A New Hybrid Approach to Sentiment Classification

Roseline Antai

A thesis submitted for the degree of

Doctor of Philosophy

School of Computer Science and Electronic Engineering

University of Essex

February 2016

Abstract

With the advancement of the World Wide Web, opinion sharing online has gained a lot of popularity. These opinions are utilized for decision making, market analysis, as well as other applications. The need to harness these opinions, and the motivation behind this need has led to the development and subsequent advancement of the field of Sentiment Analysis. Various issues have arisen from these, such as difficulty in locating these opinions in a body of text, as well as determining the sentiment/polarity of these opinions. To tackle the issue of opinion polarity determination, a number of classification approaches have been developed. These approaches have focused on opinion classification at various levels, such as document, sentence and aspect levels. Most document level approaches treat documents as a bag of words during the classification process, and hence classify them as a whole. The problem with this is that there could be a mixture of opinions directed towards various aspects, within a document. It is therefore imperative to utilize a classification approach which takes into account these constituent opinions. This is the focus of classification approaches which work at the aspect level. Another important factor in the issue of sentiment/polarity classification is the choice of the classification approach. This can be machine learning, lexical/lexicon-based, and more recently, hybrid. The machine learning approaches have the benefits of carrying out classification with high accuracies, and efficiently handling large feature sets, which makes them a favourite choice where high accuracies are desired. They however also have the drawback of difficulty in adaptability, due to the domain dependency of sentiment words. The pure lexicon-based approaches do not achieve the accuracy of the machine learning approaches, but are said to offer more explainable results and take into

consideration the information in lexicons. In this work, we present a novel hybrid approach, which incorporates information from lexicons in a machine learning classifier, and takes as features various linguistic knowledge sources. Our novel hybrid approach utilizes transitive dependencies to incorporate the opinions expressed towards different aspects of a document in determining the polarity classification of the whole document. The domain dependency of sentiment words is also addressed through the use of composite features and a domain specific lexicon created in this work. It was found that the use of transitive dependencies in an aspect-focused classification is a promising area, which has the potential of improving aspect based classification once the aspects have been properly determined. It was also found that although using composite features does not necessarily improve the classification accuracy, it gives rise to context rich classifiers, and the domain specific lexicon generated performed on par with the widely used generic lexicon, SentiWordNet.

Acknowledgement

I would first and foremost love to express my heartfelt gratitude to God almighty for seeing me through this journey. I am thankful for his grace, which lifted me up when I was down and gave me the courage to go on.

I also wish to specially acknowledge my parents, especially my mother, for constantly believing in me and encouraging me to push on and not loose hope. I acknowledge my father for always challenging me to push the boundaries and keep moving forward. I am grateful for my siblings, Margaret, Angelica, MaryMagdalene, Peter, and the little twins, Beka and Sylvester, for being a constant support structure, keeping me company, and offering me someone to talk to.

I especially acknowledge a dear friend of mine, Wilmer Ricciotti for all his encouragement, assistance and companionship throughout this journey.

I acknowledge all the efforts of my supervisors, Dr. Chris Fox, and Prof. Udo Kruschwitz, for all their guidance and support throughout this journey. Special thanks goes to Dr. Chris Fox, who believed in me, still, even when I did not believe in myself. I thank him for his unfailing support and encouragement, even during challenging times.

I thank all the friends I have met here at the University of Essex, the Catholic Chaplaincy here in the University, and all my research group members, for making my stay at the University of Essex a truly memorable one!

Contents

List of Figures

List of Tables

Glossary

Notation	Description
Feature	A feature is often referred to as a defining characteristic of an object. It is used in this work to refer to the component words within a text that are extracted as characteristics of the text. These words are used to generate feature vectors for the classifier
Feature vector	A feature vector is a vector whose elements are representatives of the features in a text. They can be the frequency values, the binary values which show presence (1) or absence (0), weight values, or any other real values that represent the selected features.
Aspect	An aspect is a feature of a product or service, about which an opinion can be expressed. It is also referred to as an opinion target.
Seed Words	This is a small set of words with strong negative or positive associations.
Seed List	The seed list is a list of seed words or seed words list.
Dependency tree	A dependency tree is a graphical representation of a sentence where the nodes correspond to words of the sentence, and the edges represent the syntactic relations between them.
Dependency relation	This is a triple, which shows the relationship type between two words, a head words and a dependent word. These relations are extracted from the dependency tree of a sentence, and show the non-local relations between words in a sentence.
Long range dependencies	The term long range dependencies is used to refer to the relationships between words in a sentence which have been captured by the dependency relations. They are different from short-range dependencies which are captured by ngrams.

Notation	Description
Ngrams	These are a set of co-occuring words in a sentence, within a certain window. They are usually extracted from consecutive words.
WLLS/WLLR	This is a scoring scheme which has the merit of capturing relevancy, with respect to each class. The log ratio gives a low score to entities which are uniformly distributed over all classes, and gives a high score to those which are more specific to classes.
Wildcarding	A technique in which some words in a dependency tree or subtree are replaced by a generic node which can match any term.
PMI/IR	Point-wise Mutual Information and Information Retrieval. This computes the mutual information between a given phrase and a highly negative or positive word. Information Retrieval is added when a search engine is used to retrieve these relationship.
POS	Part-of-speech. This is a linguistic categorization of words, for example, nouns, verbs and adverbs.
POS tagger	A software which reads in text and assigns a part of speech to each of the constituent words.
Term polarity	The measure of the positivity or negativity of a term.

Part I

Theory

Chapter 1

Introduction

Sentiment Analysis is a branch of text analysis, which rather than focus on topics, focuses on opinions expressed in text. It has become more relevant and beneficial to analyze opinions expressed in text. As the internet becomes more widespread and accessible to people all over the world, so has the field of Sentiment Analysis experienced a surge in applications, and in the development of approaches for analysing different forms of opinions.

The field of Sentiment Analysis has seen applications in the business sector, especially in market intelligence, in the political sector, in social web mining and analysis, as well as in customer reviews analysis.

Cutting across the fields of Social Science, Psychology, Linguistics and Computer Science, the tools developed for sentiment analysis vary, in order to cater to these fields.

Sentiment Analysis has not been limited to text only, but there is also work done in the areas of sentiment in speech analysis, and facial sentiment detection.

There are two broad areas of Sentiment Analysis; Subjectivity Detection and Sentiment or Polarity classification. Subjectivity Detection focuses on identifying opinionated/subjective sentences in text, and sometimes, leads on to polarity classification. Polarity classification focuses on the determination of the sentiment orientation of the text. This sentiment orientation refers to what type of sentiment is expressed, positive, negative or neutral.

There are two major approaches to polarity classification, the Machine Learning approach

and the Lexicon-based approach. The Machine Learning approach is seen to consider the sentiment classification problem as a topic classification problem, and hence, one which can be solved by applying machine learning algorithms directly to it. The lexicon-based approach mainly focuses on determining the sentiment in text through the use of lexicons.

While the Machine Learning approach performs polarity classification with a high accuracy, it is seen to produce results which are not as easily explainable as those produced from the lexicon-based approaches. The Machine Learning approaches are also easily affected by the domain style and time dependencies. They are domain dependent, and a classifier trained on one domain cannot be easily transferable to another domain.

The lexicon-based approaches on the other hand are domain independent, but do not achieve the accuracy that is obtainable from machine learning approaches.

Our motivation in utilizing a hybrid approach is to take advantage of the high accuracies reported for machine learning algorithms, and to incorporate our own features with the aim of obtaining a classifier that is trained on a rich set of linguistic features. This hybrid approach also aims to discover features which influence classification accuracy.

1.1 Challenges in Sentiment Analysis

The area of Sentiment analysis has a number of challenges, mostly owing to the fact that opinions are difficult to classify. Some of the challenges which are still prevalent in the field are outlined here.

- "The whole is not always the sum of its parts". Document level sentiment classification sometimes assumes a document to be a bag of words, and aims to give a general classification to the document. Such an approach stands the risk of missing out on the opinions which are expressed about components within the document, which though might be useful, may have an opposite sentiment polarity to that of the overall document classification. As such, it is important for a classification approach to take these sentiments into consideration. This poses a challenge, as more fine-grained approaches

are required. This generic classification also leads to other sub-challenges such as:

- A disregard of the varying degrees of sentiments expressed within the document. The classification of sentiment orientation as mere positive and negative, and sometimes neutral, without regard to the degree of polarity/sentiment may pose some problems. In some occasions, opinions are assigned a degree of polarity, for instance, a positive opinion can be weakly positive, mildly positive, or strongly positive [73]. This approach models sentiment orientation more realistically.
- Sentence/ Document complexity is not considered. Sentence complexity also plays quite an important role in making sentiment classification a complex issue. Some approaches have classified complex sentences by breaking them up, and solving each part separately [107]. Not analysing the whole sentence structure as a whole may lead to the overall sentiment being wrongly classified, and hence could affect the accuracy. Document complexity can also pose a big problem for document sentiment classification. An example is in movie reviews where the reviewer can express opinions about different aspects of the movie, making it difficult to pick the actual sentiment of the reviewer towards the movie. One approach to handling this could be breaking down the document into different aspects, and classifying these aspects separately, instead of the entire document.

- "What is good for the goose, is not always good for the gander". Opinions are expressed in a different way than topics are, hence, applying topic-based classification approaches directly, would not yield the same results, or be as efficient as these algorithms are in classifying topics. Sentiment classification tools and algorithms are still being developed. In another vein, lexicons, which play a very important role in sentiment classification are mainly generic lexicons, while sentiment bearing words are mostly domain specific, hence these lexicons may not always be sufficient. Under this challenge, we consider:
  - Domain Dependence : As mentioned above, most sentiment words' polarity is

domain dependent. This poses a problem with developing generic lexicons or classification approaches aimed to work across all domains. Related to the issue of contextual sentiment, the domain information is believed to improve classification accuracy. This leads to the next point on sentiment lexicons.

- Sentiment Lexicons : The unavailability of a standard lexicon for sentiment analysis is another challenging issue. Though there have been a number of developed lexicons, the method of development, which involves manual definition of sentiment expressions by denoting the polarity, part-of-speech (POS) tagging, canonical form and argument type (subject or object) for the corresponding sentiment words, is tedious, inefficient, and could also be inaccurate [95].There is also a lack of collaboration within the sentiment analysis community, because these lexicons are not made publicly available. In constructing SentiWordNet, [27] made a laudable contribution, but there are still many features of the terms in the sense of subjectivity and sentiment orientation yet to be added to solve the problems still remaining in this field.

- Contextual Sentiment : The context environment is very important in sentiment orientation identification. The same words in different contexts can have different part-of-speech (POS) tags, as well as different meanings. POS tagging in this case, serves as a good pre-processing tool, to help with word sentiment polarity identification. Though some words may have the same POS tag, they may have different meanings and sentiment orientations, in different contexts. Also, a worst case can occur in which words in different contexts have the same POS tag and meaning, but different sentiment orientations.

- Heterogeneous Documents : Sentiment classification of documents of different types, or documents of the same type, from different domains varies in terms of difficulty. This may be as a result of differences in the level of sentence/document complexity for different domains/types.

- Sentiment bearing words may not always be in close proximity of the words they are modifying. As such, approaches which attempt to extract words, together with the sentiments connected to them based on direct neighbour relations may miss important and relevant relations which do not appear next to each other.This could also lead to mishandling of negations in sentences,for example, though this is not covered in this work.

- Finding the right features, or feature combination for polarity classification is still a challenging aspect in Sentiment Analysis. These features could be the class of words to be considered as sentiment bearing terms, or the frequency with which such words are used, or just the presence or absence of these words in the text. Certain features would be more suitable for classification than others, and the challenge lies in being able to determine what these features are, as well as determining techniques which combine them adequately for sentiment classification.

1.2 Research Questions

This work aims to address the following research questions;

- Can we learn which sentences correlate well with the overall sentiment classification? Can such sentences be extracted to generate valid representative summaries of the opinions expressed within a body of text?

- A document can comprise of different sentiments expressed towards various aspects, which could be different from the overall sentiment expressed by the document when it is considered as a whole. How can we extract the component sentiments within the document, and how do we recognise the targets/aspects of each the constituent sentiments?

- Can we learn from sentence or clause level word correlations to take account of polarity contexts?

- Are there certain features which can be utilized for classification that would directly influence the overall polarity classification accuracy? How do we determine such a feature set in terms of what features it should be made up of?
- The polarity of sentiment bearing terms can be domain dependent. How do we derive a classification approach that addresses this challenge?
- Does the proposed approach generalise to other problem domains, in terms of applicability?

1.3 Goals

The first goal of this work is to perform sentiment classification of documents into two classes, positive and negative, taking into consideration the sentiments expressed about the various aspects (be they movie aspects or product aspects), in deciding the overall document polarity. This work will perform sentiment classification at the document level, but will also identify the potential aspects about which opinions are expressed. The polarity of the document will be determined as an aggregation of the sentiments expressed in relation to the various aspects.

The second goal of this work is to develop an approach which in addition to being aspect-focused, also tackles the issues associated with domain dependency of sentiment bearing terms. Such an approach will be easily adaptable to other domains, other than those considered in this work.

The third goal of this work is to determine a feature set which a machine learning algorithm will be trained on, which will invariably lead to improvements in the classification accuracy.

An additional goal of this work is to determine the influence of certain sections of a body of text on the overall document polarity. The work aims to identify such sections, and investigate the suitability of such a section for representing the whole text in sentiment

classification. This will lead to ease of classification, as classification systems will only have to work on a certain section of the document, rather than the whole,. It will also feed into recommender systems and search engines which will display these sections of the text to users, and save them having to read the full documents to determine the sentiment expressed.

A further goal of this work is to investigate the existing approaches which are the machine learning and lexicon-based approaches to determine their suitability for sentiment classification, and hence justify the need for a new hybrid approach that combines the two in a novel way.

1.4 Contributions

This thesis makes the following contributions:

- A novel hybrid sentiment classification approach that performs aspect-focused document level sentiment/polarity classification. Our approach first performs aspect identification, and then utilizes the property of transitivity to extract long-range dependencies between the aspects and sentiment bearing terms in a novel way. These relationships are then added to the feature set and used in conjunction with other features to determine the overall sentiment of the text.

- A novel technique to tackle the issues associated with domain dependency of sentiment bearing terms in the new hybrid approach, through the use of composite features. In achieving this, the following are performed:

  - A domain specific lexicon is generated through a bootstrapping approach using two existing lexicons, and the focus domain. This process was performed using the corpus only, with no external resources, such as online lists of terms, to ensure that where such external resources are not available, the approach would still be functional and executable. This lack of reliance on domain-specific resources helps ensure portability across multiple domains.

  - A set of composite features were generated utilizing parts-of-speech, prior polarity and contextual polarities determined through our domain specific lexicon, and these were added as context rich features.

- Another contribution this work makes is an extensive experimentation with the machine learning and lexicon-based approaches with various feature combinations to determine how each feature set affects the accuracy obtained. Summarization of texts before sentiment classification is also explored under these approaches, to determine the suitability of generated summaries in capturing the sentiment of a full text. A comparison is also made between using traditional off-the-shelf summarization tools and these generated summaries to determine overall sentiment polarity.

1.5 Organization of the Thesis

This thesis is organized as follows; Chapter Two covers the relevant related work to the area of concern of this thesis. We also explain the hybrid approach, the lexicon-based approach and the machine learning approach are also covered. We explain the tools used within the different approaches to sentiment classification, which include lexicons for the lexicon-based approach, and also machine learning algorithms, for the learning based/ machine learning approach.

Chapter Three is where we discuss our approach in detail. We explain the design of the approach, in relation to what obtains in the literature, in a bid to highlight what we are introduce to the approach.

In Chapter Four, we cover one of the pre-requisite works we carried out, which involved using a purely lexicon-based approach to classify documents according to their sentiment polarity. We carry out different variations of experiments. We conduct a classification on different segments of the document, to determine the effect of summarization on classification.

Our pure learning based approach is covered in Chapter Five of this work. We conduct

a purely learning-based polarity classification to provide a baseline with which to compare our work on hybrid classification against. We also introduce the machine learning algorithm that we use in this work, and go on to show how it is implemented. We also classify the same document segments classified in Chapter Four, to determine if certain parts of a document can be classified as a representation of the whole.

Chapter Six is the Chapter where we explain and implement our hybrid approach to classification. We show how our approach works, and the classification results. we go further to validate the approach by testing it on data from other domains.

Chapter Seven is the discussion Chapter, and we also explain our future work.

Chapter 2

Related Work

In the past, text mining was essentially involved with analyzing documents or pieces of text based on topics only. As the field evolved, and the World Wide Web became more popular and accessible to more people, another branch of text mining evolved with it. This branch deals with opinions which have been expressed in the body of the text, rather than topics.

