and *surprise*) as having mainly positive orientation and six emotions (*anger,* *disgust, fear, guilt, sadness* and *shame*) as negatively-valenced for building the SentiFul lexicon. Table 2.4 gives some examples of words with sentiment annotations from SentiFul.

Table 2.4 Examples of words with sentiment annotations from SentiFul

Affective word	POS	Non-zero intensity emotions from Affect database	Polarity scores		Polarity weights	
			+ Score	- Score	+ Weight	- Weight
tremendous	adjective	‘surprise:1.0’, ‘joy:0.5’, ‘fear:0.1’	0.75	0.1	0.67	0.33
pensively	adverb	‘sadness:0.2’, ‘interest:0.1’	0.1	0.2	0.5	0.5
success	noun	‘joy:0.9’, ‘interest:0.6’, ‘surprise:0.5’	0.67	0.0	1.0	0.0
regret	verb	‘guilt:0.2’, ‘sadness:0.1’	0.0	0.15	0.0	1.0

The Affect database comprises of 364 emoticons of the American and Japanese style (for example, “:”>” and “=^_^=” for ‘blushing’), 337 most popular acronyms and abbreviations, both emotional and non-emotional (for example, “BL” for ‘belly laughing’, “cul8r” for ‘see you later’, and “bc” for ‘because’), 1627 words of type adjective, noun, verb and adverb from WordNet-Affect, interjections such as “alas”, “wow”, and “yay”, and 112 modifiers such as “very”, “extremely”, and “slightly”. Emotion categories were manually assigned to affect-related entries of the database, where intensity values range from 0.0 (very weak) to 1.0 (very strong) as shown in Table 2.5 and Table 2.6.

Table 2.5 Examples of emoticons and abbreviations taken from Affect database

Symbolic representation	Meaning	Category	Intensity
:-)	Happy	Joy	0.6
:-o	Surprise	Surprise	0.8
:-S	Worried	Fear	0.4
PPL	People	-	-

Table 2.6 Examples of affective words taken from Affect database

Affective word	Part of speech	Category	Intensity
cheerfulness	Noun	Joy	0.3
frustrated	Adjective	Anger, Sadness	0.2, 0.7
Dislike	Verb	Disgust	0.4
remorsefully	Adverb	Guilt, Sadness	0.8, 0.5

A novel approach to automatically generate an affect lexicon was introduced by He, Ballard and Gildea (2004). They harvested common sense from the Open Mind Common Sense (OMCS) knowledge base (Singh et al., 2002) using seed words from WordNet and some linguistic patterns such as *subject-verb-object* and *subject-verb-object-objective complement* among others. In their lexicon, the affect of words and phrases is represented as a seven-element vector, one element for each emotion and its value representing that emotion's intensity. The elements are listed in the order of [happy, sad, angry, surprise, frightened, disgusted, interested]. The intensity ranges from the minimum 0 to the maximum 1. A vector may contain multiple non-zero elements, which reflects the fact that textual affect contains ambiguity. A word or sentence is tagged as "neutral" if there is no significant intensity found for the seven emotions. Table 2.7 lists some examples from an automatically generated affect lexicon.

Table 2.7 Examples of emotional words/ phrases from automatically generated affect lexicon

Word/ phrase	Emotion Category	Emotion Intensity
play game	Happy	-
wedding	Happy	-
losing a pet	Sad	-
cemetery	Sad	-
glad	Happy	[1 0 0 0 0 0 0]
cheerful	Happy	[0.5 0 0 0 0 0 0]
funeral	Sad	[0 0.5 0 0 0 0 0]
mourn	Sad	[0 0.25 0 0 0 0 0]
upset	Sad, Angry, Frightened, Disgusted	[0 0.5 0.25 0 0.25 0.25 0]
turn down	Sad, Angry	[0 0.25 0.25 0 0 0 0]

The OMCS project is a commonsense knowledge base which asks visitors to its web page to supply commonsense statements related to all aspects of human life. The main advantage of this approach is that the effort is distributed to a large pool of people from all over the world. However, since the contributors usually use plain natural language, a fair amount of language processing has to be made to recognize and organize commonsense. Some examples of the statements include "Sometimes people weep when they are sad" and "Happy is the opposite of sad". OMCS contains close to half a million sentences in its corpus.

Two other outstanding commonsense knowledge bases include OpenCyc, the largest one, with over 3 million assertions about the world and ThoughtTreasure (Mueller, 2000) that contains 100,000 concepts and relations.

2.2 Affect detection approaches

Affect recognition approaches can be broadly classified into three categories: keyword-based techniques, linguistics rule-based techniques and machine learning techniques (supervised and unsupervised). There could also be a hybrid of some of the previous techniques. One common factor between these techniques is their reliance on or lack of affect lexicons and therefore we subdivide these categories based on the extent of dependence on affect lexicons.

2.2.1 Keyword-based approaches using affect lexicons

Traditionally, keyword-based approaches are applied at the basic word level. If a word exists in an affect lexicon, it is tagged with the emotion category under which it has been listed.

Keyword-spotting techniques have been employed by some researchers (Olveres, Billinghurst, Savage, & Holden, October 1998) to recognize emotion in text. However, the use of a purely word-level analysis model cannot cope with cases where affect is expressed by phrases requiring complex phrase or sentence-level analyses, as words are interrelated and influence each other’s affect-related interpretation.

2.2.2 Linguistic rule-based approaches

Language structures can be complex with innumerable rules and many exceptions to the rules. However, over the years, computational linguists have deciphered these rules to build artificially intelligent systems that are, successfully to an extent, able to “communicate” with humans.

In the field of affect recognition, linguistic rule-based approaches can be further classified according to their reliance on affect lexicons.

2.2.2.1 Linguistic rule-based approaches with affect lexicons

The Emotion Sensing News Agent (ESNA) system (Masum, Prendinger, & Ishizuka, 2007) was developed by defining rules as shown in Table 2.8 for eight emotion types (joy, sad, hope, fear, admiration, reproach, love and hate) and building a polarity lexicon called SenseNet by employing WordNet and ConceptNet lexical resources, to classify emotions engendered by news headlines.

Table 2.8 Definitions of rules for eight emotion types taken from (Masum et al., 2007)

Emotion	Rule
Joy/happy	'pleased' about a 'desirable' event
Distress/sad	'displeased' about an 'undesirable' event
Hope	'pleased' about 'positive' prospect of a 'desirable' 'unconfirmed' event
Fear	'displeased' about 'negative' prospect of 'undesirable' 'unconfirmed' event
Admiration	'pleased' for 'praiseworthy' action/event of other
Reproach	'displeased' for 'blameworthy' action/event of other
Love	'liking' an 'attractive' entity
Hate	'disliking' an 'unattractive' entity

Chaumartin (2007) manually added seed words to emotion lists and propagated their emotions to their neighbour synsets and created a few rules to boost some emotions. UPAR7, their rule-based system, employs a linguistic approach using a syntactic parser to identify what is being said about the main subject of the sentence by exploiting the dependency graph obtained from the parser. They rate each word separately for each emotion and then boost the rating of the main subject. The ratings are obtained from a combination of SentiWordNet (Esuli & Sebastiani, 2006) and WordNet-Affect (C. Strapparava & A. Valitutti, 2004). The system suffers low recall due to the limited coverage of the lexical resources.

The effect of conjuncts in polarity recognition (Meena & Prabhakar, 2007) was studied using rule-based methods over the syntax tree of the sentence and lexical resources such as General Inquirer and WordNet. Lu, Hong and Cruz-Lara (2006) proposed a rule-based *subject-verb-object* approach to classify sentences into emotion categories using adjectives manually assigned to certain emotion categories.

A text-to-emotion engine based on word tagging and analysis of sentences was developed by Riva, Davide, & Boucouvalas (2003). The proposed system employs a parser that generates emotional output only if an emotional word refers to the person himself/herself and the sentence is in present continuous or present perfect continuous tense. As a result, sentences like ‘Onion pie is disgusting’ and ‘It was the most joyous feeling!’ are disregarded by the parser despite the fact that they evidently carry affect. They also employed a specially designed dictionary containing 16,400 words where each word was tagged with one of Ekman’s six emotions.

One of the most recent and effective rule-based approaches has been proposed in (A. Neviarouskaya, Prendinger, & Ishizuka, 2010) to recognize and interpret nine emotion categories (*joy, sadness, anger, disgust, fear, guilt, interest, shame* and *surprise*) as well as the *neutral* category, communicated through informal style of communication such as instant messaging or blogs. Their Affect Analysis Model (AAM) processes each sentence in stages using various rules as shown in Table 2.9, including symbolic cue processing, detection and transformation of abbreviations, sentence parsing and word/phrase/sentence-level analyses. They employ the Affect database (A. Neviarouskaya et al., 2007) to get emotional score vectors for the affective terms.

Table 2.9 Examples of some rules taken from (A. Neviarouskaya et al., 2010)

Condition	Action
With comma and coordinate connectors ‘and’ and ‘so’ (e.g. ‘It is my fault, and I am worrying about consequences’, ‘Exotic birds in the park were amazing, so we took nice pictures’), or with a semicolon with no conjunction	output the vector with the maximum intensity within each corresponding emotional state in the resulting vectors of both clauses.
With coordinate connector ‘but’ (e.g. ‘They attacked, but we luckily got away!’, ‘It was hard to climb a mountain all night long, but a magnificent view rewarded the traveler in the morning.’)	the resulting vector of a clause following after the connector is dominant.
Prepositions such as ‘without’, ‘except’, ‘against’, ‘despite’ cancel vectors of related words (e.g. the phrase ‘despite his endless demonstrations of rude power’ and the sentence ‘I climbed the mountain without fear’)	neutralized due to prepositions.

Although Neviarouskaya et al. (2010) report very high accuracies (up to 77% on diary-like blog posts and 70.2% on fairy tales data set), they use a commercially available syntactic parser and the limitations of a rule-based approach depending on lexical resource such as the Affect database can be noticed in the fact that their system is capable of, and fixed to, recognizing nine emotions. Hence, when running their method on a data set such as the one in (S. Aman & Szpakowicz, 2007) which has been annotated with only six emotions, there arises a need to reduce the number of labels by mapping categories such as ‘interest’ to ‘joy’, and ‘guilt’ and ‘shame’ to ‘sadness’. This mapping, while possible for smaller category changes, becomes difficult to manage when dealing with a larger or completely different set of emotion models.

Most of these approaches do an excellent task of designing rules that can understand complex language structures to recognize underlying fine-grained affect and therefore have high accuracies on emotion classification tasks. However designing such linguistics rules is a time-consuming and effort-intensive process. Moreover, expanding these rules to incorporate different emotion categories is not a trivial task. Similarly, approaches that depend on manually developed affect lexicons suffer from the inflexibility of catering to newer or different emotion categories other than those that have been listed in these affect lexicons.

2.2.2.2 Linguistic rule-based approaches without affect lexicons

As an alternative to using affect lexicons, Liu, Lieberman and Selker (2003) proposed an approach for understanding the underlying semantics of language using the large-scale real-world commonsense knowledge base OMCS. This allows their classification to output any set of emotions such as the ones proposed by Ekman, Frijda, W. James and Plutchik. They incorporated the affect sensing engine into an affectively responsive email composer called EmpathyBuddy. A suite of hand-coded linguistic processing rules including part-of-speech tagging, phrase chunking, constituent parsing, subject-verb-object-object identification and semantic class generalization (e.g. “I” → narrator; “People” → ep_person_class) were designed to generate various affect models from OMCS to extract affective terms called “emotion grounds”. These “grounds” get their initial emotional value from a bag of affect keywords which is then propagated to other concepts.

Another interesting approach has been proposed to recognize emotions from text rich in metaphorical data. Neuman, Kedma, Cohen and Nave (2010) use the Web to harvest metaphorical relations in which the target concept is an argument. For example, they search the Web by using patterns such as “Happiness is like *” where the asterisk ‘*’ is a wildcard, and build a collection of words returned by such searches.

The flexibility of using any set of emotions makes these techniques very practical in terms of expanding to other emotion categories. However, with the current approach, the analysis techniques and knowledge extraction rules are specific to the representation of the knowledge base from which they are extracting the affect knowledge and designing or modifying these rules is not a minor task.

2.2.3 Machine learning approaches

To overcome the limitations faced by linguistic rule-based approaches, researchers have devised some statistical machine learning approaches that can provide as good classification results as obtained by the rule-based methods.

Machine learning algorithms leverage empirical data from large corpora to allow computers to capture unknown underlying probability distribution which can then be used to automatically perform complex pattern recognition. In the field of affect recognition, machine learning approaches can be subdivided into supervised and unsupervised techniques.

2.2.3.1 Supervised machine learning approaches

Supervised machine learning methods (e.g. Support Vector Machines, Naive Bayes in the realm of text classification) generate functions that map inputs to desired outputs by looking at *labelled* training examples. This means that annotated data are required to train the classifiers before they can begin classifying text. On the other hand, unsupervised machine learning methods (e.g. clustering) model a set of inputs without any need for *labelled* training examples.

For classification tasks such as sentence-level affect recognition, approaches that require annotated data sets (sentences labelled with emotions) to train classifiers are grouped under “supervised machine learning approaches”. Techniques that do not require any such annotated data sets are grouped as “unsupervised machine learning approaches”.

2.2.3.1.1 Supervised machine learning approaches with affect lexicons

One of the earliest supervised machine learning approaches was employed by Alm, Roth, and Sproat (2005). They explored the text-based emotion prediction problem empirically in the narrative domain of children’s fairy tales using the SNoW (Sparse Network of Winnows) learning architecture, which is a multi-class classifier. It learns a sparse network of linear functions in which the class labels are represented as linear functions over a common feature space. As they did not have sufficient training data to classify sentences according to fine-grained distinct emotions, in their preliminary study, they focused only on three categories: neutral, positive emotion and negative emotion. Alm (E. C. O. Alm, 2008) later, in her dissertation, described the refined and improved feature set and presented the results of experiments on the fine-grained emotion classification of text using a hierarchical sequential model. She also used SentiWordNet lists and the guess_mood method (now defunct) from the OMCS knowledge base. In order to classify the emotional affinity of sentences in the narrative domain of children's fairy tales, an annotated corpus of 22 Grimms' tales on the sentence level with eight emotion categories (*angry, disgusted, fearful, happy, sad, positively surprised* and *negatively surprised*) was compiled.

The task of classifying blog sentences into Ekman's six basic emotions (*happiness, sadness, anger, disgust, surprise* and *fear*) using a supervised machine learning model Support Vector Machines (SVM) trained on annotated sentences was undertaken by S. Aman and Szpakowicz (2008). Their approach utilizes corpus-based features (unigrams) and the following two emotion lexicons: Roget's Thesaurus (Jarmasz & Szpakowicz, 2001) and WordNet-Affect (C. Strapparava & A. Valitutti, 2004).

The Support Vector Machine classifier was also trained and employed by Ghazi, Inkpen, and Szpakowicz (2010) who arranged neutrality, polarity and emotions "hierarchically" i.e. categories are organized in hierarchies. They tested their method on two datasets and showed that it outperforms the corresponding "flat" approach in (S. Aman & Szpakowicz, 2008). Hierarchical text categorization places new items into a collection with a pre-defined hierarchical structure. The categories are partially ordered, usually from more generic to more specific. For example, sentences can be first classified as emotional or non-emotional. Then, emotional sentences can be further classified into positively emotional and negatively emotional. Finally, positively emotional sentences can be classified into *happiness*, whereas negatively emotional sentences can be classified into *sadness, surprise, fear, disgust* and *anger*. This is an intuitive way of recognizing affect but unless the accuracy at the top level is very high, misclassification errors at the top level can be easily propagated to the lower levels, making this approach slightly more detrimental than profitable.

Using the k-nearest-neighbour approach Tokuhisa, Inui and Matsumoto (2008) classified Japanese sentences using a similarity measure based on cosine similarity. More recently, Quan & Ren (2010) used Probabilistic Latent Semantic Analysis (PLSA) and the supervised machine learning Polynomial Kernel method to recognize emotions. Their work uses the Ren-CECps (a Chinese emotion corpus) as the lexical resource.

Although supervised learning techniques such as SVM and Naive Bayes have been known to perform well in the field of text classification, they have the distinct disadvantage that large annotated datasets are required for training classifiers before classifying data. The process of annotation can be very time consuming and expensive as emotional interpretation of text is usually highly subjective and requires agreement from two or more annotators. Furthermore, in addition to requiring long processing times, classifiers trained on one domain of text generally do not perform so well on another domain. For this reason, unsupervised methods are normally preferred in the realm of Natural Language Processing (NLP) and emotion detection.

2.2.3.1.2 Supervised machine learning approaches without affect lexicons

In the field of sentiment classification, which is a coarse-grained form of emotion classification in the sense that text is classified into positive or negative, Pang, Lee and Vaithyanathan (2002) present a comparison for sentiment classification between three machine learning algorithms (Naive Bayes, Maximum Entropy and Support Vector Machines) tested on a movie review data set and conclude SVM as the best.

The problem of sentiment classification for movie review data sets was also attempted by Martineau and Finin (2009) who used a statistical technique to score words using the Delta TF-IDF function before classifying documents.

Supervised approaches were also applied in Thai emotion classification by Inrak and Sinthupinyo (2010) who used Latent Semantic Analysis (LSA) coupled with Naive Bayes, SVM and Decision tree classification methods to classify text into Ekman's six basic emotions (*anger, disgust, fear, happiness, sadness* and *surprise)*. They build two models – one using only single word occurrence whereas the second using single word combined with bi-words occurrence. Their results demonstrate that the second model can yield higher accuracy than the first model using the Naive Bayes classification method.

Another supervised approach using a Naive Bayes classifier trained on a corpus of blog posts annotated by emotions was exploited by C. Strapparava and Mihalcea (2008).

2.2.3.2 Unsupervised machine learning approaches

To avoid the need for annotated data sets for training supervised classifiers, unsupervised machine learning approaches have been preferred in recent years.

2.2.3.2.1 Unsupervised machine learning approaches with affect lexicons

An evaluation of two unsupervised techniques using the WordNet-Affect affective lexicon for automatically detecting four emotions (anger, fear, joy, sadness) in text has been presented in (Kim et al., 2010). They exploited a Vector Space Model (VSM) in which terms and textual documents are represented through a term-by-document matrix. More specifically, terms are encoded as vectors, whose components are co-occurrence frequencies of words in corpora documents. Frequencies are weighted according to the log-entropy with respect to a *tf-idf* weighting schema (Baeza-Yates & Ribeiro-Neto, 1999). Finally, the number of dimensions is reduced through the dimension reduction methods. The vector-based representation enables words, sentences, and sets of synonyms (i.e. WordNet synsets) to be represented in a unifying way with vectors. They used cosine angle between an input vector (input sentence) and an emotional vector (i.e. the vector representing an emotional synset) as similarity measures to identify which emotion the sentence connotes.

Statistical dimensionality reduction techniques such as Latent Semantic Analysis (LSA), Probabilistic LSA (PLSA) or Non-negative Matrix Factorization (NMF) reduce the computation time and noise in the data by dissipating the unimportant data and making the underlying semantic text to become more patent.

Latent Semantic Analysis (LSA) is the one of the earliest approaches successfully applied to various text manipulation areas (Landauer, Foltz, & Laham, 1998). LSA maps terms or documents into a vector space of reduced dimensionality called the latent semantic space. The mapping of the given terms/document vectors is based on Singular Value Decomposition (SVD) which is a reliable technique for matrix decomposition. It can decompose a matrix as the product of three matrices,

$$A = U \sum V^{T} \approx U_{k} \sum_{k} V_{k}^{T} = A_{k} \qquad (2.1)$$

where A_k is the closest matrix of rank k to the original matrix. The columns of V_k represent the coordinates for documents in the latent space.

Probabilistic Latent Semantic Analysis (PLSA) (Hofmann, 2001) is different from LSA in the sense that it defines proper probability distributions and the reduced matrix does not contain negative values. Based on the combination of LSA and some probabilistic theories such as Bayes rules, the PLSA allows us to find the *latent topics*, the association of documents and topics, and the association of terms and topics. In Equation 2.2, z is a *latent class variable* (i.e. discrete emotion category), while w and d denote the elements of term vectors and document vectors, respectively.

$$P(d, w) = \sum_{z} P(z) P(w|z) P(d|z) \qquad (2.2)$$

where $P(w|z)$ and $P(d|z)$ are topic-specific word distribution and document distribution, respectively. The decomposition of PLSA, unlike that of LSA, is performed by means of the likelihood function. In other words, $P(z)$, $P(w|z)$ and $P(d|z)$ are determined by the maximum likelihood estimation (MLE) and this maximization is performed through adopting the Expectation Maximization (EM) algorithm. For document similarities, each row of the $P(d|z)$ matrix is considered with the low-dimensional representation in the semantic topic space.

Non-negative Matrix Factorization (NMF) (Lee & Seung, October 1999) is another dimension reduction technique successfully applied to semantic analysis. Given a non-negative matrix A, NMF finds non-negative factors W and H that are reduced-dimensional matrices. The product WH can be regarded as a compressed form of the data in A.

$$A \approx WH = \sum WH \qquad (2.3)$$

W is a basis vector matrix and H is an encoded matrix of the basis vectors in Equation 2.3. NMF solves the following minimization problem (Equation 2.4) in order to obtain an approximation A by computing W and H in terms of minimizing the Frobenius norm of the error.

$$\min_{W,H} \left\| A - WH \right\|_F^2 , \text{ s.t. } W, H \geq 0 \qquad (2.4)$$

where $W, H \geq 0$ means that all elements of W and H are non-negative. This non-negative peculiarity is desirable for handling text data that always require non-negativity constraints. The classification of documents is performed based on the columns of matrix H that represent the documents.

Another unsupervised technique involves dimension-based estimation using features derived from ANEW (Affective Norm for English Words) (Bradley, Lang, &

Cuthbert, 1999), a normative database for a collection of 1035 English words where subjects reported their emotions in a three dimensional representation after reading the words. For each word w, ANEW provides coordinates $\overline{w}$ in an affective space as:

$$\overline{w} = (valence\ , arousal\ , dominance\ ) = ANEW\ (w)$$

The occurrences of these words in a text can be used, in a naive way, to weight the sentence in this emotional plane. This approach assumes that an input sentence pertains to an emotion based on the least distance between each other on the Valence-Arousal-Dominance (VAD) space. The VAD value of an input sentence is computed by averaging the VAD values of the words it contains:

$$\overline{sentence} = \frac{\sum_{i=1}^{n} \overline{w}}{n} \tag{2.5}$$

where n is the total number of words in the input sentence.

Since ANEW is a relatively small dataset, a series of synonyms from WordNet-Affect are also used in order to calculate the position of each emotion. These emotional synsets are converted to 3-dimensional VAD space and averaged to produce a single point for the target emotion as follows:

$$\overline{emotion} = \frac{\sum_{i=1}^{k} \overline{w}}{k} \tag{2.6}$$

where k denotes the total number of synonyms in an emotion. The four emotions – anger, fear, joy and sadness, are mapped on the VAD space. Let A_c, F_c, J_c and S_c be the centroids of four emotions. Then the centroids, which are calculated by Equation 2.6, are as follows: $A_c = \langle 2.55, 6.60, 5.05 \rangle$, $F_c = \langle 3.20, 5.92, 3.60 \rangle$, $J_c = \langle 7.40, 5.73, 6.20 \rangle$, and $S_c = \langle 3.15, 4.56, 4.00 \rangle$. Apart from the four emotions, they manually define *neutral* to be $\langle 5, 5, 5 \rangle$. If the centroid of an input sentence is the most approximate to that of an emotion, the sentence is tagged as the emotion (with the nearest neighbour algorithm). The centroid $\overline{sentence}$ might be close to an $\overline{emotion}$ on the VAD space, even if they do not share any terms in common. They define the distance threshold (empirically set to 4) to validate the appropriate proximity like the categorical classification.

The emotion classification experiments conducted by Kim et al. (2010) show that a categorical model such as VSM reduced with Non-negative Matrix Factorization and using WordNet-Affect as a linguistic lexical resource results in better performances for SemEval and fairy tales, whereas a dimensional model performs better with ISEAR.

News headlines were classified by Strapparava and Mihalcea (2008) using several methods ranging from simple heuristics (e.g. directly checking specific affective lexicons) to more refined algorithms (e.g., checking similarity in a latent semantic space in which explicit representations of emotions are built). Two of their unsupervised approaches that use affect lexical resource WordNet-Affect are called WN-A and LSA AEW. WN-A is ‘WordNet-Affect presence’ method, which computes the scores based on the frequencies of the direct affective words found in the headlines. LSA AEW is the ‘LSA all emotion words’ method, which extends the previous set by adding the words from all the synsets labelled with a particular emotion in WordNet-Affect.

Approaches that rely on manually generated affect lexicons are limited to recognizing the emotion categories which exist in these lexicons. Some researchers argue that a different set of emotions is required for different domains. For instance, emotion classes such as *boredom, delight, flow, confusion, frustration* and *surprise* are more appropriate labels in the field of teaching and education instead of the six basic categories *anger, disgust, fear, joy, sadness* and *surprise* found in (Ekman, Apr 1993). Therefore, approaches, supervised or unsupervised, that rely on lexical resources are unable to easily adapt to newer or different emotion categories.

We feel that for a practical system, a more generic unsupervised machine learning approach, which does not rely on a fixed set of emotion categories as specified in an affect lexicon, is a more feasible solution and we discuss some inspiring related work that has been done in this direction in the next section.

2.2.3.2.2 Unsupervised machine learning approaches without affect lexicons

Strapparava and Mihalcea (2008) also proposed two more unsupervised approaches (which did not use manually generated affect lexicon) to classify news headlines. The two unsupervised approaches are called LSA SW and LSA ES. LSA SW is the 'LSA single word' method, which measures the similarity between the given text and each emotion, where an emotion is represented as the vector of the specific word denoting the emotion (e.g. *joy*). LSA ES is the 'LSA emotion synset' method, which uses the synonyms from the WordNet synsets in addition to the word denoting an emotion.

Our approach shares the similar intuition as that demonstrated by the LSA ES method. However, there are some notable differences between the two techniques. First, we use Pointwise Mutual Information to compute the semantic relatedness between the text and each emotion set. Second, we enrich pure statistical machine learning approach by using some context-dependency rules to adjust the emotion score of a word based on the context the word is in.

Similarly, the work presented in (Kozareva, Navarro, Vazquez, & Montoyo, 2007) is a statistical approach using the co-occurrence frequency counts from three web search engines (MyWay, AlltheWeb and Yahoo). They extend the work of (P. D. Turney, 2002) and (P. Turney & Littman, 2003) to classify emotions into six classes instead of a binary classification using not only adjectives but also nouns, verbs and adverbs. Kozareva et al. (2007) use statistics gathered from the three search engines to determine the kind and the amount of emotion in each sentence. Emotion scores are obtained by using Pointwise Mutual Information (PMI). The number of documents obtained from the three search engines using a query that contains all the headline words and an emotion is divided by the number of documents containing only an emotion and the number of documents containing all the headline words to get the PMI score. Although we use the PMI measure of association as well, our approach differs from theirs in several ways. To start with, we compare each relevant word from a sentence to a list of representative words from each emotion category to get a better aggregated association between each word and the emotion category, whereas they compare all the words from the headline (i.e. a phrase) to one emotion word due to long processing and search results return times of a word and emotion pair comparison. We perform this calculation offline as opposed to searching for number of hits using search engines over the Web. And lastly, we adjust the scores of the words based on their context which is gleaned from the syntactic relationships they share with the other words in the sentence.

The aforementioned approaches do not use any affective lexicons. We believe it is beneficial not to rely on an affective lexicon because, besides obvious emotion keywords indicating the presence or lack of emotions, there are words that can potentially convey emotions as an indirect reference. Also, quite often words that bear emotion ambiguity and multiple emotions are difficult to be recognized simply by using affect lexicons. Therefore, emotion of a word should be determined within its context. We propose to test this idea by developing a context-based approach where the emotion vectors of neighbouring words influence (boost or reduce) other words' emotion vectors. Moreover, instead of using just one emotion word for a category, we define a set of closely related synonyms which express an emotion concept. The next chapter describes our methodology in detail.

Chapter 3
Methodology

Textual data are composed of various structural levels such as document, paragraph, sentence, clause, phrase and word. Affect recognition can be done at any of the levels, for different purposes. For example, the document-level recognition is usually performed for news articles to convey the overall emotional tone of the article, whereas sentence-level recognition is useful when deciphering the underlying emotions in an instant messenger chat conversation.

Affect recognition at the sentence level has been increasingly attracting the attention of the research community lately. For many applications, the document-level emotion identification may not be enough and the word-level identification becomes too fine-grained to be of any practical value. Sentence-level classification strikes a fine balance between the two extremes and is beneficial as it can be aggregated to obtain the overall emotional affinity of a document if needed, while helping to understand a deeper analysis of the underlying emotional tones.

A sentence structure is made up of various units such as different parts of speech, negations, modifiers, context of the words and so on, which can provide clues to emotion expression. Using a range of combination of such units can improve the accuracy of the detection process.

In this chapter we propose a novel approach for sentence-level fine-grained emotion recognition using unsupervised statistical machine learning algorithms to obtain emotional affinity of words and some context rules to obtain an emotion vector for the sentence. We explore this task as a multi-class classification problem where one or more nominal emotion labels are assigned to a sentence from a pool of target emotion labels.

3.1 Overview of the framework

Let s be a sentence and ω_s an emotion label. Let e be a set of m possible emotion categories (excluding neutral), where $e = \{e_1, e_2, ..., e_m\}$. The objective of the emotion recognition task is to label s with the best possible emotion label ω_s, where $\omega_s = \{e_1, e_2, ..., e_m, \text{neutral}\}$.

The overview of our emotion recognition framework is shown in Figure 3.1. Broadly speaking, there are four main components: preprocessing module, semantic module, syntactic module and sentence-level analysis. The preprocessing task consists of sentence parsing, parts-of-speech tagging and syntactic dependency tagging. This enables us to extract relevant affect bearing words and syntactic dependencies between them. The next module which performs word-level analysis computes an emotion vector for these affect bearing words by calculating their semantic relatedness to emotion concepts such as *happiness, sadness,* etc. Then the syntactic module which performs phrase-level analysis adjusts the emotion vectors of words computed in the previous step by using context-based information derived from syntactic dependency relationships between words. Finally, the sentence-level analysis module aggregates the emotion vectors of all the relevant words of a sentence to compute the emotion affinity of the whole sentence.