This branch of text mining is called Sentiment Analysis, and is also known as Opinion Mining. Sentiment Analysis has been referred to by a lot of names, besides Opinion Mining; Opinion Extraction, Sentiment Mining, Subjectivity Analysis, Affect Analysis, Emotion Analysis, and Review Mining, to mention a number of them. Amongst all these names, Sentiment Analysis and Opinion Mining are more frequently used interchangeably.

By way of definition, Sentiment Analysis has been defined as the field of study which is concerned with analyzing peoples' opinions, sentiments, attitudes, as well as emotions towards entities such as products, services, organizations, individuals, issues, events and topics [16].

Sentiment analysis is also used to refer to the use of natural language processing, text analysis and computational linguistics to identify and extract subjective information from source materials [48].

Sentiment Analysis is concerned with determining the sentiments, emotions as well as attitudes of a user towards a product, good or service. In achieving this, there are two main

approaches used in Sentiment Analysis. These approaches are the lexicon-based approach, and the machine learning approach. A third approach which combines the two approaches, lexicon-based and learning based, has also been applied in the area. This is referred to as the hybrid approach, and we will elaborate more on this subsequently.

The task of Sentiment Analysis may be subdivided into two; sentiment detection or subjectivity detection, and sentiment or polarity classification.

Subjectivity detection is concerned with determining if a particular piece of text is opinionated or not. That is, if the text is made up of facts or opinions. This can be achieved through the detection of sentiment carrying terms present in the text. Not all subjective text is opinionated, but subjectivity detection is usually used as a generic term for the detection of opinionated text.

Sentiment classification on the other hand is more concerned with determining what sentiment class the text belongs to. The text can either be expressing a positive sentiment, or a negative sentiment, or be neutral.

This work focuses on the sentiment classification aspect of sentiment analysis, and our discussions are in relation to this.

Whilst both Sentiment Analysis and Topic/text classification are branches under text mining, they differ greatly. Text classification tasks are concerned with features that distinguish different topics, while sentiment analysis on the other hand deals with features concerned with subjectivity, affect emotion, and points-of-view that both describe and modify the related entities [25]. Text classification is concerned with facts, while Sentiment Analysis is concerned with opinions. Facts are objective expressions which describe entities, events and properties, whereas, opinions are subjective, and they describe the emotions, feelings and sentiments towards the entities, and their properties [80].

There have been various approaches proposed and implemented to deal with Sentiment Analysis, and the work done on these will be discussed in this Chapter. There are also a host of challenges connected with text classification based on opinions, one of which is the complexity of the natural language, and it's differing expressions. Chinsha and Joseph

[16] describe opinion mining or sentiment analysis as a hard problem, due to the highly unstructured nature of natural language and the difficulty which is encountered in getting a machine to interpret the meaning of a sentence. We have discussed these challenges in Chapter One of this work. In subsequent sections and subsections, we will discuss on the various levels of the sentiment analysis task, applications of Sentiment Analysis, and the various domains that it has been applied in.

2.1 Applications of Sentiment Analysis

Due to the importance placed on opinions given online, and the ease of accessing these opinions, the areas of application of Sentiment Analysis has broadened, going from business management and sales, to the social media domain. Some areas of application of Sentiment Analysis are given in the subsections below.

2.1.1 Business Management and Market Intelligence

Sentiment Analysis is used in businesses and organizations to provide market intelligence as well as product and service benchmarking. Opinions of previous customers are sought by new customers in the decision making process, with regards to purchasing new products. Advert placement is another key area of market intelligence, where Sentiment Analysis is applied. It involves placing the advert of a product or service in a site where the sentiment about the said product or service is detected to be positive, or the sentiment expressed for competitors' products or services is detected as being negative [78].

Sentiment analysis is also used to gain and analyze feedback obtained from the release of a new product, and hence assists in investment decision making. Additionally, it is provides a source for predicting customer behaviour [37].

2.1.2 Politics

Sentiment Analysis has also been applied in politics in order to detect changes in potential voters' perception of a candidate, and to predict election results. It also finds application in determining election debate winners [86]. Sentiment analysis can enable politicians detect reactions to policy changes, and hence, make the decision to either go ahead with the changes, or change tact. Voting advice applications like SmartVote.ch, also assists voters with determining which political party has a close position to theirs. SmartVote.ch works by asking the voter to declare his/her degree of agreement with a select number of policy statements, and then matching his/her position with that of the political parties [31].

2.1.3 Trend Detection

Another area of application of Sentiment Analysis in social media is the detection of trends and spikes. This information can further be utilized by organisations or news sites as they wish.

2.1.4 Hate Speech/ Bullying

Sentiment Analysis is also used to detect hate speech from the social web, and this can be utilized by security services. Online Bullying can be detected with Sentiment Analysis, especially on social media websites where people can tend to get personal in their attacks on other users.

Sentiment Analysis can be used to detect overly heated arguments, so that antagonistic language and inappropriate remarks made about individuals can be analysed and removed using Sentiment Analysis techniques [25].

2.1.5 Consumer Decision Making

This application of Sentiment Analysis has been at the centre of a lot of work due to the reliance of new consumers on the opinion of past customers, as mentioned earlier, under

Market Intelligence. The information about this is mostly generated from online reviews and blogs. These decisions could include, but are not limited to; movies which are worth watching, holiday destinations to visit, as well as preferred restaurants to patronise.

This is not an exhaustive list of the applications of Sentiment Analysis. There are a number of other ways which this area has been applied. More applications of the area are highlighted in the next section under data sources and domains of application of Sentiment Analysis.

2.2 Data Sources/ Domains of Application

There are a number of sources through which data can be obtained for analysing sentiments. These data could be in the form of short texts, or longer texts. It could also be pre-processed, or may have to be prepared and processed before being used. Among these data sources are those listed in the subsections below.

2.2.1 Blogs

These are pages that hold the expression of an individual's personal opinions, and may contain reviews on products, social or other issues and possibly services [31]. Though blogs hold the views of one individual, this opinion has the potential of having a broad impact, because bloggers (people who write blogs) sometimes have a huge following. Blogs are hence considered as a means of spreading, sharing, or breaking information with a large audience, and hence marketing can be conducted through this medium [116]. The data generated from blogs has been analysed for Sentiment Analysis.

2.2.2 Review Sites

A number of review sites have been used as a data source throughout the literature. These sites are used by consumers of a product or service to express their opinions of the product or service. Some of these sites are CNET.com, Amazon.com (for product reviews), Epin-

ions.com, IMDB and rottentomatoes.com for movie reviews. User feedback, suggestions and opinions, such as hotel reviews and online shopping reviews which help in the decision making process, can be found on review websites [80].

2.2.3 Datasets

There are also raw datasets that have been developed through past studies, such as the Cornell Movie reviews and the Multi-Domain Sentiment Dataset (MDSA) or through competitions, like the SemEval competitions.

2.2.4 Micro-Blogging

The social web is also a rich source of data for Sentiment Analysis. Information on Twitter is usually represented with a short text message referred to as "tweets", and these have also been used to express opinions about different topics [31]. Tweets have been used widely in the literature for short text sentiment analysis [86][69]. A few other popular micro-blogging websites are Facebook, MySpace and Tumblr.

2.2.5 Newspapers

Newspapers are a traditional, but still effective tool for individuals to share their views and thoughts, and also for business organizations to market their products [116]. In recent years, there have been a number of online newspapers, which makes it easier to employ automated techniques in mining information from them. The purpose of analysing news articles for expressions of sentiment could also be focused on detecting the polarities of opinions expressed in references to companies or organisations [15]. The data collected from such opinions would further be utilized by the companies or organisations.

2.3 The Task of Sentiment Analysis

The task of Sentiment Analysis may be subdivided into two; sentiment detection or subjectivity detection, and sentiment or polarity classification.

Subjectivity detection is concerned with determining if a particular piece of text is opinionated or not. That is, if the text is made up of facts or of opinions. This can be through the detection of sentiment carrying terms present in the text. Not all subjective text is opinionated, but this subjectivity detection is usually used as a generic term for the detection of opinionated text.

Sentiment classification on the other hand is more concerned with determining what sentiment class the text belongs to. The text can either be expressing a positive sentiment, a negative sentiment, or be neutral.

The focus of this work is on the sentiment classification aspect of Sentiment Analysis. In the next sections, we will go on to explain the various approaches to Sentiment Analysis, and the levels of text at which this analysis is performed.

2.4 Approaches to Sentiment Analysis

The approaches to Sentiment Analysis can be described as the techniques used for Sentiment Classification. There are three major approaches:

- Machine Learning Approach
- Lexicon-based Approach
- Hybrid Approach

These approaches are divided into components, which can be further divided into other subcomponents. Figure 2.1 depicts these applied approaches, and the subdivisions that follow. In this section, we will explain these approaches in detail, and discuss their applicability in Sentiment Analysis, particularly in the aspect of polarity classification.

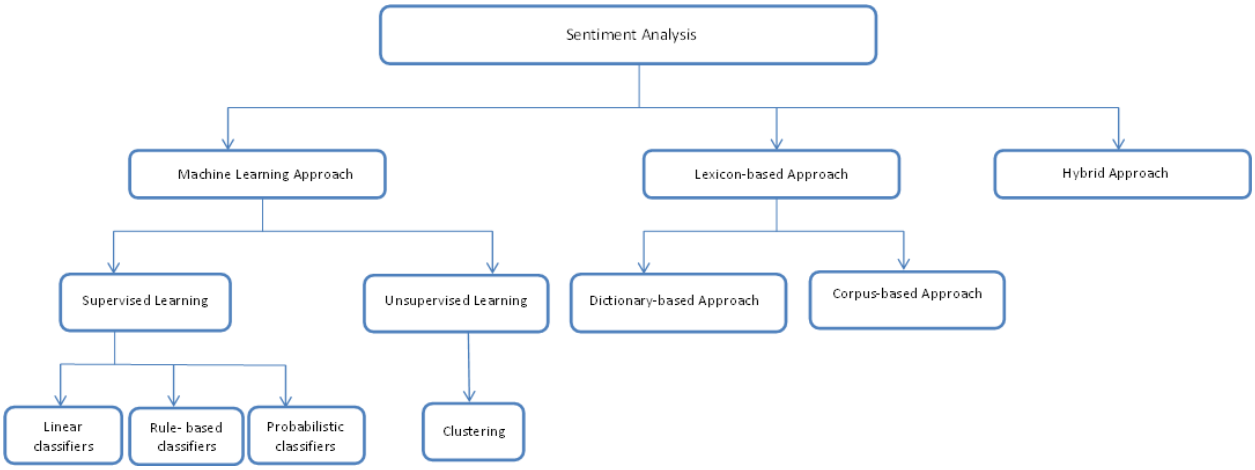

Figure 2.1: Components of Sentiment. Analysis adapted from [98]

2.4.1 The Lexicon-based Approach

The Lexicon-based approach is usually characterized as an approach which bases the definition of sentiment on the analysis of individual words and/or phrases, and uses emotional dictionaries, from where emotional lexical items are searched, and their sentiment weights calculated [11].

The lexical approach, sometimes referred to as the semantic orientation approach, relies solely on lexical resources such as lexicons to determine the semantic orientation of a document. Semantic orientation is the measure of the subjectivity and opinion in text. It captures the polarity; which is negative or positive, and the strength of sentiment towards a subject topic, a person, or an idea[92].

The assumption of a basic lexical approach is that the contextual sentiment orientation of a piece of text or document is the sum of the sentiment orientation of its constituent words or phrases [71].

The general architecture of the lexicon-based approach is depicted in the structure given in Figure 2.2.

Some of the strengths of the Lexicon-based approach as described in the literature is that in order to carry out sentiment classification, it does not require prior training on the data to be classified, and offers better generality [18]. They are also said to provide easily

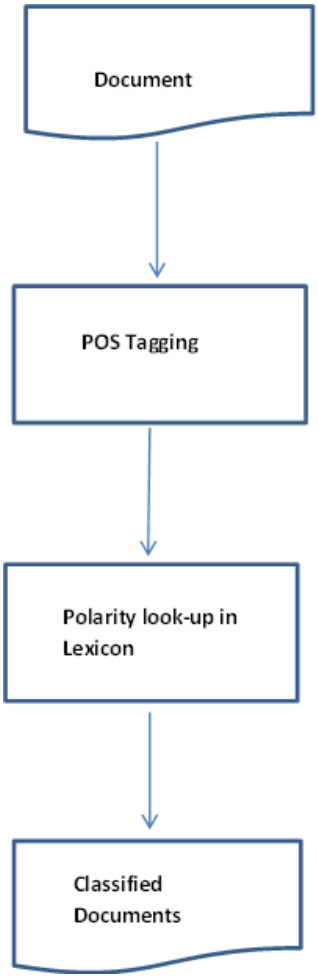

Figure 2.2: General Architecture of Lexical Approach Classification

explainable results [11].

Lexicon-based approaches are also seen to be domain independent, that is, not being reliant on the properties of the data in a certain domain. This is achieved through their use of lexicons to determine the sentiment orientation of words [86]. This can also constitute a drawback for the approach, as these methods are described as being restricted by their lexicons, and forcing words from other domains to take up a polarity based on the lexicon may lead to low precision, due to the domain dependency of certain words [82].

Another drawback of the Lexicon-based approach is that the lexicons may not always be available [11].

The Lexicon-based approach is further split into two sub-approaches, corpus-based approaches and the dictionary-based approach.

We explain these two sub-approaches in the subsections below.

2.4.1.1 Corpus-based approach

Corpus-based techniques work by trying to find co-occurrence patterns of words, in order to determine their sentiment [18]. They utilize the statistics of very large corpus in order to be able to make this classification[104], and this can also constitute a disadvantage, especially in the case where such a corpus is unavailable. Corpus based methods usually employ a set of seed words of known sentiment orientation, and exploit their co-occurrence patterns with other words, in order to identify new sentiment words, and subsequently determine their polarity in a large corpus [113].

Notable approaches that have utilized the corpus-based approach are [36], who discovered opinionated adjectives from a large Wall Street Journal corpus, through the study of conjunctions. They determined that adjectives which were linked with the conjunction "and", were more likely to share the same polarity, while those linked with the conjunction "but", were more likely to have opposite polarities. Thus, the knowledge of the polarity of one of the pair, made it easier to infer the polarity of the other.

Turney in [97] determined the polarity (positivity or negativity) of phrases by computing the PMI - Point-Wise Mutual Information and Information Retrieval for their co-occurence with two words, each of which was identified as being highly polar, with respect to its sentiment class. The two words were "excellent" and "poor", and phrases which occurred in the web more frequently with "excellent" were classified as positive, while words which occurred more frequently with "poor" were classified as negative. The overall polarity of the text was computed as the average semantic orientation of all the phrases that contained adjectives and adverbs. Turney's work was based on the assumption that positive words will tend to co-occur with other positive words, while negative words will tend to co-occur with other negative words [82].

Riloff et al [83] manually evaluated and selected seed words and iteratively collected opinionated nouns, based on a developed template.

Corpus-based techniques have the disadvantage of difficulty in obtaining such a large corpus, but also have the advantage of simplicity, and the ability to discover domain-specific opinion words and their orientations, if a corpus from a specific domain is used for the discovery process [104][113]. In addition to this, another advantage of a method that relies on corpus-based approaches is that due to the fact that domain-dependent information is extracted, the method can automatically be adapted to a new domain when the corpus is changed [36].

2.4.1.2 Dictionary-based Approach

Dictionary-based approaches mainly employ the strategy of manually composing the sentiment orientation of a set of opinion words, and then growing this list through a search of a well known lexicographical resource, such as WordNet, to extract their synonyms and antonyms. The newly found words are added to the seed list, and the search continues iteratively, until no more new words are discovered [98]. These are then vetted manually, to correct any errors.

Dictionary-based techniques use synonyms, antonyms, as well as hierarchies which exist in WordNet, or other sentiment lexicons to determine the sentiment of words [18].

Kamps in [42] found the polarities of adjectives in WordNet by measuring the relative distance of each term from predefined sentiment.

Hu and Liu [38] utilize the antonym and synonym set of WordNet to determine the semantic orientation of adjectives. Working on the intuition that adjectives share the same orientation with their synonyms, and opposite orientations with their antonyms, they manually came up with a seed list comprising of very common adjectives, then grew the list by searching in WordNet.

Adjectives are not the only word groups which are indicators of sentiment. There are other word groups which also act as sentiment indicators. A simple method was proposed in [47], where all the synonyms of a polar word were added to a list with the same polarity, while its antonyms were added with an opposite polarity.

The dictionary-based approach is seen to be more effective than the corpus-based technique, especially due to the latter's reliance on a large corpus, which may not always be available. They are seen to be more advantageous than corpus-based techniques because they are domain independent, and rely on the existing resources, not on specific search facilities [44].

This approach has the advantage that a large dictionary of opinionated words can be built, even though these dictionaries may contain ambiguous words [104].