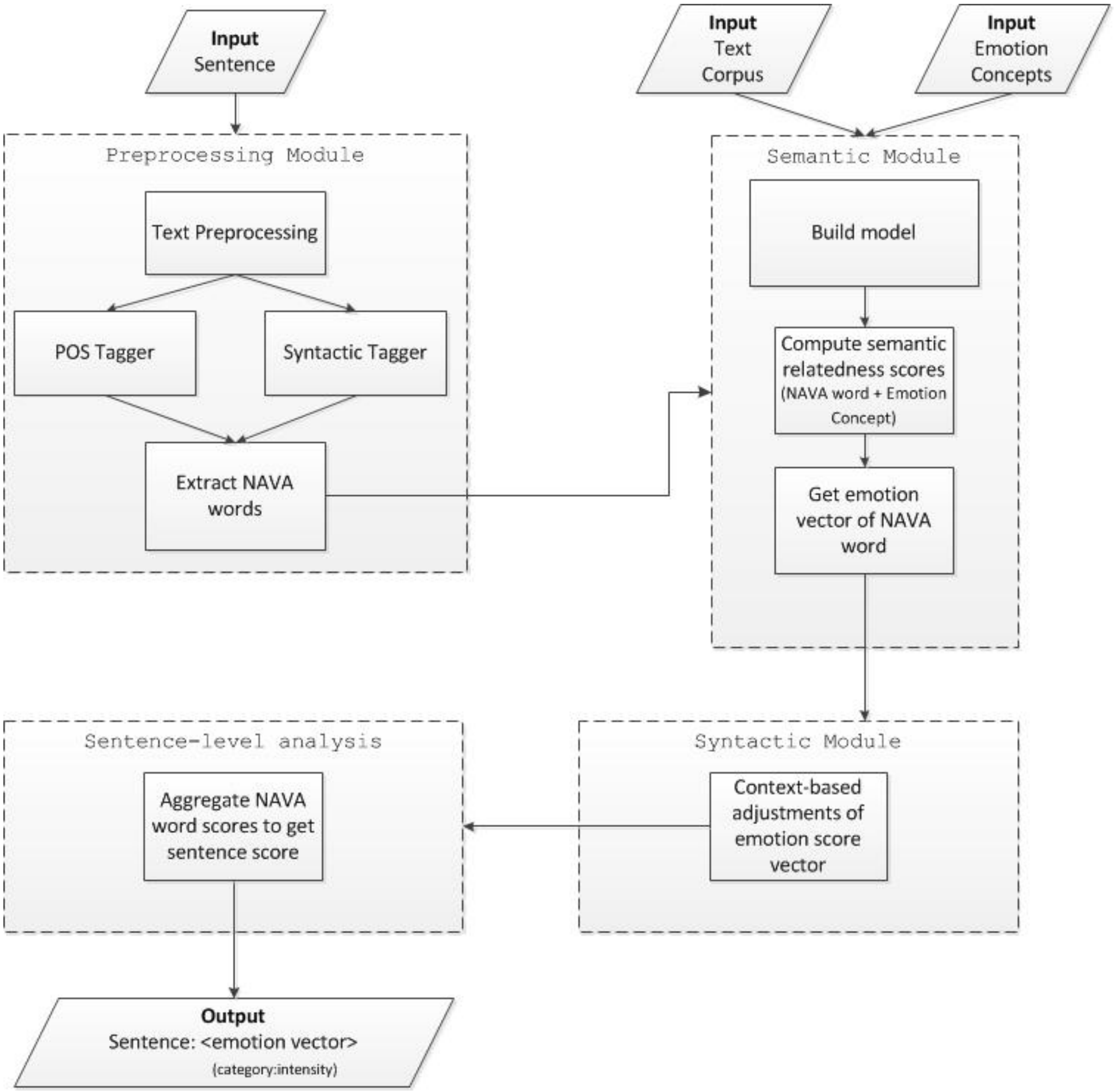

Figure 3.1 Emotion recognition framework

The following sections give a detailed description of each component.

3.2 Extracting affect words

Most sentences contain some words which express affect more apparently than others. Usually, these are the nouns, verbs, adjectives and adverbs classes of a sentence (Polanyi & Zaenen, 2006). Table 3.1 shows some examples of such words which can be readily characterized as having some sort of affect.

Table 3.1 Examples of affect bearing words

	Noun	Verb	Adjective	Adverb
happiness	excitement, joy	to delight, to love	glad, happy	cheerfully, happily
sadness	sorrow, misery	to cry, to regret	unhappy, sad	miserably, woefully
anger	wrath, fury	to anger, to irritate	annoying, frustrating	annoyingly, furiously
fear	horror, fear	to scare, to frighten	afraid, horrified	dreadfully, awfully
surprise	awe, surprise	to amaze, to surprise	amazing, dazed	astonishingly, wonderfully
disgust	disgust, nausea	to sicken, to repulse	distasteful, yucky	repulsively, disgustingly

However, other words of these parts-of-speech (POS) categories may also carry affect, although not obviously when looking at them individually. For example, consider this sentence again: “Izzy got lots of new toys for her birthday yesterday!”. Here, the combination of words such as “new”, “toys” and “birthday” conveys *happiness*. Thus, these words should be used in affect identification. Let us call these potentially affect bearing words as NAVA (Noun Adjective Verb Adverb) words. We begin by extracting a set of such NAVA words from the input sentence. Using Stanford Parser[3], various parts-of-speech of a sentence are tagged and words that have been tagged as noun (NN*)[4], verb (VB*), adjective (JJ*) or adverb (RB*) are extracted from the parsed output.

For example, consider a sentence (1): “It feels sad”. Its POS tag output is shown in Table 3.2.

Table 3.2 Stanford Parser’s POS tag output for sentence 1

Sentence 1: It feels sad.
POS tag: It/PRP feels/VBZ sad/JJ

Words “feels” and “sad”, which have been tagged as a verb (VB*) and an adjective (JJ*) respectively, are the NAVA words in this sentence.

Traditionally, according to some previously tried approaches, these NAVA words are then looked up in some lexical resource such as WordNet-Affect to gauge their emotion affinity and by combining the emotion affinities of all the words, the overall sentence emotion label can be computed.

But consider another sentence (2): “The performers were greeted with joyless cheer.” Its POS tag output is shown in Table 3.3.

[3] Stanford Parser obtained from http://nlp.stanford.edu/software/lex-parser.shtml

[4] Singular and plural proper nouns are excluded to prevent them from influencing the classification process. For example, in the domain of story books, the name Voldemort generally has a negative connotation which is specific to this character. However, in another domain, this word might not have such negative sentiments associated with it and it would be unfair to influence the semantic scores negatively.

Table 3.3 Stanford Parser’s POS tag output for sentence 2

Sentence 2: The performers were greeted with joyless cheer.
POS tag: The/DT performers/NNS were/VBD greeted/VBN with/IN joyless/JJ cheer/NN.

In this case, the NAVA words include “performers”, “greeted”, “joyless”, and “cheer”. Lexical resource such as WordNet-Affect understandably lists the affect word “cheer” under the emotion category *joy*. However, if we look closely, it will be found that in this particular sentence, the emotion affinity of “cheer” is being influenced by the word “joyless” turning the emotion value of the phrase “joyless cheer” to resemble more like the emotion concept of *sadness* than that of *joy*. This is essentially a case of an affect bearing word conveying different emotions depending on the *context* it is used in. Hence, it can be said that “joyless”, which is an adjective here, influences the noun “cheer”. We call “joyless” an influencing word, whereas “cheer” is the dependent word as its emotional vector is dependent on the preceding influencing word.

Conventionally, if this sentence were to be classified using a keyword-based approach, it would consider the words “joyless” and “cheer” to be *sad* and *happy* respectively and cancel out their effect, thus resulting in a possibly neutral sentence. But by using the *context* approach, “joyless” can influence the emotion vector of “cheer” and the ratio switches from 1:1 to say, 1.5:0.5 (if the ratio of influence is taken to be 50%), thus resulting in the final label to be *sad*.

To explore this idea of influencing and dependent words in a sentence, and adjust their emotional vectors accordingly, we propose to exploit the syntactic dependencies in a sentence structure to capture and incorporate some of the contextual information surrounding the NAVA words. The next section discusses the idea of syntactic dependencies further.

3.3 Considering context using syntactic dependencies

Words are embedded in a larger structure such as a sentence and it therefore seems natural to use surrounding contextual emotional expression of words to help inform the classification of the sentence. The context-based model using syntactic dependencies is useful in modeling phenomenon at a level that extends beyond the current bag-of-words approach.

3.3.1 Syntactic dependency

To exploit this context-based model using syntactic dependencies, we first need to select binary syntactic dependencies to be extracted, taking into account the syntactic information of the sentence.

The internal structure of a syntactic dependency consists of two words and the grammatical relationship between them. A dependency is represented as the following binary relation (Gamallo, Gasperin, Agustini, & Lopes, 2001):

$$d\left(w1^{\downarrow}, w2^{\uparrow}\right)$$

where

- the binary predicate d denotes a specific syntactic grammatical relation such as nominal subject, direct object, negation, adjectival modifier, etc.;
- arrows "$\downarrow$" and "$\uparrow$" represent the *modified* and *modifier* position, respectively (or vice versa, depending on the convention used);
- $w1$ and $w2$ represent the two syntactically related words, where $w1$ is the word in the *modified* position (which we call the dependent word), and $w2$ is the word in the *modifier* position (which we call the influencing word).

Binary syntactic dependencies denote grammatical relationships between the *modified* (dependent word) and its *modifier* (influencing word). Although in semantic terms, both words seem to be interdependent, these relationships are asymmetric, usually directed and sometimes typed to distinguish between different kinds of dependencies. We utilize the Stanford POS tagger to generate such *typed dependencies*. Some of the dependencies in Stanford representation are shown in Table 3.4.

The representation describes the grammatical relationships in a sentence. The definitions make use of the Penn Treebank POS tags and phrasal labels (Marneffe & Manning, 2010). These kinds of elaborate representations provide us with fine-grained syntactic contexts. In the following section, we will select some of these dependencies to be extracted.

Table 3.4 Examples of syntactic dependencies in Stanford typed dependencies format

Syntactic dependency	Example	Representation
Abbreviation modifier	"The Australian Broadcasting Corporation (ABC)"	abbrev(Corporation, ABC)
Adjectival complement	"She looks very beautiful"	acomp(looks, beautiful)
Conjunct	"Bill is big and honest"	conj(big, honest)
Negation modifier	"Bill doesn't drive"	neg(drive, n't)
Prepositional modifier	"I saw a cat with a telescope"	prep(saw, with)

3.3.2 Which syntactic dependencies to use?

We begin by focusing on three types of typed dependencies – adjectival complement, adjectival modifier and negation modifier.

An adjectival complement of a verb is an adjectival phrase which functions as the complement, like an object of the verb. For example, one of the dependencies from the sentence "She looks very beautiful" is represented as *acomp(looks, beautiful),* where 'beautiful' is the adjectival complement of the verb 'looks'. Mapping this to binary dependency relation notation $d\left(w1^{\downarrow}, w2^{\uparrow}\right)$, 'looks' becomes the dependent word and 'beautiful', the influencing word.

An adjectival modifier of a noun phrase is any adjectival phrase that serves to modify the meaning of the noun phrase. For example, one of the dependencies from the sentence "Sam eats red meat" is *amod(meat, red)*, where 'red' is the adjectival modifier of the noun ‘meat’. Therefore we can say that ‘meat’ is the dependent word in this case and ‘red’, the influencing word.

A negation modifier is the relation between a negation word and the word it modifies. For example, one of the dependencies from the sentence “She is not happy today” is *neg(happy,not)*, where the adverb ‘not’ is the negation modifier of the word ‘happy’. Therefore, we can say that ‘happy’ is the dependent word in this case and ‘not’, the influencing word.

Currently, we use only these three dependencies as both the words contained in these relationships belong to the NAVA word set and the relationship between them is strong. For other dependencies, the influence of one word over another is not so clear and they might include other parts of speech such as prepositions, conjunctions and so on which have been excluded from the NAVA word set.

The need for considering the appropriateness of such dependencies in a context-based model can be empirically provided by the following example in Table 3.5. Consider again sentence 2: “The performers were greeted with joyless cheer”, where the dependent word “cheer” would be related to the emotion concept of *joy* as per WordNet-Affect, but its emotional connotation changes to *sadness* when its influencing word “joyless” is considered.

Table 3.5 Stanford Parser's syntactic dependencies output for sentence 2

Sentence 2: The performers were greeted with joyless cheer.	
POS tag:	**Syntactic dependencies:**

The/DT competitors/NNS were/VBD greeted/VBN with/IN joyless/JJ cheer/NN.	det(competitors, The)
	nsubjpass(greeted, competitors)
	auxpass(greeted, were)
	prep(greeted, with)
	amod(cheer, joyless)
	pobj(with, cheer)

Note that the syntactic dependency adjectival modifier *amod(cheer, joyless)* represents the scenario where the emotion vector of the word "joyless" influences the emotion vector of the word "cheer".

Consider another example sentence, "The drunk kids turned it into a disgusting party". The syntactic dependencies of this sentence are shown in Table 3.6.

Table 3.6 Stanford typed syntactic dependencies for example sentence

The drunk kids turned it into a disgusting party.	
POS tag:	**Syntactic dependencies:**

The/DT drunk/JJ kids/NNS turned/VBD it/PRP into/IN a/DT disgusting/JJ party/NN.	det(kids, The)
	amod(kids, drunk)
	nsubj(turned, kids)
	dobj(turned, it)
	prep(turned, into)
	det(party, a)
	amod(party, disgusting)
	pobj(into, party)

In this example, there are two *amod* relations. In the *amod(kids, drunk)* dependency, the word “drunk” influences the vector of the word “kids” and in the *amod(party, disgusting)* relation, the word “disgusting” can be considered as influencing the vector of the word “party”. As it can be noticed, generally on their own, the words “kids” and “party” would not exhibit any negative sentiments, and are usually rather positive. However, in this case, due to the influence of words such as “drunk” and “disgusting” over the words “kids” and “party” respectively, the latter also start to show some negative emotions in this context.

It will be observed that two of the syntactic dependency relationships that we use, namely *adjectival modifier* and *adjectival complement*, represented as *amod* and *acomp* respectively, are either of the form (noun, adjective) where the adjective is the influencing word of the dependent word noun or of the form (verb, adjective) where the adjective is the influencing word of the dependent word verb. This is different from simply considering the adjectives of the sentence and adjusting their emotion vector. What we are specifically looking for are the interesting adjective-noun and adjective-verb pairings and adjusting the emotion vector of the dependent noun or verb based on the emotion vector of the influencing adjective. The syntactic dependency structure of the sentence helps us derive this handy adjective-noun and adjective-verb relationship.

The other dependency i.e. negation, has the form (verb, adverb) or (noun, adverb) where the negative adverb modifies the dependent verb or noun. Negation is one of the most obvious and widely used processes in linguistics which turns an affirmative statement into its opposite or neutral.

In the next section, we define what an emotion vector of a NAVA affect word is and how it is calculated, before moving onto describing the emotion vector of a sentence which is derived by aggregating emotion vectors of various affect words including dependent and influencing words.

3.4 Representing emotion as a vector

NAVA words carry certain emotional value, some stronger than others. Instead of defining, say, “happiness” as one emotion, we believe it is more appropriate to call it an emotion concept, made up of smaller emotions such as “gladness”, “content”, “excitement”.

Some words such as “angry” have a very strong bias towards one type of emotion concept such as “anger” but most words fall under a multi-emotion category. For example, the word “birthday” taken as is (without context) could carry a combination of some emotion value for “happiness” or “surprise” or even “sadness” (if it was spent alone). Therefore, it is important to calculate a vector of emotions for each NAVA word with varying intensities for each emotion.

Formally, an emotion vector of a NAVA word can be defined as a vector of scores where each value represents the strength of the affinity of the word for an emotion category. Suppose that we use Ekman’s model which contains six basic emotion categories – *happiness, sadness, anger, fear, surprise* and *disgust*. Then, the emotion vector of a NAVA affect word consists of a six-valued vector notation represented in the form of $\sigma_{\text{NAVA\_word}} = \langle \text{happiness, sadness, anger, fear, surprise, disgust} \rangle$. For example, the emotion vector of "birthday" may look like $s_{\text{birthday}} = \langle 0.25, 0.01, 0.01, 0.0, 0.16, 0.0 \rangle$

An advantage of using vectors, which we can exploit later, is that they allow for expressing multiple emotions. This is applicable to any textual composition level i.e. we can have an emotion vector of a word or a sentence or a document and so on.

Traditionally the emotion vector of a word is calculated by directly matching it against an affect dictionary such as WordNet-Affect and checking if the word exists in one of the emotion categories. One of the shortcomings of this approach is that it cannot detect emotion from sentences that do not contain any obvious emotional keywords. However, emotion can be expressed by sentences that do not contain any obvious emotional keywords and therefore, we need to use a more robust detection method.

For example, consider sentence 3: "That is nonsense". Clearly, it sends out a very angry vibe but unless the word "nonsense" exists in the affect lexicon (which it does not in WordNet-Affect), it would be difficult for a system that relies on such manually created lexical resources to identify the emotional connotation of words/sentences to detect the emotion label for this sentence. Let us look at another sentence 4: "They walked away together, singing". Again, this sentence does not contain any obvious emotional keyword but still falls under the *happiness* category, presumably because of the word "singing". As "singing" is not generally found in the commonly-used affect lexicon, its emotion content will be overlooked, thus leading to misclassification of the sentence.

To this effect, we propose to use the concept of semantic relatedness where words are compared against each other over a large text corpus, based on which an emotional affinity score is derived to determine how semantically related a NAVA word is to an emotion category. The next section discusses the measure used for computing the semantic relatedness between two words.

3.5 Semantic relatedness between two words

As per Hatzivassiloglou and McKeown (1997), adjectives with the same polarity tend to appear together. We propose to extend this idea further to determine if affect words including adjectives along with nouns, verbs and adverbs in general, that co-occur together frequently have the same polarity or emotional conceptual tendency.

Co-occurrence, specifically in a linguistic sense, can be defined as an above-chance frequent occurrence of two terms, stems or concepts from a text corpus alongside each other in a certain order. It assumes interdependency of the two terms and can be interpreted as an indicator of semantic proximity. Basically, if two words co-occur more frequently, then they tend to be semantically related. For example, the word “snowboarding” immediately makes us think of “winter”, not of “summer” or “autumn”. Similarly, “singing” sends out happy vibes, whereas “nonsense” is more closely related to angry feelings. This is known as semantic relatedness between two words.

Measures of semantic relatedness provide models of human semantic associations and, as such, have been applied to predict human text comprehension (Lemaire, Denhière, Bellissens, & Jhean-Larose, 2006), modeling language acquisition (Landauer & Dumais, Apr 1997), human web-browsing behaviour (Fu & Pirolli, November 2007), semantic maps (Veksler & Gray, 2007) and many other applications. These measures have been used to develop a wide variety of applications such as augmented search engine technology (Dumais, 2003) and automated essay-grading algorithms for the Educational Testing Service (Landauer & Dumais, Apr 1997). The techniques that use semantic relatedness try to learn word relations from text corpus and consequently their effectiveness is dependent on the text corpus from which they glean information. The measures give computers the ability to quantify the meaning of text by defining words in terms of their connection strengths to other words, and they define connection strengths in terms of words co-occurrence. In other words, two terms are related if they often occur in the same contexts (Lindsey, Veksler, Grintsvayg, & Gray, 2007).

The motivation behind scoring words comes from the need to know the general lexical affinity of a word to a particular emotion concept. We aim to determine if, for example, the word “thunderstorm” indicates *fear* whereas the word “party” invokes a feeling of *happiness* or *surprise*.

There are various models for measuring semantic relatedness and although the models differ considerably in the algorithms they use, they are all fundamentally based on the principle that a word’s meaning can be induced by observing its statistical usage across a large sample of language. Typically, the first step is to train a model on a corpus of text, after which the model generates semantic relatedness scores between a pair of words. Ideally, these semantic relatedness scores should closely correspond with human ratings. For example, if humans rate the words “birthday” and “party” as more similar than the words “birthday” and “hurricane”, then it is desirable for a semantic space model to do so as well (Recchia & Jones, 2009).

Several different methods and their variations have been proposed for estimating a semantic relatedness score between a pair of words. We explored a few of these options and chose to use the Pointwise Mutual Information as discussed in the following subsection.

3.5.1 Pointwise Mutual Information

Pointwise Mutual Information (PMI) is a simple information-theoretic measure of semantic relatedness that measures the similarity between two terms by using the probability of co-occurrences. It was first introduced in the context of word associations (Church & Hanks, 1990).

Mathematically, simple PMI between two words *x* and *y* in a corpus is calculated using Equation 3.1.

$$PMI\ (x, y) = \frac{\text{occurrence}\ (x, y)}{\text{occurrence}\ (x)\text{occurrence}\ (y)} \tag{3.1}$$

where occurrence(*x*) is approximated as the number of times that a word *x* appears in the corpus, occurrence(*y*) as the number of times word *y* appears in the corpus, and occurrence(*x*,*y*) as the number of times the two words *x* and *y* co-occur within a specified number of words of each other in the corpus. For our experiments, we use a window size of sixteen words as previous findings report that counting co-occurrences within small windows of text produces better results than does counting co-occurrences within larger contexts (Bullinaria & Levy, 2007).

Being a measure of the degree of statistical dependence between two words, the purpose of PMI is to determine how closely two words are related and this forms the basis of PMI's application to semantic relatedness scoring where we compare the affect word to a set of emotion concepts such as *happiness, sadness, anger* and so on.

Another motivation for using PMI as a measure of semantic relatedness stems from the results of experiments run by Recchia and Jones (2009). They found that PMI, which is a scalable, incremental and simple measure of semantic similarity, greatly benefits from training on large corpus of data and can outperform a commonly used version of LSA trained on the TASA corpus. For five out of six tests they conducted on similarity relatedness judgements, the model built on Wikipedia using PMI was the second highest performing measure, outperformed solely by the model built using the WordNet similarity vector measure. PMI trained on Wikipedia was the highest performing measure on the remaining test.

We choose to use PMI instead of the WordNet similarity vector measure for one big reason – the WordNet similarity vector measure is a model based on hand-coded intelligence and is limited to the words in the WordNet lexicon since it relies heavily on

WordNet's dictionary glosses. Therefore, as impressive as WordNet's performance is on approximations of semantic similarity between English nouns, since our approach needs to cater to various other parts-of-speech and involves words not included in the WordNet hypernym hierarchy, PMI trained on a suitably large corpus seems to be a more feasible approach. This is interesting from a practical standpoint as PMI is fast and easy to calculate even on huge datasets.

For our experiments, we use the freely available version of Lightweight Models of Semantic Similarity[5] to train the PMI models. The models were trained on three text corpora (to investigate the effects of different type of corpora on the emotion detection task):

- Wikipedia: The Wikipedia data were collected from the data dump[6] dated 5 April 2011, which was converted into text using WP2TXT v 0.3.0[7].
- Gutenberg: Project Gutenberg is a collection of over 36,000 free ebooks. We wonder if this type of data which intuitively seems to be more "emotional" than the objective data found on Wikipedia, would provide more relevant scores when computing semantic space models containing emotion concepts. We decide to explore this idea by downloading the ebooks in zipped files, collected using Wget[8].

[5] LMOSS, available at www.indiana.edu/~clcl/LMOSS/

[6] http://download.wikimedia.org/enwiki/latest/enwiki-latest-pages-articles.xml.bz2

[7] http://wp2txt.rubyforge.org/

[8] http://www.gnu.org/s/wget/

- And what if the two corpora were combined? We go ahead to investigate this by merging the Wikipedia and Gutenberg corpora into the Wiki-Guten corpus.

3.5.2 Stemming

Intuitively, it feels that before one starts computing semantic similarity scores using PMI trained on a large text corpus, stemming should be performed on the text corpus. This is a process for reducing words to their stem, base or root form. The advantages of stemming could be two-fold: firstly, this would enable us to treat words with the same stem as synonyms as a kind of query broadening, a process called conflation; secondly, it would help to keep the learned model space relatively small as well as reduce computation time by reducing the noise in the data. We use the Porter Stemmer[9] to investigate if stemming the text corpus can have any positive effect on the emotion detection task.

In the next two sections we describe how to first compute the emotion vector of a NAVA word and then aggregate these scores to calculate the emotion vector and therefore the final emotion label of the sentence.

3.6 Calculating emotion vector of a NAVA affect word

Let us begin by defining some notations.

Let w be a set of n NAVA affect words of a sentence s, where $w \subset s$ and $w = \{w_1, w_2, ..., w_n\}$.

[9] http://tartarus.org/~martin/PorterStemmer/index.html

Let α be a set of c influencing affect words, where $\alpha \subset w$ and $\alpha = \{\alpha_1, \alpha_2, ..., \alpha_c\}$.

Let β be a set of d dependent affect words, where $\beta \subset w$ and $\beta = \{\beta_1, \beta_2, ..., \beta_d\}$.

Let e be the set of m possible emotion concepts that a sentence can be classified into, where $e = \{e_1, e_2, ..., e_m\}$. For example, if we choose to classify a sentence using Ekman's model of six basic emotions[10], then $e = \{\text{happiness, sadness, anger, fear, surprise, disgust}\}$.

The following two subsections describe how to first compute the emotion vector of a NAVA word and then how to adjust these scores by incorporating the context information gleaned from syntactic dependencies within the sentence.

3.6.1 Computing emotion vector of a word without context information

We aim to derive the emotion vector σ_{wi} for each NAVA word w_i in a sentence, where each score in the vector represents an affinity of w_i to an emotion concept, as defined in Section 3.4. A simple solution is to use the PMI score between w_i and the word representing an emotion concept (such as happiness) as the affinity score of w_i to the emotion concept. However, since an emotion concept can often be expressed through different words (e.g., glad or joy for happiness), we would like to test a hypothesis that an emotion category is more accurately and neatly defined by a set of words rather than just one generic word representing a whole emotion category concept. If a word belongs to an emotion category, it will be closely related to most of the representative words that

[10] six emotions or neutral label which is calculated using a cut-off threshold explained in 3.7.

comprise that emotion concept instead of a random off-chance association with one word. Thus, we define a set of representative words to represent an emotion category and use the PMI scores between w_i and each of representative words in an emotion category to compute the affinity score of w_i to the category. Let K_j be a set of r representative words for emotion concept e_j. The semantic relatedness score between an affect word w_i and an emotion category e_j is calculated as follows:

$$PMI\,(w_i, e_j) = \sqrt[r]{\prod_{g=1}^{r} PMI\,(w_i, K_{j_g})} \tag{3.2}$$

where an affect word w_i is compared with each representative word K_{j_g} of an emotion concept e_j, *PMI*(w_i, K_j) is the PMI score between w_i and K_j, and the geometric mean of the scores gives the semantic relatedness score between the affect NAVA word and the emotion concept category. Geometric mean was chosen for its usefulness in indicating the central tendency of a set of numbers as opposed to arithmetic mean or harmonic mean which is biased towards the greatest of the values or the least of the values respectively.

We experimented with different numbers of representative words and based on the empirical results, we decided to choose five representative words. The advantage of having a small set of synonyms makes it easier to choose the words without any trade-offs on accuracy. Examples of representative words for some emotion concepts are listed in Table 3.7.

Table 3.7 Sample set of representative words for some emotion concepts

Emotion concept	Representative words
Happiness	happy, glad, joy, good, love
Sadness	sad, sorrow, hurt, cry, bad
Anger	angry, irritate, stupid, annoy, frustrate
Fear	fear, afraid, frighten, scare, terrify
Surprise	surprise, amazing, astonish, incredible, wonder
Disgust	disgust, dislike, hate, sick, ill
Guilty	guilt, ashamed, blame, dishonour, criminal

Note that using representative words is different from consulting an affect dictionary in the sense that we only need a small, fixed number of representative words for each emotion concept. Obtaining such representative words and expanding them to more categories is easier than constructing a whole lexicon or dictionary.

Using Equation 3.2, semantic relatedness scores of a NAVA word are calculated for all the emotion concept categories and finally an emotion vector s_{w_i} of the affect NAVA word w_i is derived, which is represented as:

$$\sigma_{w_i} = \left\langle PMI\ (w_i, e_1), PMI\ (w_i, e_2),..., PMI\ (w_i, e_m)\right\rangle \tag{3.3}$$

Recall sentence 3: “That is nonsense”. The emotion vector of the word “nonsense” in the six emotions format derived using PMI on Wiki-Guten (section 3.5.1) would be:

$$\sigma_{\text{nonsense}} = \langle PMI(\text{nonsense, happiness}), PMI(\text{nonsense, sadness}), PMI(\text{nonsense, anger}), PMI(\text{nonsense, fear}), PMI(\text{nonsense, surprise}), PMI(\text{nonsense, disgust}) \rangle$$

$$\sigma_{\text{nonsense}} = \langle 0.041, 0.0606, 0.123, 0.074, 0.050, 0.054 \rangle$$

where “nonsense” has the highest relatedness score with the *anger* category.

Similarly, consider sentence 4: “They walked away together, singing”. The emotion vector of “singing” is $\sigma_{\text{singing}} = \langle 0.103, 0.06\ 5, 0.037, 0.\ 027, 0.042,\ 0.015 \rangle$ where “singing” strongly correlates with the feeling of *happiness*.

3.6.2 Adjusting emotion vector of a word using context information

As discussed in section 3.3.2 earlier, we use the syntactic dependencies of a sentence structure to see which influencing words sway the emotional affinity of dependent words to one emotion concept category or another. Here we describe how we adjust the emotion vector of the dependent word based on the influencing word. There are two ways of fine-tuning the emotion vector of the dependent word based on the syntactic dependency it is part of.

3.6.2.1 For adjectival modifier and adjectival complement relations

One of the ways to adjust the scores of the dependent word is by weighing the dependent word based on some percentage of the scores of the influencing word. This can be done by taking 50% of the dependent word’s score and adding to it 50% of its influencing word’s score i.e. combining the two words and averaging them. Using Equation 3.2, we first obtain emotion vectors for the influencing word s_{a_p} and the dependent word s_{b_q}. Then using Equation 3.4 below, we adjust the emotion vector of the dependent word. The new emotion vector of the dependent word is:

$$s_{b_{q'}} = \frac{s_{b_q} + s_{a_p}}{2} \tag{3.4}$$

Consider again the sample sentence (2): "The performers were greeted with joyless cheer", where the word "joyless" is the influencing word and "cheer", the dependent word. To demonstrate the score adjustment, let us use the Ekman's model of six emotions – *happiness, sadness, anger, fear, surprise* and *disgust*. Maintaining the order of emotions, suppose the emotion vector of "joyless" obtained from semantic relatedness computation is $\sigma_{\text{joyless}} = \langle 0.20, 0.85, 0.19, 0.10, 0.02, 0.30 \rangle$ which has a strong affinity towards *sad*ness and the emotion vector of "cheer" is $\sigma_{\text{cheer}} = \langle 0.78, 0.20, 0.02, 0.03, 0.29, 0.18 \rangle$ which is strongly biased towards *happiness*. The adjusted emotion vector of the dependent word "cheer" therefore would be $\sigma_{\text{cheer'}} = \langle 0.49, 0.53, 0.11, 0.07, 0.16, 0.24 \rangle$, which shows almost equal preference towards *happiness* and *sadness* categories. We can see possibly two advantages of performing this emotion score adjustment: one is that "cheer" gains some weight from its influencing word "joyless" as it exhibits some emotion value for *sad* now; secondly, "cheer" is no longer strongly related to *happiness*, thereby reducing the chances of the sentence being mislabelled as *happiness*. So when the emotion vectors of these words along with other NAVA words of the sentence will be combined to output the aggregated sentence level emotion vector, “cheer” will not bias the scores towards *happiness* and the sentence can be classified as *sad*.