A disadvantage of this approach is that a number of issues could arise from the manual construction of the seed list, making it more desirable and efficient to determine an automatic approach to determining the seed list, if this is to be avoided.

2.4.1.3 Lexicons

A Sentiment lexicon is a set of words indicating a certain sentiment aspect [76]. A lexicon usually contains words, together with their sentiment polarities. Lexicon-based methods are usually focused on using a lexicon to compare as well as provide the polarity to the desired opinion words, and then assign a score based on the results [80].

We will discuss the commonly used Lexicons and the concept of employing lexicons under two categories, generic lexicons and domain-specific lexicons.

Generic Opinion Lexicons These are lexicons with a fixed categorization of sentiment words. Once they have been constructed, these lexicons are applicable to a wide variety of domains. They have static sentiment values, regardless of what context the word is used in [86]. Some of these lexicons are lexicons like the General Inquirer [89], WordNet [57], SentiWordNet [27], the Subjectivity lexicon [83] and the Google PMI dictionary [91]. Some of these lexicons only contain information of the relation between words, but do not provide the semantic orientation information of each word. An exapmle of this is WordNet.

WordNet WordNet [57], is a lexical database for the English Language which groups English words into sets of synonyms, referred to as synsets. It also provides short, general

definitions, as well as records the various semantic relations between synonym sets [32]. These short general definitions are referred to as WordNet's glosses. A WORDNET synset represents a unique sense, defined by a unique gloss, and is associated to a set of terms, all with the same part-of-speech (POS). Each one of these synset is associated to a sense number, for example, the adjective blasphemous (2), blue (4), profane(1), are all contained in the same synset, whose sense is defined by the gloss characterised by profanity or cursing [26].

As WordNet does not include semantic orientation information for each word, [38] applied WordNet in determining sentiment tags by utilizing the adjective synonym and antonym set to predict the semantic orientations of adjectives. Agarwal and Bhattacharyya [1] determined the strength of adjectives in a polarity classification approach using WordNet synonymy graph.

SentiWordNet SentiWordNet is a lexical resource produced by asking an automated classifier A to associate to each synset s of WordNet (version 2.0), a triplet of scores A(s,p) (for p P = positive, negative, objective), describing how strongly the terms contained in s enjoy each of the three properties. The score triplet is derived by combining the results produced by a committee of eight ternary classifiers, all characterised by similar accuracy levels but extremely different classification behaviour [27]. SentiWordNet provides each synset of Wordnet with a triplet of polarity scores (positivity, negativity, objectivity), and the value of this triplet adds up to "1" [24].

The SentiWordNet lexicon has been used quite broadly for the purposes of sentiment classification. SentiWordNet has been used in conjunction with WordNet's semantic content for sentiment polarity detection in financial news.

It has been used as a resource for the identification of sentiment carrying words [7][10][79], and estimating the probability that a document contains opinion bearing expression through the extraction of subjective adjectives from SentiWordNet [112].

The semantic score of subjective sentences was extracted from SentiWordNet and utilized

for the calculation of their polarity, based on the sentence structure [34][44]. SentiWordNet scores were also used in [5] to determine the sentiment polarity of sentences.

Sentiment classification across multi-domains using SentiWordNet as a lexical resource, was explored in [24], where SentiWordNet scores were exploited as features for classification. The results obtained were compared with machine learning classification features across the same domains, and were found to be outperformed by the machine learning results.

Domain Specific Lexicons Opinion and subjectivity are quite domain dependent, with the same words sometimes offering different meanings in different domains [25].

Common examples of such words are words like "unpredictable", which have a positive meaning when used in the movie review domain, but a negative meaning if used in product domains, such as automobile reviews.

Generic opinion lexicons like SentiWordNet normally struggle with slow adaptability to different domains, in addition to the fact that their embedded sentiment words make it difficult for them to determine the sentiment orientation of words with regard to the context of its use [76]. Park et al [76] propose that a solution to this inadaptabilty and lack of context awareness of generic opinion lexicons is a dynamic generation of sentiment lexicons which are based upon a specific domain.

Yang et al [104] attribute the difficulty in maintaining a universal sentiment lexicon for general cases to the context-dependency of sentiment bearing words. As such, lexicon-based methods are seen to be unable to find domain dependent sentiment words since many entries in these dictionaries are domain independent. This is seen to be especially difficult when detecting the sentiment of new words which have been used in social media sites, like Twitter [104].

The domain dependency of sentiment terms has led to research into the creation of dictionaries that are more tailored towards a particular domain, or corpus. It is imperative to first explain what seed words are, before explaining in detail about domain specific lexicons. This is because seed words are often a starting point in the creation of domain specific

lexicons.

Seedwords Seed words are a small set of words which have strong negative or positive associations. Most commonly, seed words extracted and used in determining the semantic orientations of other words, and subsequently, other sentences or documents are adjectives. This stems from the belief that adjectives are a strong indicator of subjectivity and sentiment. In determining semantic orientation, [38] selected a set of seed adjectives and utilized this with WordNet, in the prediction of almost all adjective words in the review collection. This was done under the presumption that adjectives share the same orientation as their synonyms, and opposite orientations as their antonyms. In determining the seed adjectives, commonly used adjectives with known semantic orientations were manually selected.

In principle, a positive adjective should occur more frequently with positive seed words, and as such will obtain a positive score, while negative adjectives will occur more in the neighbourhood of negative seed words, and thus obtain a negative score [92].

Examples of seed words are words like "Excellent" which has very strong associations with positive sentiment, and "Appalling", which has very strong associations with the negative sentiment. Basically, if an adjective is close in terms of synonymy to a positive word, or close in terms of antonymy to a negative word, then the adjective is classified as positive, and vice versa.

Seed words are not limited to adjectives only, and could also be other word classes, such as adverbs, verbs and some nouns [28].

In selecting seed words, the frequency of emotive words may also be used as a selection criteria, as in [115]. Seed words have been manually picked from the corpus [28]. Blinov et al [11] manually selected a subset of words which they believe clearly express positive and negative emotions, and used this to create an emotional dictionary.

There are a number of drawbacks to this manual selection of seed words from the corpus. The choice is seen as not being sufficiently reliable, based on the fact that they could easily be affected by the human subjects educational and cultural background [32]. Manual lexicon

creation is also seen to be time consuming and labor intensive, and can lead to lexicons with a limited number of words [19]. Additionally, [74] show that humans may not always have the best intuition with respect to choosing sentiment discriminating words. Labelling out of context could also pose a problem for humans, as pointed out by [36] who encountered problems in labelling some adjectives, as they were unable to decide on a unique label.

The lexicons produced through automatic approaches have been sometimes referred to as being larger, and hence susceptible to noise [92]. In their simple unsupervised approach to classification, [97] who determined semantic orientation from assessing word associations in their automatic approach, using an AltaVista search engine, highlighted the time to send queries to the search engine as a drawback of the approach.

Generation of Domain Specific Lexicons The creation of domain specific lexicons in lexicon based approaches [84][38] have recently begun with a small set of seed words which are grown through synonym detection using bootstrapping, or various online detection resources to arrive at a larger lexicon.

Manual approaches have been employed in the creation of domain specific lexicons, as have automatic appraoches. Blinov et al [11] manually selected a subset of words which they feel clearly reflect each sentiment (negative and positive), and then supplemented these words with synonyms and antonyms obtained from wiktionary[1].

In the same vein, word types are also taken into consideration. Some researchers in the literature have used only adjectival seed words, like [38], who manually created a small list of seed adjectives tagged with negative and positive labels and then polpulated this list using WordNet. This was achieved through the use of synonyms and antonyms, in generating a domain specific lexicon for identifying opinion sentences. Opinion sentences were identified as sentences that contain at least one opinion word. Hatzivassiloglou and Mckeown [36] also utilized a seed list of adjectives appearing at least twenty (20) times or more, in determining their orientation of the adjectives in their document collection. This

[1] urlhttp://www.wiktionary.org

was implemented through the analysis of a conjunction relationship.

There are a number of drawbacks to manually creating domain specific lexicons. In addition to the high cost, there are seen to not be sufficiently reliable due to influences such as a the creator's educational and cultural background [32]. Some words may be used in a different manner based on a person's level of education, as well as where they come from. In addition to this bias, manually creating lexicons is also a time consuming effort [67].

Based on the results obtained by [74] from using two different individuals to manually tag opinion words, they concluded that humans do not have the best intuition for discerning discriminating words. There have also been issues with being unable to arrive at a unique sentiment label for certain adjectives out of context [36]. Such adjectives have subsequently been left out of consideration, even though they might have contributed to the classification process, had they been used.

Automatically creating lexicons can be done by association, using the proximity information of the polar word, in relation to one or more seed words. Automatic methods have the advantage of discovering novel words, which are new words used on the web [92]. It also has the disadvantage of producing unstable dictionaries, for example, when using a Google search engine for this, the results obtained for each word could be subject to change with every search [91].

A midpoint approach appears to be the preferred choice. Such an approach is utilized in [94], who first collect a set of domain specific opinion words for the movie review domain by semi-automatically analyzing a different dataset to theirs but from the same domain, and selecting opinion words considered important based on their calculated information gain. The selected words are then manually examined and added to the domain specific lexicon, if considered domain specific.

2.4.2 Machine Learning Approach

The second approach used for sentiment classification is the machine learning approach. generally, machine learning is taken to encompass computing procedures which are based

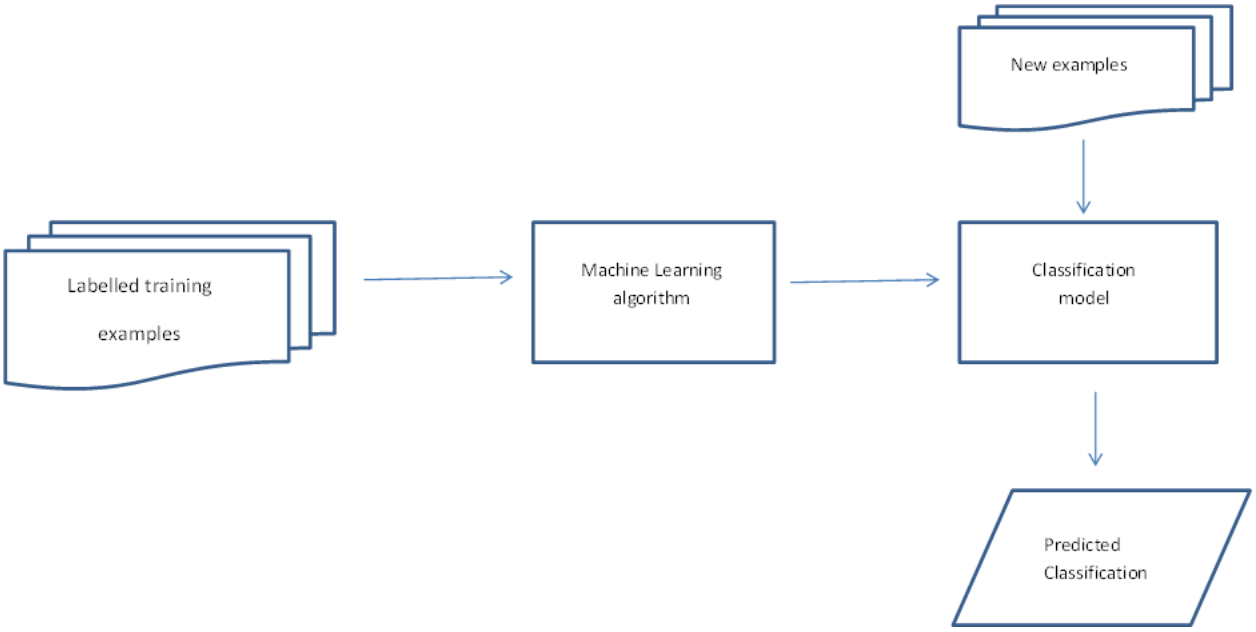

Figure 2.3: General Architecture of Machine Learning Approach Classification

on logical operations, and/or binary operations that learn from a series of examples. These examples are referred to as training examples. The training is usually done using various algorithms, after which another set of unseen examples referred to as a test set are presented to the algorithms for implementation.

The Machine learning approach tends to treat the task of Sentiment Analysis as a simple topic-based text classification problem. Though it does achieve high accuracies, this simple classification provides limited information about sentiment topic or rationale [59]. Pang et al [74] found that machine learning algorithms did not perform as well in sentiment classification as they did in topic classification. They speculated that this was probably due to topics being often identifiable solely by the presence of keywords, while sentiments can be expressed in a more subtle manner.

A generic structure of the machine learning approach is given in Figure 2.3.

There are two approaches of machine learning used for classification:

- Supervised Learning
- Unsupervised Learning

2.4.2.1 Supervised Learning

Supervised machine learning techniques are used to classify documents or sentences into a finite set of classes. The machine learning algorithm is required to generalize from the training data to previously unseen data in a way which is considered reasonable [78]. For these approaches to work, they require the availability of labelled data. Labelled data is data whose classification has been predetermined by humans.

Supervised machine learning techniques require a large corpus of training data, and their performance depends on the existence of a good match between the training and the test data, with respect to domain, topic, as well as time-period [113].

There are a number of supervised learning algorithms that have been commonly used in the literature. We give a list of these here, and go further to explain the four which have been broadly utilized in the literature.

- Naive bayes
- Maximum Entropy
- Decision Trees
- Support Vector Machines (SVM)
- Hidden Markov Models (HMM)
- Random Forests

Supervised Machine Learning Algorithms in Sentiment Analysis

Naive Bayes The Naive Bayes classifier is a probabilistic classifier which makes use of Bayes Theorem [74] with strong independence assumptions. These assumptions are seen as simplistic, as they assume that all attributes of the classified objects are independent [37].

Despite the simplifying assumptions of this algorithm, it has been shown to perform well. It has the advantage of being easy to train, where even a small training set is sufficient to

train the classifier [116]. Another advantage of this algorithm is that its training time is significantly smaller [37].

The conditional independence of Naive Bayes does not hold in real world situations, but it still surprisingly performs quite well, but not quite as well as other sophisticated algorithms like SVM and Maximum Entropy [74].

Maximum Entropy The Maximum Entropy (Max Ent) algorithm has been proven to be a good approach in many applications dealing with automatic text processing. Being more effective than the Naive Bayes classifier, it is not simplified by the assumption of statistical independence, as the former is [37].

Max Ent makes no assumptions about the relationships that exists between features, and hence, might perform better when conditional independence assumptions are not met [74].

Decision Trees The principle by which the decision tree works is to split the dataset recursively into subsets, in such a way as to ensure that each subset contains more or less homogeneous states of a predictable nature [21]. The decision tree can be defined as a tree where each non-leaf node of the tree is one test about the variable value, and the branches represent the possible results [37].

In decision tree classifiers, the training data spaces are hierarchically decomposed, and in this decomposition process, the conditions on the attribute are utilized to split the data [81].The splitting of the data space or this decomposition of the data is performed in a recursive manner to the stage when the leaf nodes may contain at least some number of the records which then are utilized for classification [51].

The decision tree is a technique applied in data mining and in machine learning. There are two major types of these trees; the classification tree, and the regression tree. The classification tree analysis is applied when the prediction output is discrete classes, while the regression tree is used when the predicition outcome is a continous value [110]. The J48 and C45 decision tree classifiers are inbuilt in Weka, which is an open source Java tool and can

be used for text classification[1].

Support Vector Machines The goal of the Support Vector Machines (SVM) is to find a hypothesis which can guarantee the lowest generalization error. This generalization error hypothesis is the probability that the hypothesis is false for a randomly selected object from the test set [37]. SVMs are large margin classifiers,and not probabilistic classifiers like Naive bayes and Maximum Entropy. Though they require a large dataset to enable them build a super quality model for classification [17][116], Support Vector Machines have been shown to be highly effective in traditional text classification and outperform Naive Bayes classifiers [74].

Support Vector Machines are one of the most popular supervised learning algorithms, not just in the area of Sentiment Analysis, but also in other areas of classification. Support Vector Machines are fast algorithms and perform with state-of-the-art accuracy. They are used for creating feature-vector-based classifiers. Each instance which is to be classified is basically represented by a vector of real-numbered features.

The text to be classified is converted into word vectors, and a hyperplane is drawn using these word vectors to seperate the data instances of one class from the other. SVM uses training instances also known as support vectors to find this hyper-plane [3]. The training data is used to generate a high-dimensional space which can be divided by this hyperplane, between the positive and negative instances, as the case may be. New instances are subsequently classified by finding their position in space with respect to the hyperplane.