3.6.2.2 For negation relations

Consider the sentence: “She is not sad”. The negation dependency relation is of the form (sad, not) where ‘sad’ is being negatively modified by ‘not’. In this case, one of the ways of adjusting the dependent word’s score is to cancel it out, i.e., set it to zero. This way, the word ‘sad’ becomes neutral and does not contribute to the overall emotion vector of the sentence. One may ask why we cancel out the score instead of say, reverting the strongest emotion (*sadness* in this case) to its counterpart i.e. *happiness*. The reason is that easy as it may be to understand that ‘not sad’ might imply *happiness* in this case, it may not always be as simple as not every emotion has its direct reverse. ‘Not sad’ could simply mean *not sad*, therefore expressing a *neutral* state. Consider another sentence: “She is not afraid of the dark”, with its negation dependency (afraid, not). The decision to revert the scores in this case is not as straight-forward. Hence, we choose to set the scores of each dependent word of a negation modifier relationship to zero.

3.7 Calculating emotion vector of a sentence

Putting it all together, the emotion vector of a sentence can be computed by aggregating the emotion vector scores of all the affect words of that sentence and averaging it by the total number of affect words:

$$\sigma_s = \frac{\sum_{i=1}^{n} \sigma_{w_i}}{n} \tag{3.5}$$

In Equation 3.5, w_i is the NAVA affect word, n is the number of such affect words and σ_s is a vector of size *m* consisting of average context-based emotion scores of all the affect words in the sentence for each emotion category, where *m* is the number of emotion categories.

After obtaining the emotion vector $\sigma_s = \langle s_1, s_2, ..., s_m \rangle$ of a sentence s using Equation 3.5, if the highest final emotion score in the vector is above a certain threshold *t*, the sentence is labelled with that emotion. Otherwise, it is classified as neutral. That is, the emotion label ω_s of sentence *s* is computed as shown in Equation 3.6:

$$\omega_s = \begin{cases} e_k & \text{if } \max\limits_{i=1,..,m} (s_i) = s_k \text{ and } s_k \geq t \\ \text{" neutral"} & \text{otherwise} \end{cases} \tag{3.6}$$

To see the usefulness of adjusting the score vectors according to the context, let us consider the sentence “The drunk kids turned it into a disgusting party” again. The unadjusted scores for each NAVA word would be as shown in Table 3.8:

Table 3.8 Unadjusted emotion scores for sample sentence

NAVA word	Unadjusted Emotion vector <happiness, sadness, anger, fear, surprise, disgust>
drunk	$\sigma_{\text{drunk}} = \langle 0.06, 0.04, 0.09, 0.02, 0.03, 0.07 \rangle$
kids	$\sigma_{\text{kids}} = \langle 0.06, 0.03, 0.04, 0.02, 0.02, 0.01 \rangle$
turned	$\sigma_{\text{turned}} = \langle 0.01, 0.01, 0.02, 0.01, 0.01, 0.01 \rangle$
disgusting	$\sigma_{\text{disgusting}} = \langle 0.10, 0.40, 0.70, 0.20, 0.10, 0.80 \rangle$
party	$\sigma_{\text{party}} = \langle 0.05, 0.02, 0.02, 0.01, 0.04, 0.01 \rangle$
Sentence vector/label	$\sigma_{s} = \langle 0.07, 0.10, 0.18, 0.05, 0.05, 0.18 \rangle$ anger or disgust?

As we can see from the example, the scores for *anger* and *disgust* almost tie-up, which could lead to misclassification of the sentence. Now, let us adjust the vector scores to see what happens.

Table 3.9 Adjusted emotion scores for sample sentence

NAVA word	Adjusted Emotion vector <happiness, sadness, anger, fear, surprise, disgust>
drunk	$\sigma_{\text{drunk}} = \langle 0.06, 0.04, 0.09, 0.02, 0.03, 0.07 \rangle$
kids	$\sigma_{\text{kids}} = \langle 0.06, 0.04, 0.07, 0.02, 0.03, 0.04 \rangle$
turned	$\sigma_{\text{turned}} = \langle 0.01, 0.01, 0.02, 0.01, 0.01, 0.01 \rangle$
disgusting	$\sigma_{\text{disgusting}} = \langle 0.10, 0.40, 0.70, 0.20, 0.10, 0.80 \rangle$
party	$\sigma_{\text{party}} = \langle 0.08, 0.21, 0.36, 0.11, 0.07, 0.41 \rangle$
Sentence vector/label	$\sigma_{s} = \langle 0.06, 0.14, 0.24, 0.07, 0.05, 0.27 \rangle$ disgust

Notice that after adjusting the scores, there is a clear difference between the scores of *anger* and *disgust* which makes it convenient to label the sentence as *disgust* with more confidence.

3.8 Summary of the algorithm

The emotion detection paradigm can be summarized into pseudocode as shown in Algorithm 1:

Algorithm 1

```
Input:
 A sentence $s$
 Set of emotion categories $e = \{e_1, e_2, ..., e_m\}$
 Set $K_j$ of $r$ representative words for each emotion category $e_j$
 Neutral threshold $t$
Output:
 Affect label $\omega_s$ for $s$
Procedure:
   let set $w$ be empty
   tag sentence $s$ with tagger to obtain the POS tag for each word in $S$
      and the set $R$ of all relevant syntactic relations in $S$
   for each word $w_i$ in $S$
       if POS tag of $w_i$ ∈ {noun, adjective, verb, adverb} then
           insert $w_i$ into $w$
           for each emotion concept category $e_j$ do
               $PMI(w_i, e_j) = 1$
               for each representative word $K_{j_g}$ do
                   compute semantic relatedness score $PMI(w_i, K_{j_g})$
                   $PMI(w_i, e_j) = PMI(w_i, e_j) \times PMI(w_i, K_j)$
               end for
               $PMI(w_i, e_j) = \sqrt[r]{PMI(w_i, e_j)}$
           end for
           get emotion vector of $w_i$
```

```
$\sigma_{w_i} = \langle PMI(w_i, e_1), PMI(w_i, e_2), ..., PMI(w_i, e_m) \rangle$
    end if
end for
for each dependent word $\beta_q$ in relation set $R$ do
     get $\beta_q$'s influencing word $\alpha_p$
     if syntactic relation ∈ {amod, acomp} then
         adjust the emotion vector of $\beta_q$ by $s_{b_{q'}} = \frac{s_{b_q} + s_{a_p}}{2}$
     end if
     if syntactic relation = neg then
         set the emotion vector of $\beta_q$ to zero
     end if
end for
initialize $\sigma_s$ to a zero vector of length $m$
let $n$ be 0
for each word $w_i$ in $w$
    n = n+1
    $\sigma_s = \sigma_s + \sigma_{w_i}$
end for
$\sigma_s = \frac{\sigma_s}{n}$
maxScore = the first element of $\sigma_s$
maxEmotion = $e_1$
for i=2 to m
    if the $i^{th}$ element of $\sigma_s$ > maxScore
       maxScore = the $i^{th}$ element of $\sigma_s$
       maxEmotion = $e_i$
    end if
end for
if maxScore ≥ t then
    $\omega_s$ = maxEmotion
else
    $\omega_s$ = "neutral"
end if
return $\omega_s$
```

It is worth mentioning that since our approach is not dependent on any affect dictionaries, it is easy to expand, shrink or change the emotion categories we work with.

Most of the times, Ekman’s six basic emotions – *happiness, sadness, anger, fear, surprise* and *disgust* suffice. However, due to the ambiguous nature of emotions, it is sometimes hard to label a sentence strictly with one of the six emotions and there arises a need for a new category such as *shame, guilt, anticipation* or *love*. This can be easily incorporated into our algorithm by calculating semantic relatedness scores between the affect NAVA words and the new emotion concepts. We demonstrate the usefulness of this functional flexibility when evaluating our algorithm on the ISEAR data set in section 4.3.

Chapter 4
Evaluation and Results

In order to evaluate the performance of our system and to compare it with other related work, several sets of experiments were conducted. In this chapter, we report and discuss the results of the evaluation of our algorithm.

Different variations of the algorithm and model building were put to test to answer the following questions:

1. How much effect does the text corpus have on the semantic relatedness scores, and ultimately on the accuracy of emotion detection? We aim to determine the extent of effect of the corpus used to build a model for retrieving semantic relatedness scores by comparing three different corpora:
   a. Wikipedia, as used in (Recchia & Jones, 2009), is the starting point of verification of the claim whether more data help in better semantic analysis.
   b. In the spirit of continuing exploring large data sets, we also use the Gutenberg[11] corpus, which intuitively contains "more" emotional text as it is made up millions of stories, as opposed to Wikipedia data which are more objective and factual in nature.
   c. After using two large volume data sets, we ask ourselves – what if we combined the two abovementioned large data sets to create an even bigger

[11] http://www.gutenberg.org/

data set? How would that affect the semantic relatedness scores? To test this idea, we concatenate the Wikipedia and Gutenberg corpora into one to get an even larger Wiki-Guten corpus.

2. The second question we would like answered is whether stemming will improve or worsen the accuracy? On one hand, it feels like stemming would generalize the words too much e.g. words “love”, “loved”, “loving” would all be stemmed to “love”. On the other hand, since all these three words share one stem, most likely they represent a similar emotion concept and it may be advantageous to exploit this latent relationship. We test this hypothesis by using two different versions – stemmed and *unstemmed* (original) – of the corpora.
3. Thirdly and more importantly, can the underlying syntactic dependency structure of a sentence provide some context and therefore help in improving the average accuracy? We answer this question by comparing the results of two versions of the algorithm – the context-based version where the syntactic dependencies are used and the context-free version where the use of such dependencies is excluded.
4. Finally, how does our proposed context-sensitive emotion detection method based on semantic relatedness scores perform compared to other unsupervised and supervised methods? To answer this question, we compare the results of our proposed method with other recently proposed methods reported in related work literature.

The results are stated using widely used metrics such as precision, recall and F-measure, which can be viewed as extended versions of the simple metric “accuracy”.

Accuracy refers to the percentage of correct predictions made by the model when compared with the actual classifications in the test data.

In the context of classification task such as emotion classification, the terms true positives, true negatives, false positives and false negatives are used to compare the predicted classification of a sentence (the emotion label assigned to the sentence by the classifier) with the actual correct classification (the annotated label of the sentence). Consider an emotion category (such as sadness) as a positive class. True positives are the sentences correctly labelled as belonging to the positive class i.e. the predicted label matches the annotated label. False positives are the items incorrectly labelled as belonging to the positive class. False negatives are the items which were not labelled as belonging to the positive class but should have been. We use *tp* to denote the number of sentences labelled correctly as belonging to the positive class, *fp* to denote the number of sentences that were classified as the positive class but in fact belong to the negative class, *tn* to denote the number of sentences that were labelled correctly as belonging to the negative class, and *fn* to denote the number of sentences that were classified as the negative class but in fact belong to the positive class. This is illustrated in Table 4.1 below.

Table 4.1 Coincidence matrix for a two class classifier

		predicted class	
		class positive	**class negative**
actual class	**class positive**	true positive	false negative
	class negative	false positive	true negative

Precision is defined as:

$$\pi = \frac{tp}{tp + fp} \tag{4.1}$$

Recall is defined as:

$$\rho = \frac{tp}{tp + fn} \tag{4.2}$$

F-measure or F-score, which is the harmonic mean of precision π and recall ρ, is defined as:

$$\mathrm{F} - \mathrm{measure} = \frac{2\pi\rho}{\pi + \rho} \tag{4.3}$$

Precision measures the number of correctly identified sentences as a percentage of the number of sentences identified i.e. how many of the sentences that the system labelled were actually correct. Recall measures the number of correctly identified sentences as a percentage of the total number of correct items i.e. how many of the sentences that should have been identified actually were identified.

Precision is seen as a measure of exactness whereas recall is a measure of completeness. In other words, the higher the precision, the better the system is at ensuring that what is identified is correct. A high recall rate means that the system is good at not missing correct items.

Since emotion detection is a multi-class classification task (i.e., there are multiple emotion categories, such as happiness, sadness, surprise, etc.), we calculate precision, recall and F-score on each class (i.e., considering each class as the positive class in turn) and when necessary make an average over all classes to see the overall performance on a measure.

The performance of a sentence classifier can also be measured using *accuracy*, defined as the percentage of the sentences that are correctly classified, as shown in Equation 4.4:

$$\text{Accuracy} = \frac{\text{number of correctly classified sentences}}{\text{total number of sentences}} \tag{4.4}$$

In this work, the task of sentence-level emotion recognition motivated a number of assumptions and methodological decisions as outlined below.

- Even though the perception of emotion is usually subjective, the data sets that we work with are gold standard benchmarks i.e. two or more annotators have agreed on the same emotion label for a sentence.
- Since we do not use any handcrafted lexical resource, we train a model of semantic space using a measure of semantic relatedness to compute semantic similarity between a word and an emotion concept.
- To differentiate this approach from simple machine learning algorithms, we incorporate contextual information by taking into account the syntactic dependencies within a sentence structure to see how the emotion vectors of certain words affect others.

The evaluation is performed on three commonly used data sets: Alm fairy tale data set, ISEAR data set and Aman blog data set. We talk about these data sets and present the results of our algorithm as well as the most recent reported results from related work in the following sections. A brief discussion ensues.

4.1 Evaluating the effect of stemming

Stemming is the process of reducing words to their stem, base or root form. Usually, related words map to the same stem, e.g., a stemming algorithm would reduce the words "fishing", "fished", "fish", and "fisher" to the root word "fish".

To test the effect of stemming on emotion detection, we ran our emotion detection algorithm on a subset of Alm’s fairy tale data set using a corpus of stemmed or *unstemmed* Wikipedia and Gutenberg, respectively. Table 4.2 presents the comparison between the stemmed and *unstemmed* emotion classification.

Table 4.2 Effect of stemming on overall accuracy

Alm data set (subset)	**Unstemmed (original) accuracy %**		**Stemmed accuracy %**	
	Gutenberg	Wikipedia	Gutenberg	Wikipedia
	54.76	52.20	55.34	58.08

As we can see, stemming the corpora considerably improved the accuracy of emotion detection. This is because, as stated earlier, stems/roots of words closely relating to a particular emotion concept are grouped together. This latent clustering enables computing comparatively more accurate frequency counts. Since stemming is beneficial to the overall emotion detection process, henceforth, stemmed versions of all corpora have been used.

4.2 Evaluation on the Alm fairy tale dataset

Emotions are particularly significant elements in the literary genre of fairy tales. In this experiment, we compare the performance of our approach with Alm's (E. C. O. Alm, 2008) system on sentences from children's fairy tales. To directly compare the results with recent related work, we replicate two different test sets for evaluation – one set is identical to that used by (Kim et al., 2010) which contains only four classes of emotions from Alm's data set; the other set is identical to that used by Alm (E. C. O. Alm, 2008) which contains five classes of emotions as well as the neutral category.

Some sample sentences with annotated emotion labels from this data set appear in Table 4.3.