Symbolically, this search corresponds to a constrained optimisation problem which lets $c_j \in \{1, -1\}$ which correspond to the positive and negative class, be the correct class of a document d_j. The solution will be written as shown:

$$\vec{v} := \sum_j \alpha_j c_j \vec{d_j}, \alpha \geq 0, \tag{2.1}$$

Where α_j's are obtained from solving a dual optimization problem. Those $\vec{d_j}$ where

[1]urlhttp://www.cs.waikato.ac.nz/ml/weka/

α_j is greater than zero are referred to as 'support vectors', as they happen to be the only document vectors which contribute to $\vec{v}$. Classification of test instances is therefore about determining which side of $\vec{v}$'s hyperplane they fall on [74].

Support Vector Machines are good at combining diverse information sources. They do not assume that features are independent, and deal well with overlaps in information sources. They are however sensitive to sparse and insufficient data. They also have the capability to handle high dimensional spaces through their use of kernels.

For linearly inseperable problems, the kernel function of SVM can be used to concert low dimensional space nonlinear problems to high dimension space linear problems. The mapping of the kernel function can act as a good control of the computational complexity of nonlinear expansion, and avoid the curse of dimensionality [114].

Application of Supervised Learning Algorithms in Sentiment Analysis The Bayesian classifier was used by [20], as one of the algorithms implemented in their work for extracting small investor sentiment from stock message boards. The Bayesian classifier relies on a multivariate application of Bayes theorem, and makes use of word-based probabilities. As such, it is indifferent to the language structure. This language independence gives it a wide applicability, which enables investigation of message boards in other financial markets, where the underlying language may not be English.

Pang et al [74] consider the problem of classifying documents not by topic, but by overall sentiment, through the use of three different machine learning techniques; Nave Bayes, Maximum Entropy and Support Vector Machines (SVMs). They implemented these algorithms on a movie reviews domain, the Internet Movie Database (IMDb). The aim was to examine if it was enough to consider sentiment classification as just another form of topic classification. They found Nave Bayes performed the poorest, while SVM performed the best, though the difference margin was not so large. They also reported that these algorithms had given better results on previous studies in topic-based classification.

In analyzing the performance efficiency of machine learning algorithms against their

lexical counterparts, [4] carried out a comparative analysis of the effectiveness of machine learning techniques on sentiment polarization in movie blogs, in comaprison to that of lexical approaches. From the results obtained, they concluded that machine learning techniques were superior, as even the worst machine learning result outperformed the best lexical results. They suggested that this may be due to the heavy reliance of lexical approaches on semantic information.

Others, such as [73], [72] and [4] first carried out the conversion of a document into vectors and then used classifiers, which included SVM, Maximum Entropy, Nave Bayes, and ADTree to categorize the documents.

2.4.2.2 Unsupervised Learning

Unsupervised learning is concerned with classifying documents into a random number of predefined categories. This approach does not require a labelled dataset in order to perform classification. Some unsupervised approaches reported in the literature are clustering, as well as deep learning methods [113].

According to [22], there has been relatively little work on sentiment-based clustering and the related task of unsupervised polarity classification, where the focus is to cluster or to classify a set of documents such as reviews, according to the polarity (thumbs up or thumbs down) expressed by the author of the review in an unsupervised manner [22]. Clustering can be viewed as a means to overcoming the weakness of existing supervised polarity classification systems, which are typically domain and language specific. A novel approach to clustering by incorporating user feedback is proposed, where the user is required to select a dimension by examining a small number of features for each dimension. This helps direct the clustering algorithm on the dimension to cluster along. Spectral clustering is first applied to reveal the most important dimensions which exist in the data, and then the user gets to select the dimension they desire. The dataset is then clustered along this dimension.

Patra et al in [79] describe the task of automatic classification as a classic example of pattern recognition, where a classifier assigns labels to the test data based on the labels

of the training data. They also describe document classification as the task of assigning a document to one or more classes.

An unsupervised approach is also implemented by [101] in order to automatically locate the facet discussed in Chinese reviews, and to recognise the sentiment expressed in the different facets. Using the Latent Dirichlet Allocation (LDA) model, they carry out a discovery of multi-aspect global topics, and then extract the local topic and associated sentiment based on a sliding window context over the review text.

2.4.2.3 Features

Features are a very important component of machine learning algorithms, as the choice of features has a huge impact on the performance of the classifier. Features are determined from the words in the document collection.

We discuss the feature selection techniques as well as the weighting schemes that have been applied in the field in this section.

Feature Selection Not every feature adds to the polarity classification of a document, sentence or phrase. The use of some features could introduce noise and lead to errors in classification.

The aim of feature selection is to identify a minimal-sized subset of features which are relevant to the target domain [18]. There are a number of feature selection techniques that are applied to a set of features, to filter out the distinctive and unique features that contribute to determining the polarity of a body of text. Among these techniques are PMI-IR (Pointwise Mutual Information and Information Retrieval), which was applied in [97], and measures the similarity between pairs of words or phrases, Information Gain (IG), which measures the contribution of a certain word to the polarity of text, and Weighted Log Likelihood Score or ratio (WLLS or WLLR), whose value shows the relevance of a feature to a certain class.

Feature selection has been reported to further improve performance of the sentiment classification process, especially for large datasets in [18]. They used feature selection in a

machine learning classification approach, and reported an increase in accuracy of the SVM classifier, after carrying out feature selection.

In order to identify these features which contribute to the determination of sentiment polarity, a scoring scheme can be used to identify the most informative features [87][94].

Amongst the feature selection techniques are:

- Entropy
- Information Gain
- Weighted Log-likelihood Scheme

Weighted Log-Likelihood Scheme/Ratio The Weighted Log-Likelihood scheme (WLLS) was proposed by [66].

In this scoring scheme,every ngram, which can be a unigram, bigram or trigram, is assigned a weighted log-likelihood score with respect to each class of emotion. The scheme captures the relevance of an ngram with respect to each class. A low score is given to ngrams which are uniformly distributed over all classes and high scores are given to ngrams which are specific to each class.

In a similar manner to the way it is utilized for ngrams selection, this scheme has also been used for the selection of relevant dependency relations as features [66].

WLLS is defined as:

$$WLLS(w_i, c_j) = P(w_i \mid c_j) log \frac{P(w_i \mid c_j)}{P(w_i \mid \neg c_j)} \tag{2.2}$$

where,

w_i : the unigram or bigram whose score we wish to evaluate

c_j : the class (sentiment) with respect to whom the score is evaluated

$P(w_i \mid c_j)$: the ratio of count of w_i in class c_j to the count of all words in class c_j

$P(w_i \mid \neg c_j)$: the ratio of count of w_i in class $\neg c_j$ to the count of all words in class $\neg c_j$

WLLS is the weighting scheme that will be used in this work, because it has been shown to perform well in terms of selecting features which are indicators of certain polarities [66].

Information Gain Information gain is frequently employed as a term goodness criterion in the field of machine learning [105]. It is utilized as a measure of the number of bits of information obtained for category prediction by knowing the presence or absence of a term in a document [93].

Feature Weighting

Term Frequency Term frequency refers to the number of times a term occurs in a document or a sentence. The frequency of a term has been exploited as a classification measure for both the lexical and the machine learning approaches, as well as the hybrid approaches.

The term frequency count weighting scheme has been used to assign weights to features, where a feature's frequency is used as its value in a machine learning classifier.

Term Presence Term presence is a weighting scheme which measures the occurrence or non-occurrence of a word in a document or sentence.

The use of term presence as a classification feature has been found in some cases to lead to a better performance from classifiers in sentiment classification, when compared to using the trem frequency feature. This has been reported to be in contrast to what is obtainable in topic classification. A possible explanation for this is given as being that topic is conveyed mostly by particular content words that tend to be repeated [74].

Terms that appear only once are also said to be good indicators of subjectivity [100].

Term Frequency - Inverse Document Frequency (TF-IDF) The definition of term frequency (TF) is given in the section above. Inverse Document Frequency (IDF) is a measure which shows the general importance of the term [32]. A term which occurs rarely in

a document collection is regarded as a very important feature in Information Retrieval (IR) methods. Effective IR models of recent either explicitly or implicitly accommodate such a feature as an Inverse Document Frequency (IDF)heuristic [45].

The TF-IDF weight is a weight which is often used in information retrieval, as well as for text mining. It is the statistical measure used to evaluate the importance of a word to a document, and to all documents in the collection [19].

Mathematically, the TF-IDF weight of a term 'j' can be expressed as:

$$W_j = tf_j * \log \frac{D}{df_j} \tag{2.3}$$

where tf_j is the term frequency of term j in a certain document, D is the number of documents in the corpus, and df_j is the inverse document frequency [32].

Applying this weighting method ensures that the frequency of a term in a document is offset by its frequency in the corpus, and hence, produces a balancing effect, ensuring that terms which occur with high frequency in a document, but do not contribute much to the distinguishing properties of the document class, are not given more importance than necessary.

2.4.3 Existing Hybrid Approaches

Aside from the two main approaches to Sentiment Analysis described above, there is a third approach which combines lexicon-based approaches with machine learning approaches. This approach is the bybrid approach, and is especially used when a fine-grained approach to classification is desired. To perform Sentiment Analysis at a more fine-grained level, which includes using sentences, phrases or clauses, with the aim of achieving results of higher accuracy from more than just term counting or averaging scores, hybrid methods are recommended. This approach allows for the incorporation of contextual and topic-related characteristics, as well as opinion related properties of the text into the classification process.

The lexicon-based approach does not require prior training, or labelled data to enable it

carry out text classification. It also has better generality [18], which results from the lexicons being domain independent. This offers the advantage of being able to apply this approach to various domains. On the other hand, the machine learning approach tends to treat the task of sentiment classification as a simple topic-based text classification problem. Though it achieves high accuracies, this simple classification provides limited information about sentiment topic or rationale [59]. The high accuracy rate of machine learning approaches is seen to be achieved mainly through careful feature selection of labelled training texts [11].

Machine learning methods utilized for sentiment classification, especially supervised learning methods, are mostly domain dependent. Hence, training the classifiers on one domain, means that they do not perform well when applied for classification in a different domain [92][82]. This limited applicability to other subject domains, other than those which they are designed for is explained as an important disadvantage [58]. A further drawback of supervised machine learning approaches is their sensitivity to the writing style of text, as well a lack of explanation or justification for obtained classification accuracies [59].

Sentiment words have however been found to be domain dependent,where a word expresses opposite sentiment orientations, depending on the focus domain [82][115].

Hybrid approaches seek to combine the lexicon-based approaches with machine learning approaches, in order to create more content rich features for classifiers, and to incorporate lexicon information in their classification process.

Some related work in this area include [72] who classify subjective extracts, which were extracted using minimum cuts in graphs, with a machine learning classifier. They use Support Vector Machines and the Naive Bayes classifier to classify these extracts. Ng et al [66] focus on building a high performing classifier by incorporating higher order ngrams (a consecutive set of words), a lexicon of adjectives which had been manually annotated with polarity information, dependency relations and objective terms, in the feature set of a Support Vector Machine (SVM) classifier. Pak and Paroubek [70] classified their text using subgraphs extracted from sentence dependency trees as features for the SVM classifier. Rastogi et al [82] conclude that incorporating the information encoded in sentiment lexicons,

mainly domain specific lexicons can drastically improve the accuracy of sentiment classification. This conclusion is arrived at after carrying out a classification process with a domain specific lexicon and SVM.

In Table 2.1, we present an overview of some of the existing work on hybrid approach to sentiment classification. Deriving polarity scores from lexicons like the General Inquirer and SentiWordNet and using these as features in classifiers like SVM have been implemented in [43][5][41][33].

A hybrid approach which incorporates sentiment lexicons into a machine learning approach to bring about an improvement in sentiment classification in tweets is presented by [41]. They implement a novel feature weighting method, by interpolating sentiment lexicon score into unigram vectors in the SVM. Aside from using SentiWordNet as the lexicon, they also utilise an add-on lexicon, which caters to the objective words and out-of-vocabulary words, which tend to be used in tweets. They arrive at the conclusion that the add-on lexicon led to an improvement in the classification accuracy, on average, compared to using the original public lexicon.

Reporting a state-of-the-art accuracy [55], present the gains obtained from exploiting the syntactic relations between words in sentences for document sentiment classification. They extract word subsequences and dependency subtrees as features for an SVM classifier, with extensive feature selection performed. They combine these features with unigrams and bigrams which have appeared in at least two (2) distinct sentences. They conclude that feature selection played a big role in the good classification results obtained.

Aravindan and Ekbal [5] worked on classifying product reviews, using association rule mining to identify the most characteristic features of a product. Nouns and noun phrases are used as the default features, and they report encountering a lot of redundant features. To incorporate contextual information, the three preceeding and next three words surrounding the target phrase are selected as features. Information from the SentiWordNet lexicon is also included by looking up these surrounding words in the lexicon and returning the difference between the positive and negative scores of each. These scores are then fed to a classifier.

No external data source is utilized.

Other researches have opted for the creation of domain specific lexicons and utilized these for the determination of word polarity scores for classifiers [82][60][9]. The polarities of terms in these lexicons are derived from the generic lexicons and also from some external sources like online word lists [9]. Another angle that has been explored in the hybrid approach is utilizing an ensemble of classifiers that are trained from different knowledge sources like semantic values of phrases, unigrams, topic information, and sometimes, different lexicons [2][62]. In the same vein, using an ensemble of various lexicons has also been investigated [9][82]. Other syntactic information sources which have been explored include sentence position and punctuations [33], word subsequences and dependency subtree patterns [2].

Mullen and Collier [62] incorporate the semantic values of phrases and other words, determined based on their proximity information with the primary and secondary topics in the text. Mudinas et al [60] identify aspects as a means to discovering polar adjectives which affect the overall document polarity.

None of these approaches have explored the use of transitive dependencies for aspect-focused document sentiment classification, and it has also not been reported elsewhere in the literature, to the best of our knowledge.

Term Polarity Term polarity is the measure of how positive or negative a term is. In Sentiment Classification, prior polarity, which is the term's polarity in the absence of context information, as well as contextual polarity, which is the term's polarity when the context within which it is used is taken into consideration, are utilized.

In our hybrid approach, we consider two forms of polarity, which can probably be introduced into a classifier to create a more context-rich classifier. This polarity information is incorporated in the classification process in a number of ways. It could be appended to the term [6] , or it could be used in place of the term in ngrams and in dependency relations [66]. Where the polarity is used in place of a term, this accounts for a new feature, which may be a new bigram or a new dependency relation. This approach of replacing a term with

Table 2.1: Overview of hybrid approaches

Author	Objectives	Method	Dataset	Accuracy
Aravindan and Ekbal (2014)[5]	Sentence level; Aspect level sentiment	SentiWordNet scores in SVM	product reviews	79.67
Kaewpitakkun et al. (2002)[41]	Tweet level sentiment	SentiWordNet scores and SVM	Tweets	81.2
Gezici et al. (2012)[33]	Sentence level sentiment	SentiWordNet scores,sentence position, SVM, logistic regression	Hotel reviews	81.45
Kennedy and Inkpen (2006)[43]	Document sentiment	SVM + General Inquirer	Movie reviews	86.2%
Augustyniak (2014)[9]	Docs sent.	Ensemble of lexicons + C4.5 decision tree	Product and Movie reviews	Automotive (48.5), Books (50.5), Electronics (50.1), Health (48.7), Movies (50.4)
Mullen and Collier (2004)[62]	Aspect level; Document level sentiment	Hybrid SVM (PMI, Lemmas, Semantic values)	Movie and Record reviews	Movies (86.0), Records (87)
Mudinas et al.(2012)[60]	Aspect-based; Document level sentiment	domain specific lexicon + SVM	Software and Movie reviews	Software reviews (89.6), Movie reviews (82.30)
Matsumoto et al.(2005)[55]	Document level sentiment	Ngrams,syntactic information + SVM Light	Movie reviews	Set 1 (87.3), Set 2 (92.9)
Rastogi et al. (2014)[82]	Sentence level sentiment	Domain lexicons + MPQA lexicons + SVM	MDSA [12] reviews	69.2
Andreevskaia and Bergler (2008) [2]	Document level sentiment	Ensemble of classifiers (lexicon based and corpus based) + SVM	Movie reviews, news blogs and PRS	News (73.3), Movies (62.1), Blogs 5(70.9), PRS (78.0)

a more generalized feature is a form of wildcarding.

Determining the prior polarity of terms can be achieved by reading the polarity off a lexicon such as the subjectivity lexicon developed by [83], WordNet Affect [90] and also SentiWordNet [27]. These lexicons are not necessarily tailored to any domain, and as such

the polarity of a word in a certain dataset, may differ from the polarity obtainable from the lexicon.

Prior Polarity The prior polarity of a word is its polarity in the absence of context. It is the polarity of the term, which can be read directly from lexicons. This polarity has been applied in determining the contextual polairty of terms, for subsequent classification.

Thet et al [94] utilize the prior polarity of individual words for the calculation of the contextual sentiment score of a clause. They create two lexicons, a domain specific lexicon, and a generic opinion lexicon. The domain specific lexicon contains movie domain specific opinion words, and the generic opinion lexicon holds general opinion words derived from the SentiWordNet and the subjectivity lexicons. The researchers used information gain to extract important opinion words strongly associated with positive and negative movie reviews in the domain. Their approach ignores word senses in sentences, hence, the positive and negative values of the multiple senses of a word in SentiWordNet are converted to one representative positive or negative value. Ng et al [66], using the same movie review dataset, incorporated the prior polarity in their work by creating a new feature set, which comprised of bigrams in which the adjectives were replaced with their polarity label. The same was done for trigrams and dependency features, and this new feature set was added to the original feature set and WLLR was used to extract the top 5000 features from bigrams, trigrams as well as dependency relations, and these features were then classified using SVM Light, over 10-fold CV, and this led to improvement in accuracy.

Contextual Polarity There are certain words which are commonly used to depict positive sentiment, and some, which are used for negative sentiment. Examples of words like these are the words "good" and "excellent", which are mostly associated with positive sentiment. The prior polarity of these words is most often positive. There are also examples of words which commonly depict negative sentiment, like "bad" and "absurd". Then, there are those words which could depict opposite sentiments depending on the domain, or on the context in which they are used. The polarity of the same emotive word/opinion word may vary in

different domains [115][68]. An example of one of such words is the word "unpredictable", which in a movie review would mostly depict a positive sentiment towards the review, but in an automotive review for instance, would mostly be used to depict a negative sentiment. An example of this is : "the steering wheel is unpredictable". Another word with such characteristics is the word "long". "Long" could be used to depict a positive sentiment, for instance, in a product review : "the camera has a long battery-life". However, when used in a movie review, it could depict a negative sentiment, like : "this movie is very long". Due to this varying polarity which results from domain specificity of various sentiment words, an approach which incorporates contextual polarity information is desirable.

Fei et al [28] state that based on the transitivity of relations, different words in a sentence may be related. Their method uses transitivity to find words which are related to other words in sentences, as depicted by the dependency tree. They go further to explain that there are three instances about the relation for opinion mining, based on the relations defined by the dependency grammars. In these instances, the subjective words may be in the children of the aspect, the aspect and the subjective words may be in the different children of the same ancestor, and the subjective words may be in the ancestors of the aspect. Apart from these three, they also point out that there are dependency relations between subjective words and their modifiers. These modifiers affect the strength or the orientation of the sentiment. These words are referred to as words S-words. When mining the relations from a dependency tree, they first search for which sentences may express subjective opinion, by checking which ones contain subjective words. If a subjective word does appear, they search for the related aspect and S-words, like searching in a multi-way tree. They report results both from using a search depth of 4, and from not limiting the search depth at all. It is observed that not limiting the search space at all leads to a number of redundant relations being mined, which affect the level of accuracy adversely.

Also using dependency trees, [66] selected only certain dependency relations such as SV (subject-verb),VO (verb-object) and AN (Adjective-noun) relations, extracted from each document using the MINIPAR dependency parser. They selected the best 5000 features

from these, based on their WLLR and then carried out classification with unigrams, bigrams, trigrams and the selected dependencies. They however report that this did not yield an improvement in accuracy over the result obtained through the use of the n-grams only. Upon carrying out another test, they found that not including the bigrams and trigrams, but only using the dependency relations with unigrams, led to an improvement in the accuracy. This was considered unexpected, given that n-grams do not capture non-local dependencies. Further investigation of the test documents led to the discovery that the dependency relations performed badly due to a parser error. They had used MINIPAR, and this returns dependency relations in which the verb inflections are removed. Thus, the over generalisation which occurs from these stemmed relations renders the dependency information useless for polarity classification. Their conclusion was that additional experiments were needed to determine role of dependencies when stemming is disable in MINIPAR.

Li et al [52] proposed a dependency tree-based sentence-level sentiment classification approach. They added more information to the dependency tree, and then applied an algorithm to prune the dependency tree in order to reduce noise. The corpus is preprocessed, and the Stanford dependency parser is utilized for the extraction of dependency relations. A dependency tree is constructed, and its nodes are further expanded to incorporate more information. A label SemLex is tagged true for sentiment words, and false for others. Then, the part-of-speech feature is added to each node. Finally, each node besides the root node is appended with the dependency relation between the node and its parent. The dependency tree is again pruned after this, by first filtering all words that do not have a direct relation with a sentiment word, and then sorting all remaining words basedon their frequency. Only words with high frequency are retained. Eight dependency relations which are considered to benefit sentiment classification, and whose frequency is high, are selected. Some dependency relations with high frequency that do not contribute to the classification are dropped. An example of this is "det" - determiner. The composite kernel - comprising of the tree kernel(for capturing the structured information), and the basic kernel (for capturing the flat information which cannot be captured by the tree kernel) is applied for classification. The flat features

in this case are the unigrams.

Another approach by which dependency information is utilized for sentiment classification is through the use of subgraphs. A subgraph is a section of a dependency graph which may be extracted for classification purposes. It consists of a number of edges and nodes, but is not the full graph. The idea behind the use of the subgraph is to extract useful relations, as using the full graph could introduce errors due to noise.

SVM classifiers working with features based on extracted subgraphs have been reported to perform better than traditional systems based on the unigram model, in some instances [28]. Matsumoto et al [55] proposed an approach using frequent sub-sequences and sub-tree mining, and reported that these features outperformed the bag-of-words approach for sentiment classification. Subgraphs were also used in [6], where they were extracted from annotation graphs to create more complex features to be used for classification in addition to unigrams.

For the generalization of the features obtained from dependency trees, wildcarding is applied. Wildcarding is a technique in which some words in a dependency tree or subtree are replaced by a generic node, which can match any term. Wildcards have been used in a number of ways, including being used to replace the feature term in a relation, while not replacing the sentiment words. This is in order to extract relations which may vary only in the object term for instance, and have the same emotion term [70]. In a similar vein, [6] extract features from annotated graphs, where wildcards are used to replace polar words, and also append polarities to the polar words in the subgraphs.

2.5 Levels of Granularity in Sentiment Analysis

Sentiment analysis is performed at three basic levels; the document level, the sentence level and the word/phrase/aspect level. Aspect here, refers to the components of an object about which an opinion is being expressed. As will be explained later, these components are sometimes viewed as a word in the document set, or a phrase. We expantiate on these levels

below.

We also present in Table 2.2 an overview of some of the most popular and well cited works on sentiment classification of documents, using the movie reviews dataset, with the accuracy scores obtained. These results are obtained from different levels of granularity.

Table 2.2: General Overview of approaches

Author	Objectives	Method	Dataset	Accuracy
Turney (2002)[97]	Document sentiment.	PMI-IR	Auto,bank,movie and travel reviews	65.8-84.0
Pang et al (2002)[74]	Document sentiment.	NB, SVM,ME	Movie reviews	77.0-82.9
Pang and Lee (2005)[73]	Doument sentiment.	SVM, regression Metric labelling	Movie reviews	66.3
Gammon (2004)[30]	Document sentiment.	SVM	Customer feedback	77.5%
Dave et al (2003)[23]	Document sentiment.	sNB,SVM,ME	Product reviews	88.9
Pang and lee (2004)[72]	Document sentiment.	NB,SVM	Movie reviews	86.4-87.2
Parkhe(2014)[77]	Aspect-based	NB	Movie reviews	79.4
Whitelaw et al(2005)[99]	Document sentiment.	Appraisal groups	Movie reviews	90.2
Aue and Gammon (2005)[8]	Document sentiment.	SVM	Movie reviews	90.2
Pak and Paroubek (2011) [70]	Document sentiment.	SVM and Subgraphs	Movie reviews	85.1

2.5.1 Document Level

Sentiment analysis at the document level aims at classifying the sentiments of the opinions of the document as a whole. A holistic view of the document is held, under the assumption that the document focuses on a single object and contains opinions which are from a single opinion holder [80]. The opinion document is considered as a basic information unit which provides information related to one specific entity [98]. The document is essentially considered as a bag of words, with no attention paid to its structure.

An example of this could be a product review which contains opinions about different aspects of the product. The document level sentiment classification determines whether the review expresses an overall positive or negative opinion about the product, and does not analyze the various opinions which may be expressed about individual parts of the product.

First converting the documents into vectors, [73][72] and [4] proceed to utilize supervised learning algorithms which included SVM, maximum entropy, Nave Bayes, and ADTree to categorize them into their different sentiment categories. The documents were treated as a whole unit. In the same work, [4] also utilized a lexicon-based approach using a sentiment lexicon. Each word is compared against the dictionary and the extracted polarity value of the word is added to the total polarity score of the document if the word is present in the dictionary. The polarity of the document is then considered positive if the total positive score of the words is positive, otherwise, it is considered negative.

Turney's widely cited work [97] is another example of a document level sentiment classification method. He determined the semantic orientation of documents using an unsupervised approach of obtaining word polarity through co-occurrence with the words "Excellent" and "bad". Sharma et al [88] also carried out document level classification, in which they determine the polarity of movie reviews at the document level, through an unsupervised dictionary based technique [49].

2.5.2 Sentence Level

Sentences are considered to be short documents. The first step in sentence level sentiment analysis is usually determining if the sentence is a subjective sentence, or an objective sentence [31][98]. There is also the assumption that a sentence expresses a single opinion from a single opinion holder, though this in reality only holds for simple sentences [53]. A subjective sentence is one which expresses an opinion, while objective sentences are more statements of fact, hence hold no opinion. Just like document level sentiment analysis, sentence level sentiment analysis determines if the polarity expressed by each subjective sentence is negative, positive or neutral [103]. Though opinionated sentences (sentences which definitely

express an opinion) are subjective sentences, not all subjective sentences are opinionated sentences [53]. This task of determining whether a sentence is subjective or objective is called Subjectivity Analysis.

Yu and Hazivassiloglou in [108] tried to classify subjective sentences and determine their opinion orientations. In order to identify the opinion sentences, they applied supervised learning. For sentiment classification of each identified subjective sentence, they used a similar method to that used in [97], but with more seed words than just the two which were "excellent" and "poor", using log-likelihood as the score function. Kim and Hovy in [46] used a simplistic approach of summing up opinion words in a sentence to determine the sentiment polarity of the sentence.

Liu et al [54] proposed a rule-based approach to classify a user's opinion in terms of various product features. The sentence structure they considered was that of a simple sentence.

There has also been some work in the area of analysing a sentence and probably introducing some contextual information into the sentence level classification process. Identifying objective and subjective sentences in reviews and blog comments, [44] used a rule based method, for identifying subjective sentences, and then determined their semantic scores from SentiWordNet. The final weight of each individual sentence is calculated, taking into account the whole sentence structure, contextual information and word sense disambiguation. In carrying out the word sense disambiguation (WSD), they first performed part of speech tagging of the words in the sentence, and extracted the semantic scores from SentiWordNet based on the part of speech of the word.

Using dependency trees to obtain more structured features from sentences has also been implemented by [52],[55] and [63]. This is in most cases a way of improving the sentiment classification accuracy in document level classification.

2.5.3 Aspect Level

Aspect level sentiment analysis, sometimes referred to as feature level sentiment analysis is a more fine-grained approach to sentiment analysis. Aspects are the important features,

which may be a word or a phrase, which have been rated by reviewers [16], for instance, in a product or movie review. An example of an aspect when referring to product reviews, like a mobile phone, may be the "Battery Life", or "size". Different from the other two approaches, sentence and document level, aspect level is based on the idea that an opinion consists of a sentiment, which may be positive or negative, and the target of this opinion [102], which is the aspect.

Aspect level sentiment analysis has been used to evaluate sentiments in social media, for example, Twitter. In [86], sentiment detection was carried out at both tweet-level and entity-level. The entity-level analysis was concerned with detecting sentiment towards a particular entity or topic in the tweet, and the tweet-level analysis detected the overall sentiment of the tweet. In the same vein, [104] assigned sentiment polarities to new words in tweets, using statistical, as well as pattern information.

In order to implement aspect level sentiment analysis, the aspects or potential aspects have to be identified. Terminology extraction has been the most basic method used, which has still given acceptable results. In [38], Hu and Liu extracted noun phrases as aspects, and observed that their approach extracted too many irrelevant terms. In their dependency based method, [28] mine dependency relations using a multi-way tree search. They first identify the subjective words, and then go on to trace the aspects. Their potential opinion targets are popular terms from the Cornell movie review corpus, which is the same corpus used in our work, as well as from an online film glossary. They report having a number of redundant relations extracted in their method. Association rule mining, which was also used in [38] is used in [5] to determine the most characteristic aspects of a product, in a product review. Zha et al in [111] identified aspects by extracting noun terms in reviews. They retained the most frequent noun terms as aspects. Jianzing Yu et al [109] proposed a method for identifying important aspects in customer reviews based on observations that these aspects were those which were commented on the most, and hence the ones with the most influence on the overall polarity of the review. Tun Thura Thet et al [94] proposed an approach to detect the sentiment orientation, as well as sentiment strength towards a certain

aspect in a review. Domain specific and generic opinion lexicons were used to score words, and inter-word dependencies were extracted using the dependency tree. The word score was then propagated over the entire document. In an attempt to determine which aspects in a review are directly responsible, or mostly responsible for the sentiment orientation of the review, [77] analyzed movie reviews and assigned scores called driving factors to various aspects, and determined that assigning high driving factors to - "movie", "acting", and "plot" aspects of a movie review, led to the acheivement of the highest accuracy.

2.6 Summary

This chapter elaborates on the related work that has been carried out in the area of sentiment classification. We also present the different existing approaches and draw similarities as well as differences between them.

We discuss the various levels of granularity which sentiment analysis has been carried out in, that is, the document level, sentence level, and aspect level. We also discuss the areas of application of this study area.

We present an overview of some of the accuracy scores which have been reported in some of the most cited literature in the area, with preference to the movie reviews dataset. This is to offer an insight into the state-of-the-art of sentiment classification.

Chapter 3

Theoretical Framework for Polarity Classification

The approach we propose in this work is a novel hybrid approach to sentiment classification. Our aim is to develop an approach that combines lexicon-based approaches with learning based approaches. The motivation behind this is to obtain a framework for sentiment classification which achieves a high accuracy, akin to pure learning-based approaches, while maintaining the structured and explainable results obtained from lexical based approaches.

Our hybrid approach is aspect focused, as the aim of the approach is to identify aspects about which sentiment has been expressed, and extract the relationships that exist between these aspects and the sentiments expressed about them. These relationships will ultimately be incorporated in the feature set utilized for the classification of the documents.

The approach we are proposing aims to offer more structured results, and an aspect focused explanation and justification of the results that we obtain. Using the dependency tree, we exploit long range relations as features, and incorporating these with ngrams, we build a feature set which we utilize for classification in a machine learning classifier. This incorporation of lexical features in a machine learning classifier gives rise to the hybrid approach.

We propose to do two things; develop a novel aspect focused approach, which classi-

fies documents according to their sentiment class, by taking into account the sentiments expressed towards each aspect, and which also tackles domain specificity.

We propose to utilize the novel approach of transitivity in extracting features from a dependency tree. These transitive relations will ensure that we do not only use the lexicalized dependencies produced by the dependency parser, but also generate dependencies with a longer range, which increases the chances of extracting aspects with their associated sentiments as a relation pair. The aim of this is to achieve a classification which does not only consider the overall sentiment polarity of a document, while missing the intrinsic sentiments associated to the constituent aspects within the document.

To tackle the domain dependency of sentiment bearing terms, we introduce composite features and a domain specific lexicon into our hybrid approach. These features are designed to incorporate the contextual polarity information of a text in the classification process.

Multi-domain adaptability is an important focus of our work, hence to achieve this, we steer clear of incorporating knowledge sources which are too domain specific, like online resources, or online groups of identified aspects. We hope to obtain an approach which would still be implementable if certain specific resources are not available at the time. This portability is the reason we refer to it as a framework, because it can easily be applied to datasets from other domains, besides those tested in this work.

We implement the lexicon-based and machine learning approaches in this work, to explore the effects of various features on these approaches, and to investigate the potential effect of sentence position on sentiment polarity determination. We explore what these approaches encompass, and establish the importance of lexicons for classification. We go on to demonstrate the efficiency of machine learning approaches, and the factors that determine this efficiency. We propose to show how the architecture of our novel hybrid approach brings these two approaches together, to produce a polarity classification approach that classifies documents using a classifier trained on knowledge rich sources, while retaining the accuracy of a pure-learning based classifier.

We will compare the performances of these two approaches against our novel hybrid

approach, to establish the necessity of such a combined approach.

This Chapter presents a theoretical analysis of the proposed approach, and the focus is summarised.

- To examine the effect of incorporating more features derived from linguistic knowledges sources in a learning based classifier.
- To address the issue of domain specificity of sentiment bearing words.
- To investigate the suitability of certain sections of text in adequately capturing the sentiment of the entire text.
- To carry out document level classification, taking into consideration the linguistic constructs present in the constituent sentences.
- To implement an aspect-focused document classification.