Table 4.3 Sample sentences from Alm fairy tale data set

Sentence	Emotion
She sat down again, and stared mournfully at the grate.	Sadness
She seemed to be in a terrible fright.	Fear
The rabbits could not bear him; they could smell him half a mile off.	Anger-Disgust
He was sitting all over a small rocking chair, twiddling his thumbs and smiling, with his feet on the fender.	Happiness
At last the rope gave way with such a sudden jerk that it nearly pulled his teeth out, and quite knocked him over backwards.	Surprise

As well as comparing our results with other related work (supervised and unsupervised), we would also like to determine if our context-based approach is better than our context-free approach. So we test the two variations of our algorithms on each data set. Furthermore, to investigate the effect of different text corpus on emotion detection algorithms, we use the context-based and context-free approaches on three different text corpora, namely Wikipedia, Gutenberg and Wiki-Guten.

4.2.1 Alm six emotions classification

4.2.1.1 Data set description

Following the same evaluation scenario as Alm (E. C. O. Alm, 2008), we ran the experiment on the subset of 1,448 sentences, which includes sentences marked with five emotions as well as sentences with the neutral label. The five emotion classes are: *angry-disgusted, fearful, happy, sad* and *surprised*. These sentences have been taken from 176 stories by three authors (B. Potter, H.C. Andersen, and Grimm's), marked by high agreement (indicating that affect labels assigned by four human annotators for the sentence were identical).

As we did not have the subsets of neutral sentences used by Alm in her experiments, we randomly extracted them from the whole corpus of sentences that was labelled by human annotators as neutral (differences in data sets, however, might add some incomparability to the results). The number of neutral sentences was determined based on the number of affective labels at each level by Equation 4.5 (taken from (E. C. O. Alm, 2008)),

$$\left\lfloor \frac{|HA|}{|A^{i}|-1} \right\rfloor \qquad (4.5)$$

such that *HA* is the set of high-agreement affect sentences in the whole corpus; A^i is the set of affect labels at a specific level *i* in the affect hierarchy. In Alm's thesis, an affect hierarchy has been described as shown in Table 4.4. Since we perform a fine-grained classification, we consider labels at the 'all' level. The sentence distribution including the number of neutral sentences at this level is shown in Table 4.5.

Table 4.4 Affect hierarchy partly taken from Alm's thesis

Level name (*i*)	Label names (A^i)
all	angry-disgusted, fearful, happy, sad, surprised, neutral
mid	negative, positive, neutral
top	emotional, neutral

Table 4.5 Distribution of sentences in Alm data set

Emotion label	Number of sentences
Happiness	445
Sadness	264
Anger-Disgust	218
Fear	166
Surprise	114
Neutral[12]	241

[12] Out of 5380 neutral sentences, 241 sentences were randomly picked. This was repeated 2 times (as done in Alm 2008) and the average accuracy is reported.

4.2.1.2 Results and discussion

In this section, we report the results on all six emotion categories – *happiness, sadness, anger-disgust, fear, surprise* and *neutral*, along with results reported in Alm (E. C. O. Alm, 2008). Table 4.6 containing results partially taken from Alm shows a majority class baseline which is the ratio of the most frequent affect label i.e. happiness. We also implemented a simple keyword-based algorithm as a baseline which used WordNet-Affect as the lexical resource.

Alm’s unsupervised lextag method used special word lists for specific emotions. It employed a straightforward heuristic based on these word lists to assign affect labels to sentences. By counting the number of word or alternatively stem hits for extracted content-word-like words in a sentence, it output the label or tied labels with the maximum number of hits for a given sentence. A default neutral label was output in the case of no hits. In the case of overlap across word lists, a word/stem hit could count towards more than one emotion label. This method allowed multiple winners, therefore outputting any tied winning labels as prediction.

We present the results from the Wikipedia corpus, implementing the context-based and context-free approaches.

Table 4.6 Results on Alm fairy tale data set

Algorithm	Happiness	Sadness	Anger-Disgust	Fear	Surprise	Neutral[13]
	Accuracy					
Majority class baseline	31 % (Happiness)					
Keyword baseline	45 %					
Alm's unsupervised lextag method	54-55 %					
Without context (Wikipedia)	56.31 %					
With context (Wikipedia)	57.25 %					

As it can be noticed from Table 4.6, our unsupervised approaches (with and without context) perform better than Alm's unsupervised method and almost 12% more accurate than the keyword-based baseline. Additionally, context-based approach with accuracy 57.25% is slightly better than the context-free approach which gives an accuracy of 56.31%.

[13] The value of threshold *t* was empirically set to 0.015 for this data set.

4.2.2 Alm four emotions classification

4.2.2.1 Data set description

In the interest of maintaining similarity with the test set used by (Kim et al., 2010), we evaluate our algorithm on the exact sentences as reported by them. To be able to compare our results with other unsupervised approaches, we only work with four emotion categories - *anger-disgust, happiness, fear* and *sadness*. The sentence distribution used for this task is shown in Table 4.7.

Table 4.7 Sentence distribution from the Alm data set used in this task

Emotion label	Number of sentences
Happiness	445
Sadness	264
Anger-Disgust	218
Fear	166

4.2.2.2 Results and discussion

The results of the four class emotion categorization are reported in Table 4.8 which include results partially taken from (Kim et al., 2010) of four unsupervised approaches detailed in section 0 above. A categorical model based on a VSM with dimensionality reduction variants such as LSA, PLSA, and NMF using WordNet-Affect as the linguistic lexical dictionary, and a dimensional model (DIM) using the ANEW database of affect terms, are compared with our context-based and context-free algorithms on three corpora, without using any affect resource.

Table 4.8 Results on Alm data set four category classification

Algorithm		Measure	Happiness	Sadness	Anger-Disgust	Fear
Keyword baseline		ρ	0.773	0.667	0.940	0.867
		π	0.481	0.303	0.142	0.157
		F_1	0.593	0.417	0.247	0.265
LSA		ρ	0.847	0.704	0.386	0.710
		π	0.637	0.589	0.749	0.583
		F_1	0.727	**0.642**	0.510	**0.640**
PLSA		ρ	0.555	0.333	0.239	0.000
		π	0.358	0.414	0.455	0.000
		F_1	0.436	0.370	0.313	0.000
NMF		ρ	0.802	0.708	0.773	0.704
		π	0.761	0.821	0.560	0.784
		F_1	0.781	**0.760**	**0.650**	**0.741**
DIM		ρ	0.661	0.408	0.604	0.444
		π	0.979	0.169	0.290	0.179
		F_1	**0.789**	0.240	0.392	0.255
Without context	Wikipedia	ρ	0.758	0.641	0.627	0.465
		π	0.703	0.466	0.486	0.765
		F_1	0.730	0.539	0.548	0.579
	Gutenberg	ρ	0.733	0.722	0.836	0.390
		π	0.703	0.443	0.234	0.807
		F_1	0.718	0.549	0.366	0.525
	Wiki-Guten	ρ	0.785	0.742	0.798	0.433
		π	0.688	0.534	0.417	0.837
		F_1	0.733	0.621	0.548	0.571
With context	Wikipedia	ρ	0.756	0.629	0.644	0.466
		π	0.710	0.462	0.523	0.777
		F_1	0.732	0.533	**0.577**	0.582
	Gutenberg	ρ	0.736	0.694	0.828	0.398
		π	0.708	0.447	0.243	0.807
		F_1	0.722	0.544	0.376	0.533

	Wiki-Guten	ρ	0.786	0.760	0.786	0.431
		π	0.694	0.527	0.422	0.849
		F_1	**0.737**	0.622	0.549	0.572

ρ = Precision, π = Recall, F_1 = F-score

As we can see from Table 4.8 (top two F-scores shown in bold), for four-class classification, our results are comparable to other unsupervised approaches while Non-negative Matrix Factorization (NMF) applied to semantic analysis stands out as the best method. It is worth mentioning that although all the methods compared in this task are unsupervised approaches (i.e. do not require annotated data at any stage of processing), our approaches do not require any affect lexicons either, whereas the other approaches do.

Table 4.9 Average F-scores of unsupervised methods on the Alm data set

Algorithm		Average F-score
LSA		0.629
PLSA		0.279
NMF		0.733
DIM		0.419
Without context	Wikipedia	0.599
	Gutenberg	0.540
	Wiki-Guten	0.618
With context	Wikipedia	0.606
	Gutenberg	0.544
	Wiki-Guten	0.620

Table 4.9 presents the average F-score of each method over the four emotion categories. Looking at the overall performance of the algorithms, the proposed context-

sensitive method using the Wiki-Guten corpus performs better than PLSA and DIM, is comparable to LSA and slightly worse than NMF.

4.3 Evaluation on the ISEAR emotion dataset

The next data set we use is the ISEAR (International Survey on Emotion Antecedents and Reactions) data set which consists of 7,666 sentences. For building the ISEAR data set, 1,096 participants from different cultural backgrounds completed questionnaires about experiences and reactions for seven emotions including *anger, disgust, fear, joy, sadness, shame* and *guilt*. Some sample sentences extracted from ISEAR along with their emotion label appear in Table 4.10.

Table 4.10 Sample sentences from ISEAR data set

Sentence	Emotion
When I was robbed in a bus.	Anger
The smell of garlic in rush-hour bus.	Disgust
When a thief broke into my house at night.	Fear
I ignored and offended my parents on the eve of the New Year.	Guilt
I felt joy when my two twin sisters were born.	Joy
My grandfather's funeral.	Sadness
I am ashamed of the horrible way I used to treat my sister.	Shame

4.3.1 ISEAR four emotions classification

4.3.1.1 Data set description

In the interest of maintaining similarity with test set used by (Kim et al., 2010), we evaluate our algorithm on the exact sentences as reported by them. To be able to compare our results with other unsupervised approaches, we only work with four emotion categories - *anger-disgust, happiness, fear* and *sadness.* The sentence distribution[14] is kept similar to that of (Kim et al., 2010) as shown in Table 4.11.

Table 4.11 Sentence distribution from ISEAR for four emotions

Emotion label	Number of sentences
Happiness	1090
Sadness	1082
Anger-Disgust	2168
Fear	1090

4.3.1.2 Results and discussion

To begin with, we compare our results with that of unsupervised approaches evaluated by (Kim et al., 2010) as shown in Table 4.12. It should be noted that since (Kim et al., 2010) employs the use of linguistic affect lexicon WordNet-Affect, their evaluation is restricted to, at most, Ekman's six emotion classification – *joy, sadness, anger, fear, disgust* and *surprise*. However, the ISEAR data set has been annotated with seven emotions – *joy, sadness, anger, fear, disgust, shame* and *guilt*, where emotions *shame* and *guilt* are not found in WordNet-Affect. Hence, techniques that depend on WordNet-Affect are restricted to classifying sentences only into one of Ekman's six emotions and therefore

[14] The original complete ISEAR data set contains 1094 happy sentences, 1096 sad sentences, 2192 anger-disgust sentences and 1095 fear sentences. However, to maintain similar sentence distribution to that of (Kim, Valitutti, & Calvo, 2010), we reduced the number of happy sentences by 4, sad sentences by 14, anger-disgust by 24 and fear by 5. Example of sentences removed include those that are listed as [No description], [No response], [I have not felt this emotion yet] and so on. This may result in slight discrepancy, nevertheless, as we do not have access to the exact sentences used in their evaluation.

unable to perform a full seven-category classification on the ISEAR data set. Since our approach is independent of any affect lexical resource, we report results of our evaluation on all the seven categories of ISEAR, in the following subsection. To the best of our knowledge, no other emotion detection approach has reported results on all seven categories.

Table 4.12 Results of four category classification on ISEAR data set

Algorithm		Measure	Joy	Sadness	Anger-Disgust	Fear
Majority Class baseline		Precision	-	-	0.399	-
		Recall	-	-	1.000	-
		F-score	-	-	0.571	-
Keyword baseline		Precision	0.308	0.237	0.489	0.319
		Recall	0.468	0.314	0.268	0.339
		F-score	0.371	0.270	0.346	0.328
LSA		Precision	0.333	0.500	0.468	0.633
		Recall	0.061	0.059	0.970	0.038
		F-score	0.103	0.106	0.631	0.071
PLSA		Precision	0.307	0.198	0.536	0.000
		Recall	0.381	0.491	0.397	0.000
		F-score	0.340	0.282	0.456	0.000
NMF		Precision	0.385	0.360	0.410	0.689
		Recall	0.005	0.009	0.987	0.029
		F-score	0.010	0.017	0.579	0.056
DIM		Precision	0.349	0.522	0.708	0.531
		Recall	0.980	0.249	0.179	0.263
		F-score	0.515	0.337	0.286	0.351
With context	Wikipedia	Precision	0.590	0.766	0.774	0.513
		Recall	0.540	0.278	0.529	0.698
		F-score	0.564	**0.408**	0.628	**0.592**

	Gutenberg	Precision	0.534	0.750	0.753	0.414
		Recall	0.620	0.152	0.475	0.762
		F-score	**0.574**	0.253	0.582	0.536
	Wiki-Guten	Precision	0.596	0.910	0.715	0.480
		Recall	0.497	0.177	0.627	0.714
		F-score	0.542	0.296	**0.668**	0.574

From the Table 4.12 above, we notice that our context-based approach has the highest F-score values for all the four emotion categories. The best average F-score value is the result of context-based method using Wikipedia corpus to derive the semantic relatedness scores. Notice the trend for *sadness* category. Even though it has the highest precision amongst all the other categories, its F-score value is amongst the lowest due to its lowest recall value. This means that though the sentences that are labelled as *sad* have a very high probability of being *sad*, very few *sad* sentences are actually being retrieved. This is the problem of low coverage where our system is unable to recognize the *sad* affect bearing words as indeed being *sad*. Table 4.13 shows the average F-score over all the four emotion categories. The unsupervised context-based approach for all three corpora results in significantly higher average F-score values than the four other unsupervised approaches.

Table 4.13 Average F-scores of unsupervised methods on the ISEAR data set

Algorithm		Average F-score
LSA		0.227
PLSA		0.269
NMF		0.165
DIM		0.372
With context	Wikipedia	**0.548**
	Gutenberg	0.486
	Wiki-Guten	0.520

4.3.2 ISEAR seven emotions classification

4.3.2.1 Data set description

Here, we extend the previous task to a full seven emotion category classification. The distribution of emotion categories used is shown in Table 4.14. This is the original complete ISEAR data set containing all seven classes.

Table 4.14 Distribution of sentences in ISEAR data set

Emotion label	Number of sentences
Joy	1094
Sadness	1096
Anger	1096
Fear	1095
Disgust	1096
Shame	1096
Guilt	1093

4.3.2.2 Results and discussion

As mentioned earlier, the ISEAR data set contains *shame* and *guilt* emotions which do not feature in affect lexicons such as WordNet-Affect. Hence, methods that depend on affect lexicons (and most of them do) are unable to categorize the ISEAR data set in full. Since we compute affect affinity using statistical machine learning from semantic relatedness scores, it gives us the flexibility to work with any emotion category. We present the results of full seven class categorization[15] in Table 4.15.