In tackling the above issues, we hope to be able to address our research questions listed in Chapter One of this work. We discuss the general architecture of the proposed approach, aspect identification, features and composite features, transitive relations and domain specific lexicon generation in this Chapter.

Additionally, we introduce each of the processes that make up our approach, pointing out the techniques which have already been tried in the literature, which we include in our approach, the techniques we have adapted, as well as the techniques which we are proposing, which to the best of our knowledge, have not been previously tried in the literature.

We first give a detailed analysis of the different components which make up our novel hybrid approach, and then go on to elaborate on the applied methodology of the proposed hybrid approach, and the pure learning and pure lexicon-based approaches which are implemented in Chapter Four and Five of this work, respectively.

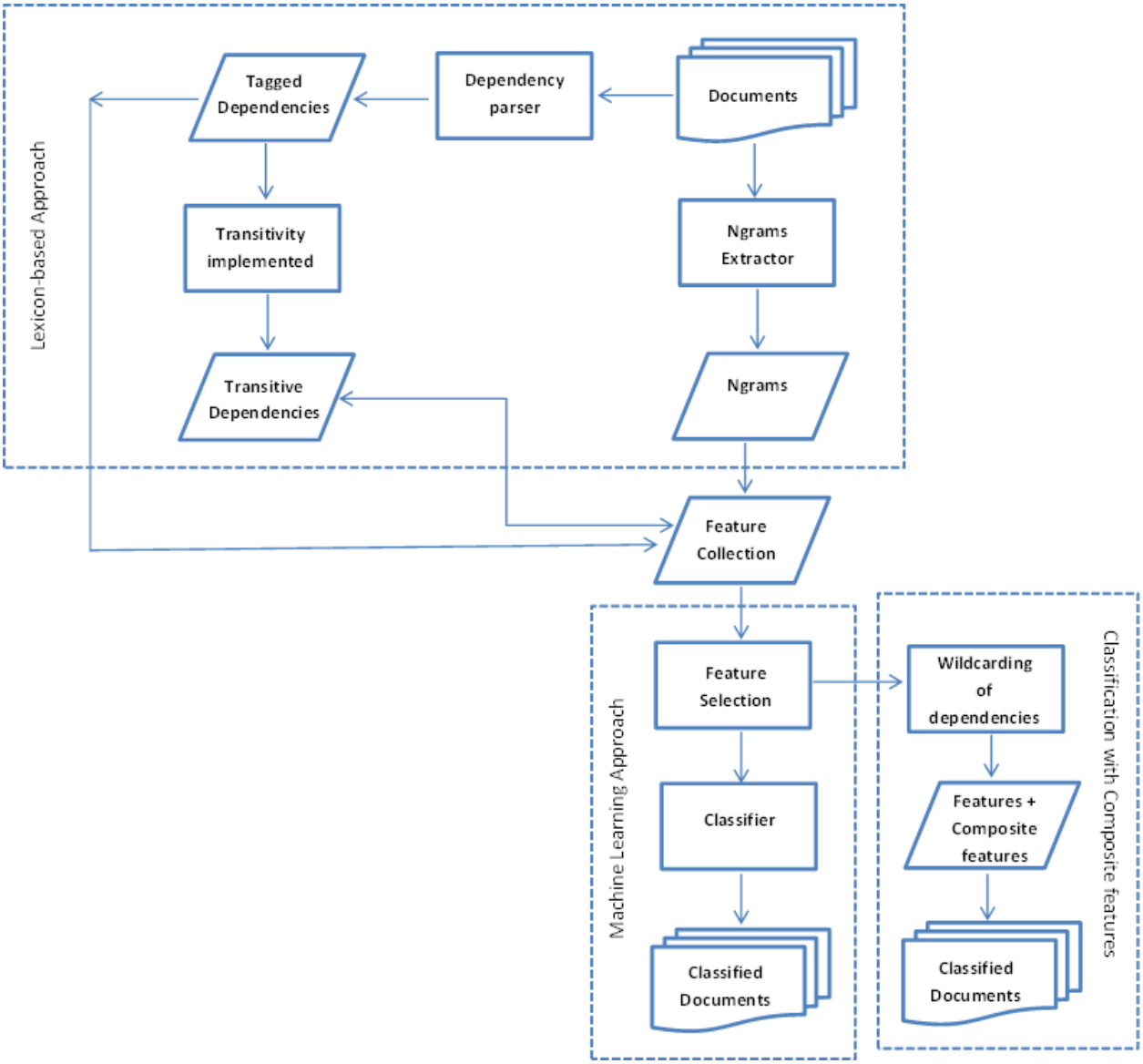

Figure 3.1: General Architecture of Hybrid Approach Classification

3.1 General Overview of the novel hybrid polarity classification approach

The general architecture of our proposed approach is given in Figure 3.1. It depicts the flow of processes from lexicon-based processes to the machine learning classifier. We elaborate more on this in Section 3.3.3, where we give the methodology of the novel hybrid approach.

3.2 Components of the Novel Hybrid Approach

In this section, we provide a description of the various components that make up the proposed novel hybrid approach. Components such as the features utilized for classification, the process of identification of the aspects, the feature selection process, as well as the creation of the domain specific lexicon.

3.2.1 Features

The features which are used in this approach are ngrams, which include unigrams, bigrams, trigrams, and dependency relations, which are obtained from the parsing of a dependency tree of each sentence. Lower order/ short range dependencies are captured through the use of ngrams [87], long range dependencies which are non local in a sentence are captured by dependency relations [66][61].

As part of our motivation to examine if incorporating further linguistic sources on a learning based classifier in terms of sentiment classification would improve classification accuracy, we derive a number of other features which we also examine. These features include composite features, derived from dependency relations, and also transitive relations, which are the novel features we derive from the lexicalized dependency relations as well.

We discuss more on composite features and transitive relations in the next subsections.

3.2.2 Composite Features

To incorporate semantic features, such as contextual sentiment polarity scores, and syntactic features, which include ngrams and part of speech tags and dependency relations in our approach, we made use of features which we refer to composite features.

Before outlining these feature, it is imperative that we discuss the concept of wildcarding, which we will employ in the development of these composite features.

3.2.2.1 Wildcarding

The means through which we introduce these features to our feature set is by introducing them into our dependency tree. We perform this through a process referred to as 'wildcarding'. Pak and Paroubek [70] define a 'wildcard node' as a node which can match any word. We thus introduce these features through the use of wildcard nodes, which replace some dependency tree nodes.

We demonstrate a form of wilcarding in Figure 3.2 using one of the sentences and dependency parse tree from out dataset. Figure (a) and (b) show the sample graphs of before and after wildcarding. The goal is to wildcard the nouns.

Let's assume we have the sentence : "The lack of gusto is the movie's major stumbling block"

The dependency relations extracted for the sentence are shown below.

```
det(lack/NN-2, the/DT-1)
nsubj(block/NN-11, lack/NN-2)
prep_of(lack/NN-2, gusto/NN-4)
cop(block/NN-11, is/VBZ-5)
det(movie/NN-7, the/DT-6)
poss(block/NN-11, movie/NN-7)
amod(block/NN-11, major/JJ-9)
amod(block/NN-11, stumbling/JJ-10)
root(ROOT-0, block/NN-11)
```

The extracted dependencies are a sample, showing the structure of the dependency relations extracted with the Stanford Dependency parser [1]. The Stanford typed dependencies representation was designed to provide a simple description of the grammatical relation in a sentence. The relation is written as *abbreviated_relation_name(head,dependent)*. The head or governor and the dependent are words in the sentence to which a number which indicates

[1] http://nlp.stanford.edu/software/lex-parser.shtml

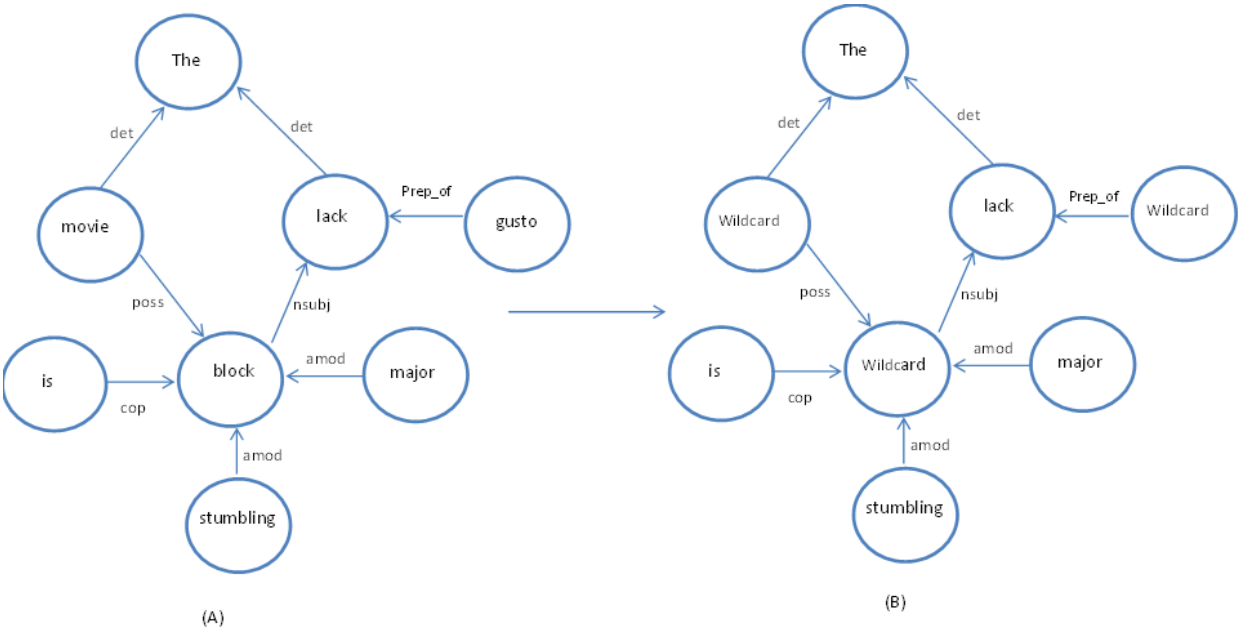

Figure 3.2: Dependency graph with wildcarding

the position of the word in the sentence is appended. We do not consider this position index in our dependency relations.

The extracted dependencies can be explained as:

```
nsubj    => nominal subject
cop      => copula
det      => determiner
amod     => adjectival modifier
root     => root
prep_of  => preposition of
vmod     => verbial modifier
dobj     => direct object
appos    => appositional modifier
```

The parser also tags the part-of-speech of words using the Penntree bank Project [1]part-of-speech tags. The tags displayed in the sample tree translate as follows:

```
NN   => Noun, singular or mass
```

[1]https://www.ling.upenn.edu/courses/Fall_2003/ling001/penn_treebank_pos.html

```
PRP  => Personal Pronoun

VBZ  => verb, 3rd person singular present

DT   => Determiner

JJ   => Adjective
```

The concept of wildcarding has been used in [70] where sentence subjects and objects are replaced by wildcard nodes. Verbs and adjectives are not wildcarded because they refer to these as sentiment words. Subgraphs were then extracted and classified.

Ng et al [66] create new features by replacing the polar adjectives in their extracted bigrams with their polarity label. Polar adjectives here refers to adjectives that have a semantic orientation of positive or negative. Neutral adjectives are ignored. These new features are added to their list of bigrams. Composite back-off features is the term used by [40] to refer to a similar technique, in which they "back-off" the head word and the modifier term of a dependency relation to their part-of-speech alternately, in order to create new features.

In our hybrid approach, we first identify the opinion terms in the corpus, and these are the adjectives, adverbs and verbs in our work, and replace them with wildcards, and repeat a number of variations for this. We do not limit our opinion terms to adjectives, but incorporate other word types as we hypothesize that adjectives are not the only indicators of sentiment.

Our approach differs from the ones listed here, in that, [40] focus on subjectivity detection at the sentence level, and consider relations with the "amod" relation only. The "amod" relation is the adjectival modifier. They implement their approach on product reviews. We focus on document level sentiment classification, and we consider all relations (except the determiner (det) relation) not just "amod" relations.

Our features also include unigrams, bigrams and trigrams, in addition to dependency relations and transitive dependencies.

Pak and Paroubek [70] used subgraphs with combined dependency nodes and wildcarded the nouns in their work. We are instead proposing to wildcard the sentiment words, and

we are using a combined feature set of ngrams , dependency relations and also transitive relations.

The variations we aim to incorporate in our wildcarded features will be part replacement of words in a dependency pair with their part of speech, part replacement with their polarity, and also a full replacement of a pair with their part of speech tag, as well as the polarity.

These features are described below.

Part of speech - Word pair In this form of wildcarding, we will replace one member of a dependency relation with its part-of-speech tag, to create a more generalized and composite feature. We will do this alternately, between the head term and the dependent term in the relation pair.

Hence, given a pair such as : $rel(head/POS, dependent/POS)$ where the "POS" is the part of speech as determined by the parser, we do not consider the relation, but create another feature, which will be considered by the classifier as a pair of $(head, POS)$. This order is swapped to create another feature $(POS, dependent)$, and these features are added as part of the feature set.

Our motivation of creating these features is to incorporate a basic form of word sense disambiguation and context in our classification process, and to introduce syntactic features into our feature set.

Word-Polarity pair Another composite feature introduced into our feature set is the wildcarded pair created by replacing one of the words in the pair with its polarity. Hence, a pair $rel(head/POS, dependent/POS)$, will give rise to two seperate features: $(head, POL)$ and $(POL, dependent)$. These features are used in two seperate runs. The motivation behind this is to incorporate the prior polarity of a term in the set of features that we use for classification. Incorporating this polarity score also enables us include contextual polarity in our features, because these relations have been extracted from the dependency tree. According to [94], the dependency tree acts as a provider for contextual sentiment.

Part of Speech-Polarity pair This composite feature is created by a full replacement of the constituent terms in the dependency relation pair with their part of speech, as well as their polarity, alternately. Given a pair $rel(head/POS, dependent/POS)$, the features we create are $(POS(head), POL(dependent))$ and $(POL(head), POS(dependent))$. POS(head) signifies the part-of-speech of the head word, and the POL(dependent) signifies the polarity of the dependent word.

The motivation behind these features is to create more composite features and generalized features which introduce deeper linguistic features and a more fine-grained analysis into our sentiment classification system. Joshi and Penstein-Rose [40] reported improvements in classification of product reviews through wildcarding, and attribute this to a better generalization of features brought about by this altering of the syntactic dependency pairs. Our intuition is that including these features introduces more linguistic knowledge to the learning based classifier, and provides a means for us to assess the effect these features have on the classification process. Our hypothesis is that this brings about a reduction in the sparsity of features, and hence leads to a better classification accuracy.

Appending Polarities Another set of features which we add to our feature set is created by appending a polarity to the part of speech of each of the constituent terms in our dependency relation pair, alternately. Hence, given a relation such as $rel(head/POS, dependent/POS)$ again, we create two composite features which will be added to the feature set in separate runs. The features are $(POL\_POS, word)$ and $(word, POL\_POS)$.

We also add another feature which is (POL_POS, POS) and (POS, POL_POS). We believe that this feature would capture more context than having a feature like (POS, POS), used by [30] for their dependency relations, and by [56] on bigrams and trigrams.

3.2.3 Transitive Relations

The technique which we employ to extract deeper relations from the dependency tree is transitivity. We employed this technique to enable us extract relations between aspects or

opinion targets and other words in sentences, which might otherwise not be captured by ngrams or lexicalized dependency relations. Lexicalized dependency relations here refers to the dependency relations obtained from the dependency parser.

According to [28], based on the transitivity of relations, different words in a sentence can be related. They demonstrated this by extracting composite relation pairs and comparing the sentiment expressed in these pairs against those expressed in the sentence from which they had been extracted. For example, taking a sentence such as : " This film is genuinely funny", they extracted the pair $film, genuinelyfunny$. They did not carry out sentiment classification of the reviews.

Some form of transitivity is performed in [61], who also perform aspect specific sentiment analysis on product reviews. They perform clustering on the dependency graph to extract only those opinion expressions that are most closely related to a certain feature. They assign words to various clusters based on the number of edges separating them, and carry out a lot of product aspect merging. It is however not clear how they determine which dependency relations to merge, but they report improvements coming about from extracting transitive relations for their classification.

We adopt the technique of transitivity in our work, because it is our intuition that this would contribute to our goal of growing our feature set with more aspect-focused or aspect-influenced features.

We carry out two variations of transitive relations extraction. In the first variation, we extract all the relations in the dependency graph around a potential aspect, to the depth of '4', and in the second variation, we extract only the relations which include a potential aspect and a sentiment term, which might be an adverb, an adjective, or a verb. We do this by means of a multi-way tree search approach. The decision to choose '4' as our search depth, is based on the work of [28], who showed that choosing a search depth of '4' gave much better results than having an unlimited search depth.It is also our intuition that having a limited search depth would help cope with sentences which might be expressing two different sentiments on two different aspects, in the same sentence. The search depth would enable

```
Data: Document d and list of selected aspects (L)
Result : Transitive Dependencies
For Sentence S in d
  if Aspect a in Sentence (S)
    if Aspect a in L
      Extract Transitive Relations around a to a depth of '4'
      Add extracted Relations to list of Dependencies
    end
  end
end
```

Figure 3.3: Algorithm for extracting Transitive Dependencies

use extract relations which are closer to a certain aspect, and as such avoid extracting the wrong sentiment terms.

We demonstrate this process in the algorithm in Figure 3.3. The motivation behind our extracting all the transitive relations is our intuition that having these included will add context information in the classification process. We believe these additional relations will also extract relation pairs which will be of the opinion word and target (aspect), a relation which might otherwise be missed in regular lexicalized dependencies.

We make an adjustment to the process by keeping only the transitive relations that have an aspect in a pair with either an adjective, an adverb, or a verb. There has been some research that have extracted patterns which might involve pairing between adjectives' adverbs and nouns, like [97], who extracted some phrase patterns like RB/RBR/RBS+JJ+NN/NNS; JJ+NN/NNS+ Anything; RB+VB+ Anything. These patterns involved adjectives, adverbs and verbs with nouns. Nasukawa [64] also used some patterns, such as :$Verb + Obj$ and $JJ + Obj$, that is, verb and objects, and adjectives and objects.

As a motivation for using transitivity, [28] described these sort of pattern extraction methods to be based on fixed position of words, and as such, having the potential to miss out on important relations.

We share this intuition, and our approach differs to the above mentioned approaches in that our relation pairs are extracted through transitivity. We hypothesize that they would provide a broader coverage of the relations within the sentence. We also believe that

```
Data: Document d and list of selected aspects (L)
Result : List of Transitive Dependencies (with an emotive word in pair) LT
For Sentence S in d
  if Aspect a in Sentence (S) then
    if Aspect a in L
       Extract Transitive Relations T around a to a depth of '4'
       Store in Temp
    end
  end
  For all T = (a,b) in Temp
     if POS (a) or POS(b) is an adverb or adjective or verb
           (POS refers to Part Of Speech)
        Add extracted Relations to list of Dependencies
        Add T to LT
    end
   end
 end
```

Figure 3.4: Algorithm for extracting Transitive Relations with Emotive Words

extracting these relations which involve an emotive word would contribute to our focus on subjective text, and reduce the amount of objective texts that is involved in the classification process, which may be potentially misleading [72].

We depict this process in the algorithm shown in Figure 3.4.

To demonstrate this, we use this sentence from the Cornell Movie reviews dataset[1] used in this work :"it's a terrible mess of a movie starring a terrible mess of a man , mr . hugh grant , a huge dork". This sentence is from a negative review in our corpus. The dependency parse tree[2] of this sentence is given below:

```
nsubj(mess/NN-5, it/PRP-1)

cop(mess/NN-5, 's/VBZ-2)

det(mess/NN-5, a/DT-3)

amod(mess/NN-5, terrible/JJ-4)

root(ROOT-0, mess/NN-5)

det(movie/NN-8, a/DT-7)
```

[1] urlhttp://www.cs.cornell.edu/people/pabo/movie-review-data/

[2] Extracted using the Stanford Dependency Parser

```
prep_of(mess/NN-5, movie/NN-8)
vmod(movie/NN-8, starring/VBG-9)
det(mess/NN-12, a/DT-10)
amod(mess/NN-12, terrible/JJ-11)
dobj(starring/VBG-9, mess/NN-12)
det(man/NN-15, a/DT-14)
prep_of(mess/NN-12, man/NN-15)
appos(man/NN-15, mr/NN-17)
```

The dependency tags produced, are basically the same as the tree produced earlier, except for the VBG part of speech tag. This tag translates into *verb, gerund or present participle.*

From the parse tree, we note that there are two popular nouns in the sentence, "mess" and "movie". Given that this is a movie reviews dataset, we hypothesize that the word "movie" could be chosen as a potential aspect on which opinions are expressed on. We can see from the parse tree that the adjective "terrible" for instance, does not have a direct a relation with "movie", but it does in fact reflect the sentiment of this author towards the movie. We hypothesize that with transitivity, we would be able to extract a relation movie, terrible, which would aid classification accuracy, and also contribute to our composite features set.

3.2.4 Feature Selection

Learning-based/machine learning methods have the advantage of having a high classification accuracy, but this accuracy can only be achieved with a representative collection of the labelled training texts, and through careful selection of features [11]. Hence, not all extracted features are required in order to obtain a good classification accuracy, as some features may end up harming the classification process, than contributing to it.

Ohana and Tierney report obtaining their best result in [67] when a feature refinement step was added in their classification using features from SentiWordNet with the Support

Vector Machine classifier. Pang et al [74] also observe that there was a possibility that applying feature selection algorithms with their SVM classifier could improve performance.

We utilize feature selection in our approach, because we believe that a careful selection of relevant features aids the learning based classifier in carrying out a more accurate classification. We apply a scoring scheme to our ngrams and dependency relations and rank them in decreasing order of these scores, and then select a number of the top ranking features to be used as features for the learning based classifier.

3.2.5 Weighting Scheme

In a vein to ensure that adequate importance is given to constituent words in the document collection, it is imperative to choose a representative weighting scheme for the features. There are varying reports on the efficiency of using frequency as the feature for classification, with some reports stating that term presence, gives a better classification accuracy than term frequency does, like [74].IR methods have long regarded the rare occurrence of terms in document collections as a very important feature, and effective IR models of today, either implicitly or explicitly accommodate this feature as an Inverse Document Frequency (IDF) heuristic, and similarly, the prominence of a term recognized by the frequency of the term in its local context, is formulated as Term Frequency (TF).

In this work, we use the term presence weighting scheme. We hypothesize that the occurrence of a distinctive word would lead to better classification than words which occur very frequently in the dataset. For this reason, we choose term presence over frequency for our approach.

3.2.6 Aspect Identification

Aspects are the features of an entity, about which an opinion is expressed. We refer to the opinion targets in this work as aspects.

Identification of the emotional attitude to some object in a body of text has been referred

to by [11] as the main problem of sentiment analysis. [61] supports the notion of aspect specific opinion classification by expressing that people tend to have mixed opinions about various aspects, where some are positive, while others are negative. Hence, the overall opinion matters less than feature specific opinion. Aspect-based sentiment analysis is also seen as an approach which performs more in-depth sentiment analysis of review texts [94].

These form part of our motivation for extracting features based on their relations with identified aspects. There have been research on the area of aspect-based sentiment classification (See Section 2.5.3). We highlight those whose approach is most similar to ours, to differentiate our work from what already obtains in the literature.

Nouns and noun phrases have been selected as aspects in documents, during sentiment classification [59][38]. In Mudinas et al[59], a combined approach between learning and lexicon approaches was implemented, and they report improvements in their approach over pure learning based and pure lexicon based approaches. While [38] mine associative relations around identified aspects in product reviews, [28] mine dependency relations by first identifying subjective words and then tracing back from them to the aspects. They report having a number of redundant relations extracted in their method. Our intuition is that this was due to not carrying out appropriate feature selection, and their relation extraction method, which starts from the subjective words and traces back to aspects.

In our approach, we extract the most frequent nouns in the corpus as potential aspects. As was implemented in [59], we set a support level of '5', meaning that we drop any potential aspects that do not appear up to at least '5' times in the corpus. Though we implement the multi-way tree search, similar to [28], we modify our approach to trace the relations around identified aspects, and not from subjective words. We run two tests on extracted relations, one in which all relations are tested, and the other, in which relations with sentiment bearing terms only are tested, to account for subjective text only classification.

Figure 3.5 depicts the process of aspect/opinion targets selection.

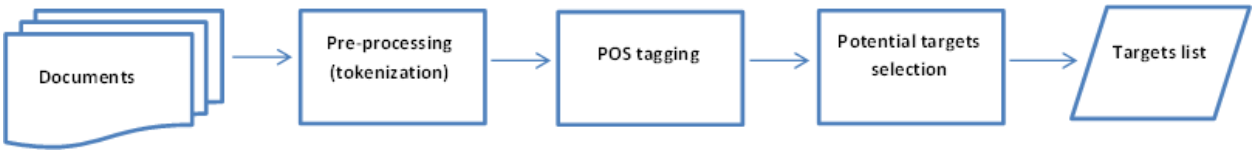

Figure 3.5: Steps in aspect/target identification

3.2.7 Domain Specificity

Sentiment bearing words or emotive words as we have been referring to them in this work, have a tendency to be domain specific. This means that some words may be used to express opposite sentiments in different domains [75]. Previous studies have indicated that the sentiments of several texts are domain and context dependent, and hence, determining the context the word is being discussed is important in determining the related sentiment [82].

The polarity of a word in the absence of context is referred to as the prior polarity. This is usually the polarity of words found in lexicons, like SentiWordNet and WordNet.

Due to the challenges which come about as a result of the domain specificity of the polarity of sentiment bearing terms, determining and incorporating the contextual sentiment of words in polarity classification is an important task to implement in sentiment analysis (See 2.4.3).

One way of tackling this domain specificity issue is through the use of domain specific lexicons. Rastogi et al in [82] have stated that incorporating the information which is incorporated in sentiment lexicons, mainly domain specific lexicons, can lead to a drastic improvement in the accuracy of sentiment analysis.

In our approach, we adopt the technique of creating a domain specific lexicon, which is hoped to help us determine the prior polarity of terms with respect to the manner in which they are used in the focus domain. We elaborate on how we create this lexicon in the following subsections. By incorporating the sentiment polarity derived from the domain specific lexicon in relation pairs extracted from dependency tree parsing and transitive dependencies, we are able to incorporate the contextual polarity of these terms in our classification process.

3.2.7.1 Seed Words selection

Seed words are a small set of words which have strong negative or positive associations. Examples of seed words are words like "excellent" which has very strong associations with the positive sentiment, and "appalling", which has very strong associations with the negative sentiment. Basically, if an adjective is close in terms of synonymy to a positive word, or close in terms of antonymy to a negative word, then the adjective is classified as positive, and vice versa [92].

In lieu of the drawbacks of the manual and the automatic approaches to seed words selection elaborated on in Section 2.4.1.3 of this work, we have chosen to adopt the semi-automatic approach to emotive seed words selection. We are not aware of this approach being used or reported in the literature for document level sentiment classification. The approach closest to ours is that which was used by [94] who semi-automatically analysed reviews from a seperate training data than that which was utilized in their work, and extracted important opinion words strongly associated with the positive and negative classes and further went on to perform manual examination of the extracted words, before using them for their lexical approach to clause level sentiment classification.

In our approach, we utilize a scoring scheme which has the ability to separate two classes and obtain informative features for both, then semi-automatically extract the top 100 negative and top 100 positive emotive words, which could be adjectives, adverbs and verbs from our training data. We use a bootstrapping technique which will be discussed in 3.2.7.3, to grow our opinion list with synonyms from a sentiment analysis lexicon.

3.2.7.2 Domain Specific Lexicon Generation

A number of techniques have been adopted in the literature for the creation of domain specific lexicons. Automatic methods involve creation through association, where the semantic orientation for each word is calculated based on the frequency of the co-occurrence of the target word with selected seed words [97][36], semi-automatic approaches include those methods which make use of resources like WordNet [115] and manual methods, which are

concerned with using existing dictionaries like the General Inquirer [38].

In terms of manual creation of lexicons, [66] hand annotated a set of adjectives with their polarity information, and then incorporated this polarity information in their lexicon. We elaborate more extensively on the the related work on this in 2.4.1.3.

The approach we propose is a semi-automatic dictionary-based approach, which involves selecting a number of seed words from the corpus under consideration, and using a bootstrapping approach to grow this list, through the use of synonyms. We do this by using a systematic approach of selecting synonyms from a generic opinion lexicon, but we introduce an additional check, in which we only include synonyms which are clearly of the same polarity of the word in question. We do not utilize any manual approach or checks in this, as we aim to develop a lexicon which is free from human bias.

The process of generating this lexicon in depicted in Figure 3.6. It shows the methodology employed in the lexicon generation starting from the document collection. The explanation for the processes involved, and data generated is given in the following subsections.

3.2.7.3 Bootstrapping

Our bootstrapping approach was developed to enable us determine the prior polarities of other emotive words in our corpus, aside from the top 200 selected opinion words in seed words list.

In order to pbtain a domain specific subjective words set, [28] used a bootstrapping technique on WordNet with manually generated seed words, and extended these by their synonyms. Working with adjective only seed words, [38] utilised the adjective synonym and antonym set in WordNet to predict the semantic orientation of their adjectives. They explain the motivation behind this as being that adjectives have the same orientation as their synonyms, and opposite orientations as their antonyms.

In our work, we use adjectives, adverbs and verbs as our emotive words, we hypothesize that they possess the same orientation as some of their synonyms. We design our bootstrapping approach around this.

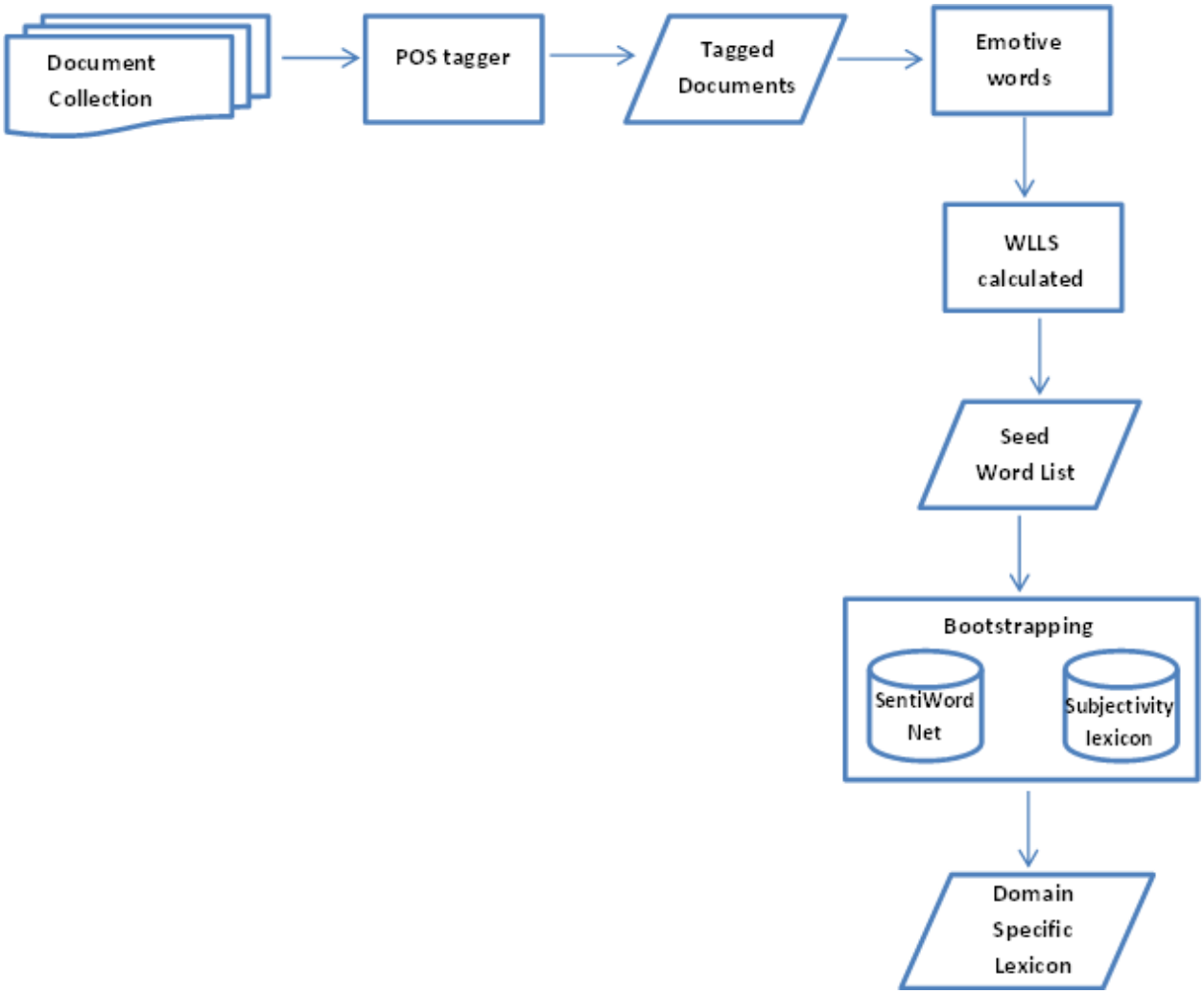

Figure 3.6: Domain Lexicon Generation

Our approach works by taking each word in the seed words list, and looking this up in SentiWordNet. If the word is found, then all the synsets of all the parts of speech that the word occurs in are examined. For every polarity class (positive or negative) being considered, the same polarity score is checked for in each synset. If that polarity score is found to be the highest in a particular synset, then the synonyms "word#1" and "word#2" from that synset are selected and added to the respective opinion words list. The process is repeated until all the words have been considered. If an emotive word does not exist in SentiWordNet, then the word is looked up in the subjectivity lexicon and then entered in the right list with respect to its semantic orientation.