Table 4.15 Results on ISEAR data set of seven emotions classification

Algorithm (with context)	Measure	Joy	Sadness	Anger	Fear	Disgust	Shame	Guilt
Wikipedia	Precision	0.492	0.706	0.347	0.410	0.828	0.378	0.364
	Recall	0.538	0.276	0.510	0.698	0.290	0.423	0.316
	F-score	**0.514**	**0.396**	0.413	**0.517**	0.430	**0.400**	0.338
Gutenberg	Precision	0.428	0.702	0.403	0.309	0.634	0.575	0.416
	Recall	0.599	0.151	0.429	0.759	0.328	0.303	0.340
	F-score	0.500	0.248	**0.415**	0.439	0.432	0.397	**0.374**
Wiki-Guten	Precision	0.504	0.872	0.323	0.366	0.656	0.542	0.338
	Recall	0.496	0.174	0.575	0.711	0.367	0.316	0.321
	F-score	0.500	0.290	0.414	0.483	**0.470**	0.400	0.329

As it can be noticed from Table 4.15, for four out of seven categories, the context-based approach based on the Wikipedia corpus results in the highest F-score values,

[15] Note that the ISEAR data set does not contain any neutral sentences. Hence the value of threshold *t* was set to null i.e. each sentence was labelled with one of the seven emotions and no sentences were classified as neutral.

followed by the Gutenberg corpus and lastly the Wiki-Guten corpus. *Fear* and *joy* report the highest F-score values, whereas the precision, recall and F-score values of *guilt* are among the lowest. This is because looking at the annotated data set, we found sentences such as “While having an argument with my daughter, I got angry and over-excited and said angry words.”, “When I beat up my son for having beaten up his sister.”, and “Falling in love with a close friend.” labelled as *guilt*. These sentences border on the fuzzy boundary of emotions and we believe that deeper syntactic and semantic analyses are required to decipher the underlying feelings. One approach could be to combine the syntactic relationships with *subject+verb+object* phrasal analysis to detect the subject f the sentence and then if the following phrase’s emotional vector indicates something negative *caused* by the subject, it could be considered as *guilt*, although the challenge remains in keeping such rules simple and to a bare minimum.

Table 4.16, which provides the average F-scores, shows that the context-based approach using the Wikipedia corpus yields the best results.

Table 4.16 Average F-scores across three corpora for ISEAR data set

Algorithm (with context)	**Average F-score**
Wikipedia	0.430
Gutenberg	0.401
Wiki-Guten	0.412

4.4 Evaluation on the Aman blog dataset

This data set was developed and kindly provided by Aman and Szpakowicz (S. Aman & Szpakowicz, 2007). It includes sentences collected from blogs, which are characterized by rich emotional content and good examples of real-world instances of emotions conveyed through text.

4.4.1 Dataset description

The complete data set has 5290 sentences but to directly compare our approach with other approaches, we consider their benchmark as the gold standard, which includes 1890 sentences annotated by one of Ekman's six emotions (*happiness, sadness, anger, disgust, fear* and *surprise*) as well as neutral, on which two annotators completely agreed. Some sample sentences and their annotated emotion labels appear in Table 4.17.

Table 4.17 Examples of annotated sentences from Aman and Szpakowicz blog data set

Sentence	Emotion	Intensity
I have to look at life in her perspective, and it would break anyone's heart.	Sadness	High
We stayed in a tiny mountain village called Droushia, and these people brought hospitality to incredible new heights.	Surprise	Medium
But the rest of it came across as a really angry, drunken rant.	Anger	High
And I realllllly want to go to Germany – dang terrorists are making flying overseas all scary and annoying and expensive though!!	Mixed emotion	High
I hate it when certain people always seem to be better at me in everything they do.	Disgust	Low
Which, to be honest, was making Brad slightly nervous.	Fear	Low

The distribution of labels across sentences from the gold standard used in this experiment is shown in Table 4.18.

Table 4.18 Distribution of sentences in Aman data set

Emotion label	Number of sentences
Happiness	536
Sadness	173
Anger	179
Fear	115
Surprise	115
Disgust	172
Neutral	600[16]

4.4.2 Results and discussion

In this section, we compare the results of different versions of our algorithm with recent related work. Since there is no unsupervised, affect lexicon independent work reporting results on this data set as far as we know, we present results of supervised approaches taken from (S. Aman & Szpakowicz, 2008) and (Ghazi et al., 2010).

Supervised Vector Machine (SVM) with tenfold cross validation was applied by (S. Aman & Szpakowicz, 2008) and results reported from two different algorithms – one using corpus-based unigrams (referred to as Aman machine learning with unigram in Table 4.19) and the other better result obtained by combining corpus-based unigrams,

[16] The original gold benchmark contains 2800 neutral sentences. However, to keep the experimental setup relatively same as the other reported results, we choose 600 neutral sentences randomly. To get a more accurate result, we run the experiment 4 times.

features derived from emotional lists of words from Roget's Thesaurus and common words between the dataset and WordNetAffect (referred to as Aman machine learning with unigrams, RT and WNA features in Table 4.19).

SVM with tenfold cross validation was also applied by (Ghazi et al., 2010). They used a combination of affect lexicons such as emotional lists of words from Roget's Thesaurus and WordNet-Affect (together referred to as the polarity feature set).

To the best of our knowledge, there are no results reported on the Aman data set using an unsupervised approach. So, we present the outcome of our unsupervised approach along with a keyword baseline and supervised approaches of Aman and Ghazi in Table 4.19. It should be noted that unlike other approaches, our technique is not dependent on any manually created affect lexicon such as Roget's Thesaurus or WordNet-Affect. However, for this particular blog data set, some domain specific keywords such as 'lol', 'haha', 'hehe' and few others which are not found in conventional textual data but are commonly used in blogosphere were manually added to the score table since it was not possible to derive semantic relatedness word scores using word frequency counts from text corpora Wikipedia, Gutenberg and Wiki-Guten.

Table 4.19 Aman data set: Results of our unsupervised approach as well as supervised approaches

Algorithm		Measure	Happiness	Sadness	Anger	Fear	Surprise	Disgust	Neutral[17]
Keyword baseline		ρ	0.717	0.455	0.495	0.408	0.271	0.667	0.375
		π	0.407	0.260	0.268	0.174	0.182	0.081	0.82
		F_1	0.519	0.331	0.348	0.244	0.218	0.145	0.51
Aman's ML with unigrams		ρ	0.840	0.619	0.634	0.889	0.813	0.772	0.581
		π	0.675	0.301	0.358	0.487	0.339	0.453	0.342
		F_1	0.740	0.405	0.457	0.629	0.479	**0.571**	0.431
Aman's ML with unigrams, RT and WNA features		ρ	0.813	0.605	0.650	0.868	0.723	0.672	0.587
		π	0.698	0.416	0.436	0.513	0.409	0.488	0.625
		F_1	**0.751**	0.493	**0.522**	0.645	**0.522**	0.566	0.605
Inkpen's Hierarchical		F_1	0.69	0.46	0.43	0.45	0.38	0.31	**0.84**
Without context	Gutenberg	ρ	0.812	0.713	0.533	0.483	0.576	0.716	0.587
		π	0.682	0.387	0.358	0.617	0.330	0.308	0.922
		F_1	0.742	0.502	0.428	0.542	0.420	0.431	0.717
	Wikipedia	ρ	0.774	0.556	0.385	0.384	0.537	0.627	0.680
		π	0.724	0.433	0.587	0.704	0.313	0.302	0.703
		F_1	0.748	0.487	0.465	0.497	0.396	0.408	0.691
	Wiki-Guten	ρ	0.854	0.738	0.563	0.845	0.690	0.701	0.600
		π	0.590	0.277	0.352	0.522	0.174	0.314	0.496
		F_1	0.698	0.403	0.433	0.645	0.278	0.434	0.453
With context	Gutenberg	ρ	0.817	0.739	0.537	0.476	0.594	0.727	0.592
		π	0.685	0.393	0.363	0.600	0.330	0.326	0.927
		F_1	0.745	**0.513**	0.433	0.531	0.425	0.450	0.722
	Wikipedia	ρ	0.777	0.566	0.380	0.391	0.594	0.635	0.680
		π	0.728	0.445	0.564	0.704	0.330	0.314	0.715
		F_1	**0.751**	0.498	0.454	0.503	0.425	0.420	0.697
	Wiki-Guten	ρ	0.899	0.820	0.571	0.836	0.800	0.718	0.494
		π	0.582	0.289	0.358	0.530	0.174	0.326	0.960
		F_1	0.707	0.427	0.440	**0.649**	0.286	0.448	0.652

[17] The value of threshold *t* was empirically set to 0.02 for this data set.

Useful as it is to see the individual category-wise breakdown of precision, recall and F-score of each method, it is also functional to average the F-scores of each method and look at them from a global perspective to compare their effectiveness. The above mentioned F-scores when averaged are shown in Table 4.20.

Table 4.20 Average F-scores on the Aman data set

Algorithm		Average F-score
Keyword baseline		0.331
Aman's supervised ML with unigrams		0.530
Aman's supervised ML with unigrams, Roget's Thesaurus and WordNet-Affect features		**0.586**
Inkpen's hierarchical approach		0.508
Without context	Wikipedia	0.527
	Gutenberg	0.540
	Wiki-Guten	0.478
With context	Wikipedia	0.535
	Gutenberg	**0.546**
	Wiki-Guten	0.516

As it can be seen from Table 4.19, for three (*happiness*, *sadness* and *fear*) out of seven emotion categories, the unsupervised context-based approach's results are better than or as good as those from supervised affect lexicon-dependent approaches. When comparing the overall performance of a method, the context-based approach using Gutenberg corpus is the best technique in unsupervised category, whereas Aman's machine learning with unigrams, RT and WNA features approach has the highest average under the supervised techniques.

4.5 Summary of Results

4.5.1 Evaluating the effect of different text corpora

In sections 4.2 to 4.4 we presented results of the context-sensitive version of our algorithm using semantic relatedness scores calculated employing Pointwise Mutual Information from three different text corpora. Now, let us see how these different corpora fared when compared to each other in Table 4.21.

Table 4.21 Effect of different text corpora on our unsupervised approach

Algorithm (with context)		Average F-score
Aman	Wikipedia	0.535
	Gutenberg	0.546
	Wiki-Guten	0.516
Alm (four emotions)	Wikipedia	0.606
	Gutenberg	0.544
	Wiki-Guten	0.620
ISEAR (four emotions)	Wikipedia	0.548
	Gutenberg	0.486
	Wiki-Guten	0.520
ISEAR (seven emotions)	Wikipedia	0.430
	Gutenberg	0.401
	Wiki-Guten	0.412

Figure 4.1 shows that the Wikipedia corpus resulted in the highest average F-score values for two out of four tasks and took second place in the other two tasks. This

could be attributed to the fact that Wikipedia contains more structured data with pages dedicated wholly to emotion concepts which includes several of their synonyms in close proximity. On the other hand, Gutenberg corpus falls in last place in three out of four tasks while being the best result in one task – the Aman data set, although it is only slightly better than the result of Wikipedia.

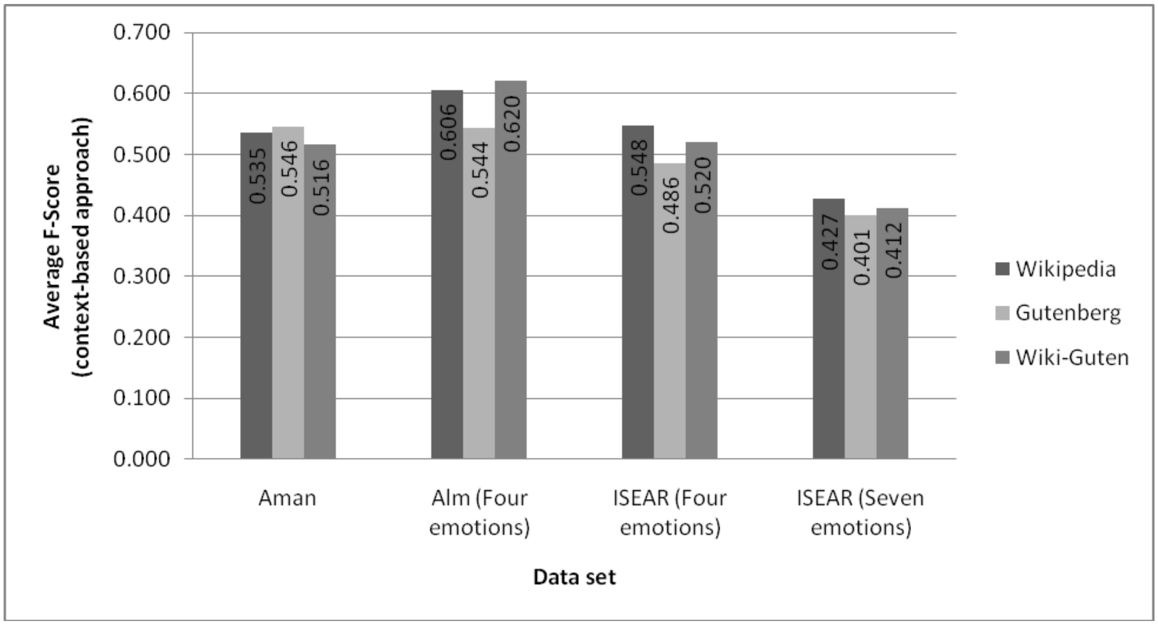

Figure 4.1 Average F-score values across three corpora for three different data sets

From the aforementioned results, we can conclude that context-based approach using semantic relatedness scores derived from the Wikipedia corpus performed relatively better than the models built using other text corpora.

4.5.2 Our proposed method compared with other unsupervised methods

As it can be noticed from the previous sections, different methods perform differently on different data sets. For example, NMF outperforms every other unsupervised method

To compare our proposed unsupervised method with other unsupervised approaches, let us look at Table 4.22 (which is derived from Table 4.9 and Table 4.13 that present the average F-score values of four emotions classification using various unsupervised approaches on the Alm and the ISEAR data set respectively).

Table 4.22 Comparison of unsupervised approaches

Method		Alm	ISEAR	Average
LSA		0.629	0.227	0.428
PLSA		0.279	0.269	0.274
NMF		0.733	0.165	0.449
DIM		0.419	0.372	0.396
With context	Wiki	0.606	0.548	**0.577**
	Guten	0.544	0.486	**0.503**
	Wiki-Guten	0.620	0.520	**0.570**

One of the first observations that can be made is that the F-score values of the Alm data set are higher than the other data set. This could be because the sentences in the fairy tale data set tend to have more emotional words. Secondly, it is noticed that although the NMF outperforms every other method on the Alm data set, it has the weakest results on the ISEAR data set. The reason for such discrepancy implies that this technique is effective for certain types of sentences only.

To see which of these unsupervised approaches can be relied upon to give a more consistent outcome, we average the values to get a global picture. Doing so enables us to infer that our proposed context-sensitive method is more suitable in a general emotion classification task.

Chapter 5
Conclusion

Emotion detection from text is an interesting problem with wide-ranging applications in market analysis, eLearning and recently social media analysis. There are several supervised and unsupervised approaches for recognizing various emotions from textual data. However, supervised approaches require large training data to build classification models. On the other hand, typical unsupervised approaches treat text as a bag of words and miss meaningful relationships between words which are important indicators conveying emotions. There are also rule-based approaches which look at the underlying language structure in detail but developing and maintaining these rules is not a trivial task.

In this thesis we proposed a novel context-based unsupervised method for detecting emotion from textual data. Our approach requires neither an annotated data set nor any affect lexicon. The results of evaluations on several data sets show that our technique yields more accurate results than other recent unsupervised approaches and comparable results to those of some supervised approaches.

5.1 Research Contributions

A summary of the contributions of this thesis is as follows:

- A novel unsupervised approach for detecting emotions at the sentence level was proposed. Unlike supervised methods, our unsupervised technique does not require any annotated data sets for training a model before classification.
- Widely used handcrafted affect lexicons such as WordNet-Affect and Roget's Thesaurus have pre-defined emotion categories. This makes them unpractical to be used in cases where different emotion category classification is needed. We do not employ any such lexical resources, thus allowing our technique to be more generic which can be adopted to fit into any emotional model. Since our approach is independent of any affect dictionary, it can classify sentences from data sets such as ISEAR which have *shame* and *guilt* categories which, to the best of our knowledge, are not found in any state-of-the-art affect dictionaries.
- To avoid using affect lexical resources, we adopt a statistical machine learning approach where we compute the emotional affinity of affect bearing words by computing semantic relatedness scores between such words and emotion concepts. These scores are computed by analyzing word co-occurrences within a publicly available text corpus such as Wikipedia or Gutenberg.
- To enrich such a statistical approach, we incorporate syntactic information using our context-based algorithm which uses two syntactic dependencies to integrate some context information and adjust word emotion vectors accordingly. Doing so helps reduce errors that result from the use of pure statistical machine learning approaches.

5.2 Results

Extensive evaluation of our classification framework on various data sets shows promising results when compared to other unsupervised techniques. Using an emotion vector of a sentence is helpful when dealing with multiple-emotion categorization, which our approach is capable of. One of the weaknesses of our approach is that the semantic relatedness scores depend on the text corpus from which they are derived. From the various evaluations carried out on different data sets, we found that the Wikipedia corpus performed better than the Gutenberg or the Wiki-Guten data set.

To test the effect of stemming on the emotion detection process, stemmed and *unstemmed* versions of Wikipedia and Gutenberg were used to perform emotion classification on one of the data sets and the stemmed versions resulted in almost 6% increase in average accuracy.

On the Alm data set, the results of our context-based unsupervised approach were comparable to other unsupervised approaches when classifying four emotions. For six emotions classification, context-free and context-based techniques yielded about 1-3% improvement in accuracy compared to another unsupervised technique and about 12% better than the simple keyword baseline. Four-class categorization on the ISEAR data set showed that the context-based approach significantly outperformed all the four other unsupervised approaches. Moreover, we report results on the full seven-class categorization but unfortunately, to the best of our knowledge, we have no results to compare them with. For three out of seven emotion categories in the Aman data set, our context-based unsupervised approach yielded better F-scores than the three recent supervised techniques.

From empirical observations, it can be concluded that the context-based approach consistently outperformed the context-free approach in all the data sets evaluated which supports the claim that it is better to look at words within their context rather than applying pure statistical machine learning techniques.

5.3 Future Work

In the future, we would like to explore the possibility that average semantic relatedness scores from multiple measures (e.g. Normalized Search Similarity or Spreading Activation) may be more accurate than semantic scores from individual measure alone. It would also be useful to set a theoretical formula for the threshold of declaring “neutral” sentences. Future developments may also include considering the use of other syntactic dependencies to see if it can result in more accurate results. For example, intensifiers are generally expressed using the adverbial modifier syntactic dependency and it will be exciting to experiment with different relationships to test their effect on the overall emotion detection process. Another direction to look into would be to consider extending the *context* to include the previous or next sentence(s) of a paragraph.

Bibliography

Alm, C. O., Roth, D., & Sproat, R. (2005). Emotions from text: Machine learning for text-based emotion prediction. *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada. 579-586. doi:http://dx.doi.org/10.3115/1220575.1220648

Alm, E. C. O. (2008). *AFFECT IN TEXT AND SPEECH.*

Aman, S., & Szpakowicz, S. (2008). Using roget's thesaurus for fine-grained emotion recognition. *Proceedings of the Third International Joint Conference on Natural Language Processing,* 296-302.

Aman, S., & Szpakowicz, S. (2007). Identifying expressions of emotion in text.*4629*, 196-205.

Argamon, S., Bloom, K., Esuli, A., & Sebastiani, F. (2007). Automatically determining attitude type and force for sentiment analysis. *In Proceedings of the 3rd Language and Technology Conference (LTC'07).*

Baeza-Yates, R. A., & Ribeiro-Neto, B. (1999). *Modern information retrieval*. Boston, MA, USA: Addison-Wesley Longman Publishing Co., Inc.

Bradley, M. M., Lang, P. J., & Cuthbert, B. N. (1999). *Affective norms for english words (ANEW): Stimuli, instruction manual, and affective ratings.* No. C-1). Center for Research in Psychophysiology, University of Florida, Gainesville, Florida.

Bullinaria, J. A., & Levy, J. P. (2007). Extracting semantic representations from word co-occurrence statistics: A computational study. *Behavior Research Methods,* (3), 510.

C., B. A. (2002). Real time text-to-emotion engine for expressive internet communications. *Proceedings of International Symposium on Communication Systems, Networks and Digital Signal Processing (CSNDSP-2002).*

Chaumartin, F. (2007). UPAR7: A knowledge-based system for headline sentiment tagging. *Proceedings of the 4th International Workshop on Semantic Evaluations,* Prague, Czech Republic. 422-425.

Church, K. W., & Hanks, P. (1990). Word association norms, mutual information, and lexicography. *Comput.Linguist., 16*(1), 22-29.

Dumais, S. (2003). Data-driven approaches to information access. *Cognitive Science, 27*(3), 491. doi:DOI: 10.1016/S0364-0213(03)00013-2

Ekman, P. (Apr 1993). Facial expression and emotion. *American Psychologist, 48*(4), 384-392.

Esuli, A., & Sebastiani, F. (2006). SENTIWORDNET: A publicly available lexical resource for opinion mining. *In Proceedings of the 5th Conference on Language Resources and Evaluation (LREC'06),* 417-422.

Fu, W., & Pirolli, P. (November 2007). SNIF-ACT: A cognitive model of user navigation on the world wide web. *Human-Computer Interaction, 22*(4), 355-412.

Gamallo, P., Gasperin, C., Agustini, A., & Lopes, G. (2001). Syntactic-based methods for measuring word similarity.*2166*, 116-125.

Ghazi, D., Inkpen, D., & Szpakowicz, S. (2010). Hierarchical versus flat classification of emotions in text. *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text*, Los Angeles, California. 140-146.

Golder, S. A., & Macy, M. W. (2011). Diurnal and seasonal mood vary with work, sleep, and daylength across diverse cultures. *Science*, *333*(6051), 1878-1881. doi:10.1126/science.1202775

Hatzivassiloglou, V., & McKeown, K. R. (1997). Predicting the semantic orientation of adjectives. *Proceedings of the 35th Annual Meeting of the Association for Computational Linguistics and Eighth Conference of the European Chapter of the Association for Computational Linguistics,* Madrid, Spain. 174-181. doi:http://dx.doi.org.ezproxy.library.yorku.ca/10.3115/976909.979640

He, S., Ballard, D., & Gildea, D. (2004). *Building and tagging with an affect lexicon.* No. 833). Rochester, NY: The University of Rochester, Computer Science Department.

Hofmann, T. (2001). Unsupervised learning by probabilistic latent semantic analysis. *Machine Learning, 42*(1), 177-196.

Inrak, P., & Sinthupinyo, S. (2010). Applying latent semantic analysis to classify emotions in thai text. *Computer Engineering and Technology (ICCET), 2010.* V6-450-V6-454.

Jarmasz, M., & Szpakowicz, S. (2001). The design and implementation of an electronic lexical knowledge base.*2056*, 325-334.

Kalev Leetaru. (2011). Culturomics 2.0: Forecasting large-scale human behavior using global news media tone in time and space. *First Monday; Volume 16, Number 9 - 5 September 2011.*

Kim, S. M., Valitutti, A., & Calvo, R. A. (2010). Evaluation of unsupervised emotion models to textual affect recognition. *Proceedings of the NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text,* Los Angeles, California. 62-70.

Kozareva, Z., Navarro, B., Vazquez, S., & Montoyo, A. (2007). UA-ZBSA: A headline emotion classification through web information. *Proceedings of the 4th International Workshop on Semantic Evaluations,* Prague, Czech Republic. 334-337.

Landauer, T. K., & Dumais, S. T. (Apr 1997). A solution to plato's problem: The latent semantic analysis theory of acquisition, induction, and representation of knowledge. *Psychological Review, 104*(2), 211-240. doi:10.1037/0033-295X.104.2.211

Landauer, T. K., Foltz, P. W., & Laham, D. (1998). An introduction to latent semantic analysis. *Discourse Processes, 25,* 259-284.

Lee, D. D., & Seung, H. S. (October 1999). Learning the parts of objects by non-negative matrix factorization. *Nature, 401*(6755), 788-791. doi:10.1038/44565

Lemaire, B., Denhière, G., Bellissens, C., & Jhean-Larose, S. (2006). A computational model for simulating text comprehension. *Behavior Research Methods, 38*(4), 628-637. doi:10.3758/BF03193895

Lindsey, R., Veksler, V. D., Grintsvayg, A., & Gray, W. D. (2007). Be wary of what your computer reads: The effects of corpus selection on measuring semantic relatedness. *Proceedings of ICCM 2007: Eighth International Conference on Cognitive Modeling*, 279-284.

Liu, H., Lieberman, H., & Selker, T. (2003). A model of textual affect sensing using real-world knowledge. *Proceedings of the 8th International Conference on Intelligent User Interfaces,* Miami, Florida, USA. 125-132. doi:http://doi.acm.org/10.1145/604045.604067

Lu, C., Hong, J., & Cruz-Lara, S. (2006). Emotion detection in textual information by semantic role labeling and web mining techniques. *Third Taiwanese-French Conference on Information Technology - TFIT 2006.*

Marneffe, M. -., & Manning, C. (2010). *Stanford typed dependencies manual.* Stanford University.

Martin, J. R., & White, P. R. R. (2005). *The language of evaluation : Appraisal in english / J.R. martin and P.R.R. white* Palgrave Macmillan, Basingstoke.

Martineau, J., & Finin, T. (May 2009). Delta TFIDF: An improved feature space for sentiment analysis. *Proceedings of the Third AAAI Internatonal Conference on Weblogs and Social Media,* San Jose, CA.

Masum, S. M. A., Prendinger, H., & Ishizuka, M. (2007). Emotion sensitive news agent: An approach towards user centric emotion sensing from the news. *Web Intelligence,* *IEEE / WIC / ACM International Conference*, 614-620. doi:http://doi.ieeecomputersociety.org/10.1109/WI.2007.124

Meena, A., & Prabhakar, T. V. (2007). Sentence level sentiment analysis in the presence of conjuncts using linguistic analysis. *Proceedings of the 29th European Conference on IR Research,* Rome, Italy. 573-580.

Miller, G. A., Beckwith, R., Fellbaum, C., Gross, D., & Miller, K. J. (1990). Introduction to wordnet: An on-line lexical database. *International Journal of Lexicography, 3*(4), 235-244.

Mueller, E. (2000). *ThoughtTreasure: A natural language/commonsense platform.* Retrieved from http://www.signiform.com/tt/htm/overview.html.

Neuman, Y., Kedma, G., Cohen, Y., & Nave, O. (2010). Using web-intelligence for excavating the emerging meaning of target-concepts. *Proceedings of the 2010 IEEE/WIC/ACM International Conference on Web Intelligence and Intelligent Agent Technology - Volume 01*, 22-25. doi:http://dx.doi.org/10.1109/WI-IAT.2010.38

Neviarouskaya, A., Prendinger, H., & Ishizuka, M. (2009). SentiFul: Generating a Reliable Lexicon for Sentiment Analysis. *Affective Computing and Intelligent Interaction and Workshops, 2009. ACII 2009.* 1. doi:10.1109/ACII.2009.5349575

Neviarouskaya, A., Prendinger, H., & Ishizuka, M. (2007). Textual affect sensing for sociable and expressive online communication.*4738*, 218-229.

Neviarouskaya, A., Prendinger, H., & Ishizuka, M. (2010). Recognition of affect, judgment, and appreciation in text. *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China. 806-814.

Olveres, J., Billinghurst, M., Savage, J., & Holden, A. (October 1998). Intelligent, expressive avatars. *Proceedings of the First Workshop on Embodied Conversational Characters,* 47-55.

OpenCyc. Retrieved from http://www.opencyc.org

Pang, B., Lee, L., & Vaithyanathan, S. (2002). Thumbs up?: Sentiment classification using machine learning techniques. *Proceedings of the ACL-02 Conference on Empirical Methods in Natural Language Processing - Volume 10,* 79-86. doi:http://dx.doi.org/10.3115/1118693.1118704

Picard, R. W. (1995). *Affective computing.*

Polanyi, L., & Zaenen, A. (2006). Contextual valence shifters.*20*, 1-10.

Quan, C., & Ren, F. (2010). An exploration of features for recognizing word emotion. *Proceedings of the 23rd International Conference on Computational Linguistics,* Beijing, China. 922-930.

Recchia, G., & Jones, M. (2009). More data trumps smarter algorithms: Comparing pointwise mutual information with latent semantic analysis. *Behavior Research Methods, 41*(3), 647-656.

Riloff, E., & Wiebe, J. (2003). Learning extraction patterns for subjective expressions. *Proceedings of the 2003 Conference on Empirical Methods in Natural Language Processing,* 105-112. doi:http://dx.doi.org/10.3115/1119355.1119369

Riva, G., Davide, F., eds, W. A. I., & Boucouvalas, A. (2003). *Real time text-to-emotion engine for expressive internet communications.*

Singh, P., Lin, T., Mueller, E., Lim, G., Perkins, T., & Li Zhu, W. (2002). Open mind common sense: Knowledge acquisition from the general public.*2519*, 1223-1237.

Strapparava, C., & A. Valitutti, A. (2004). WordNet-affect: An affective extension of wordnet. *In Proceedings of the 4th International Conference on Language Resources and Evaluation,* Lisbon, Portugal. 1083-1086.

Strapparava, C., & Mihalcea, R. (2008). Learning to identify emotions in text. *Proceedings of the 2008 ACM Symposium on Applied Computing,* Fortaleza, Ceara, Brazil. 1556-1560.

doi:http://doi.acm.org.ezproxy.library.yorku.ca/10.1145/1363686.1364052

Subasic, P., & Huettner, A. (2001). *Fuzzy Systems, IEEE Transactions on Affect Analysis of Text using Fuzzy Semantic Typing, 9*(4), 483. doi:10.1109/91.940962

The General-Inquirer. (2000). Retrieved from

http://www.wjh.harvard.edu/˜inquirer/spreadsheetguide.htm.

Tokuhisa, R., Inui, K., & Matsumoto, Y. (2008). Emotion classification using massive examples extracted from the web. *Proceedings of the 22nd International Conference on Computational Linguistics - Volume 1,* Manchester, United Kingdom. 881-888.

Turney, P. D. (2002). Thumbs up or thumbs down?: Semantic orientation applied to unsupervised classification of reviews. *Proceedings of the 40th Annual Meeting on Association for Computational Linguistics*, Philadelphia, Pennsylvania. 417-424. doi:http://dx.doi.org/10.3115/1073083.1073153

Turney, P., & Littman, M. (2003). Measuring praise and criticism: Inference of semantic orientation from association. *ACM Transactions on Information Systems, 21*, 315-346.

Veksler, V. D., & Gray, W. D. (2007). Mapping semantic relevancy of information displays. *CHI '07 Extended Abstracts on Human Factors in Computing Systems,* San Jose, CA, USA. 2729-2734. doi:http://doi.acm.org/10.1145/1240866.1241070

Wilson, T., Wiebe, J., & Hoffmann, P. (2005). Recognizing contextual polarity in phrase-level sentiment analysis. *Proceedings of the Conference on Human Language Technology and Empirical Methods in Natural Language Processing*, Vancouver, British Columbia, Canada. 347-354. doi:http://dx.doi.org/10.3115/1220575.1220619