This procedure is depicted in the algorithm in Figure 3.7.

```
Data : Seed Words lists : Positive (List(P) and Negative (List(N), SentiWordNet,
       and Subjectivity lexicon
Result : Opinion words list

For word(i) in List (P) do
  if 'i' in SentiWordNet then
      For POS(i) = adjective
          Check all synsets
          if posscore is highest then
              add word#1 and word#2 from synset to P
          end
      Repeat same for POS(i) = adverb $$ POS (i) = verb
  else
      look up word (i) in Subjectivity Lexicon
          if i = positive then
              add 'i' to P
          end
end

For word(j) in List (N) do
  if 'j' in SentiWordNet then
      For POS(j) = adjective
          Check all synsets
          if negscore is highest then
              add word#1 and word#2 from synset to P
          end
      Repeat same for POS(j) = adverb $$ POS (j) = verb
      end
   else
      look up word (i) in Subjectivity Lexicon
          if j = negative then
              add 'j' to P
          end
end

if word is not in SentiWordNet and Subjectivity Lexicon
   discard word
```

Figure 3.7: Domain specific lexicon generation

After this lexicon is created, we use it in determining the semantic orientation or prior polarity of the terms in the corpus. When creating our composite features (see section 3.2.2), we use the polarity determined from this lexicon in the wildcarding phase. Our goal is to incorporate the polarity of emotive terms which has been decided with respect to the usage of the term in the focus domain. That is, to label words with the semantic orientation determined by their relevancy with respect to a certain polarity class. It is our intuition that a certain word which is normally used to depict a particular sentiment in a certain domain will be picked up as having a higher score with respect to that polarity class in our approach. This would then override whatever semantic orientation the word is generally considered to have, out of context.

3.3 Methodology

This section covers the methodology employed in the development of our hybrid approach. We also give the methodology involved in our lexicon-based and machine learning based approaches.

The methodological approach we employ in this work is that of experimentation. In tackling our research questions, we have explored lexicon-based approaches and tested the suitability of the lexicon we have used in our hybrid approach, for determining sentiment polarity. We have also explored machine learning methods and utilized various different features , both those that have been tried and tested in the literature, and those that have not. We have then in our novel approach combined the two approaches and introduced various features, to arrive at results which we will then compare with the results obtained from the lexicon-based and machine learning , as well as what is obtainable in the literature, in order to highlight the contributions of this novel approach.

In Figure 3.8, we present a breakdown of approaches we have explored, and what each group of experiments entails. This figure shows the experiments and tasks that make up each explored classification approach. It provides an umbrella view of the subtasks that make up

each group. It is not flow diagram. We go further to explain each of these approaches, as well as the aim behind their exploration.

3.3.1 Lexicon-based Approach

As the name implies, this approach to sentiment classification exploits the knowledge in lexicons to determine the sentiment polarity of text. Important aspects of the lexicon approach include the choice of lexicons and other word sources, as well as the mode of constructively combining these lexical knowledge sources for the purposes of sentiment classification.

In the lexicon-based approach, the prior polarity of terms, which is the polarity of the terms in the absence of context is determined from the lexicon. These individual polarities of words, phrases or sentences, are then resolved to obtain the polarity classification of the document. The polarity classification of a document in such an approach can be determined by counting terms of a certain polarity and assigning the document polarity based on a higher frequency of terms of that polarity, or by computing polarity scores for each term, and comparing the total scores of the document. Weighting these scores or frequencies based on their position in text, or according to their perceived importance can also be applied to distinguish features within the feature set, prior to classification.

In the lexicon-based approach in this work, the polarity scores of the terms are taken from the lexicon, and these scores are then resolved to determine the polarity classification of the entire document.

In exploring the means of answering our research questions on the ability of certain sections of text in capturing the polarity orientation of the entire document, summarization was explored in the lexicon-based approach. This involved experimenting with various sections of the text, and comparing the obtained polarity classification of each section with the polarity of the entire document.

Whilst this has been tackled in existing work, we go further to run additional tests on certain promising sections of the text, to determine if the polarity of just a single sentence within that section has the ability to portray the polarity of the entire document.

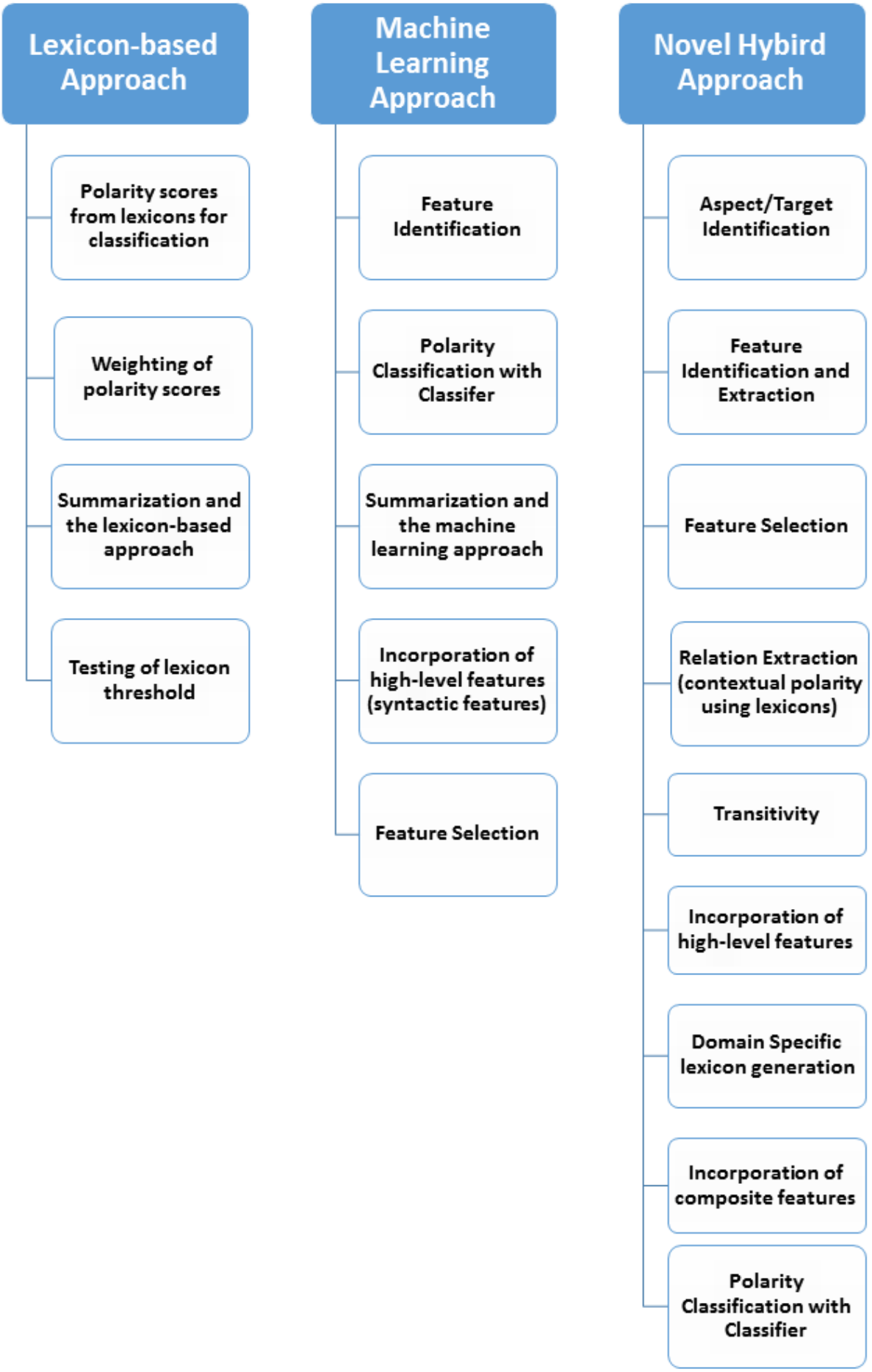

Figure 3.8: General overview of tasks in each Approach

We do not carry out word sense disambiguation as a whole in this work, because it is an area that entails a lot more than just a brief mention. Words in lexicons have various senses. We run a number of tests to determine which sense of the word to consider, when obtaining the prior polarity of terms from the lexicons.

3.3.2 Machine Learning Approach

The machine learning approach is viewed as an approach which treats the sentiment classification problem as just another topic classification approach. It carries out classification at a high level of accuracy, hence its appeal, though the classification process is not always transparent. Not being transparent means that the results obtained are not easily attributable to a certain factor.

Features are an important element in the machine learning classification approach, and can range from the frequency of terms, to the presence of terms, as well as weighted averages and scores. Determining what terms in the document should be considered features is another important factor. In the literature, adjectives have been widely viewed as polarity bearing terms, and hence the frequencies of adjectives as well as their occurrence has been exploited as features for machine learning classifiers.

In our machine learning approach, we select the frequencies of the terms in the documents as our features, and carry out normalization of these frequencies to ensure that more frequent terms that may not be adding useful information to the classification process are not given undue importance.

As was implemented in the lexicon based approach, we classify summaries of the documents in our machine learning approach and compare the results obtained with that from our lexicon-based approach. The aim of this summarization is to explore the suitability of summaries in representing full-texts in terms of their expressed sentiment polarity. The summaries we use here are mainly position-based, which means the sentences are selected based on their position in the document.

In addition to this, we also conduct a classification of summaries which are extracted

from the documents using an established off-the-shelf traditional summarization tool. This tool has been utilized as a benchmark for topic summmarization. The aim here is to test if summaries that have been utilized for topic classification would be useful in polarity classification, and if such sentences which have been shown to aid topic-based classification would assist in sentiment classification as well. To the best of our knowledge, this approach has not been reported in the literature.

Being that the aim of implementing this process is to explore a pure learning based approach, we keep the incorporation of knowledge sources, such as contextual and syntactic information to a minimum. However, we still implement a basic form of word sense disambiguation by incorporating some part-of-speech information in the subsequent experiments. Additionally, some structural information is included in the feature set through the use of higher order ngrams, which have been extracted with an ngram extracting tool from the OpenNLP toolset.

3.3.3 Novel Hybrid Approach

Our novel hybrid approach addresses a number of key challenges, as explained in Chapter One of this work. Amongst these are aspect identification, sentiment classification of documents with respect to the identified aspects in the document and addressing domain specificity of sentiment bearing terms.

Identifying the components of a document about which sentiment is expressed (aspects) is extremely important in our approach. We identify these terms as nouns in the document, setting a support value that enables us select only nouns which are potentially aspects, with respect to a given document.

Certain terms are also classed as sentiment bearing terms, some viewed as being more polar than others. We identify sentiment bearing terms using the verb, adverb and adjective class of words. Dependency relations are often used to extract long range dependencies that exist between the words in a sentence that could be missed if only ngrams are used. Incorporating rich syntactic knowledge in our approach was achieved using dependency relations

as well as ngrams.

In order to extract relations between the aspects and sentiment bearing terms, relations which may not be extracted by the dependency tree or ngrams, we introduce a novel approach of utilizing transitive relations extracted from the syntax rich dependency relations. These transitive relations are added to the feature set which will be used to train the machine learning classifier. This is one of the ways through which we enrich the machine learning classifier with knowledge rich sources.

Being that the machine learning classifier's accuracy is very reliant on the use of the right features, we consider feature selection another key component of this approach. We apply feature selection to get rid of noise in our classification process.

Domain specificity of sentiment bearing terms is another important sentiment classification issue that our approach tackles. We address this in a novel way, through the use composite features and a corpus generated domain specific lexicon. We first generate a domain specific lexicon, using a method which enables us determine and use the contextual polarity of terms from two generic lexicons, rather than the prior polarity of terms. We create variations of composite features which are further added to the feature set which the machine learning classifier is trained on.

To aid with domain transferability, the resources used for this novel approach are entirely corpus generated, with the exception of the generic lexicons which are used to create the domain specific lexicon. This contributes to the portability of the proposed approach, as it makes it easier to apply it to another corpus without the complexity of ensuring that external resources are available for that domain.

The methodology for this is clearly depicted in Figure 3.1. The feature collection is first built from the document collection, and is comprised of ngrams (unigrams, bigrams and trigrams), and the dependency relations extracted using the dependency parser. We then extract transitive relations from the dependency relations based on identified aspects. These transitive relations are added to the feature collection. Feature selection is applied, and the most informative features are identified and retained for classification. Feature vectors are

built from each document set and the machine learning classifier is trained on a selected training set, and tested on the test set, both built from the feature set.

The lexicon-based aspect of this is concerned with the determination of the features, and the building of the feature set from various knowledge sources. The feature selection process and the classification constitute the machine learning aspect of our approach.

The domain specificity problem is addressed with generalizing features and incorporating contextual information through composite features. For this classification, the most informative features are selected from the document set, and these are used to create a domain specific lexicon. Polarity information from this lexicon, as well as part-of-speech information are incorporated in the feature set by wildcarding various dependency relations. Feature selection is again performed on this set and the selected features are utilized for classification with the machine learning classifier.

3.3.4 Validation

In the validation procedure of our approach, we assess how adaptable our approach would be to other domains, besides the the domain we have utilized in its design and implementation.

Bearing in mind that domain specificity is still a big challenge in sentiment analysis, we have modelled our approach in such a manner that it can be immediately adaptable to a new domain when the corpus is changed.

As part of our validation, we will test our approach on a number of review sets from other domains, to assess how adaptable it would be to the different writing styles and formats.

We portrayed in sections 3.2.6 and 6.3 that we would be extracting relations around certain identified aspects. In order to assess the efficiency of our approach in identifying these aspects and extracting relevant aspect focused sentences, we will compare the results we obtain our transitive relations technique of extracting relations around a potential opinion target or aspect.

We will also compare the performance of our domain specific lexicon against a generic lexicon, in terms of determining the polarity of terms.

3.4 Summary

In this chapter we present a novel approach to sentiment classification that is a hybrid of learning based and lexicon based approaches. The motivation behind this is to implement an aspect-focused document classification approach that can achieve a high accuracy, but also utilizes deep linguistic constructs in its design. We aim to assess the performance of this approach as we vary the linguistic constructs that we add as features to the learning classifier. Additionally, we discuss our implementation of the two other approaches which we explore in this work.

This approach also considers the areas which still pose challenges in sentiment analysis, namely domain adaptability of classification methods as well as aspect-based classification. We aim to identify aspects and extract the semantic relationships that exist between them and other words in sentences in order to achieve a document level classification.

Our approach makes the contribution of examining these deep linguistic features in a manner that is both focused on the corpus, and has as little human intervention as possible. It also utilizes some approaches that have not been implemented in the literature to the best of our knowledge, for document level sentiment classification. The transitivity technique which we propose to implement, has not been utilized in the manner we propose, for polarity classification in the literature, to the best of our knowledge.

We also develop a semi-automatic technique of deriving a domain specific lexicon, with a scoring scheme and bootstrapping approach that we have not come across being used in the literature. This scheme which is entirely focused on corpus information, we hope will address the domain specificity issue of the polarity of sentiment words.

In summary, the contribution we make is developing an approach which works with a learning based classifier trained on rich linguistic knowledge sources, which can be easily applied to other domains, due to its lack of reliance on external domain specific knowledge sources. We also make the contribution of an extensive analysis and experimentation on the effects of different linguistic knowledge sources, and their contribution to the classification

process. We explore pure learning based and lexicon-based approaches, in the areas of summarisation, lexicon efficiency, and features exploration.

This approach will address our research questions on obtaining and analysing sentences which correlate with overall classification, identification of aspects which the documents focus on, analysing the relations around them to determine the effect on the overall classification, determining features that potentially influence the classification process and the domain generality of such an approach.

Part II

Evaluation

Chapter 4

Lexicon-based Approach

This Chapter reports on the implementation of our lexicon-based approach. A sentiment lexicon, SentiWordNet lexicon was utilized in performing the polarity classification of movie reviews. Different variations of the experiment were conducted in a bid to determine if there were certain features which could influence the accuracy of the lexicon-based approach. We report the results of these tests in this Chapter.

Our lexicon-based approach will serve as one of the baselines for our hybrid approach. In addition to extensively exploiting the SentiWordNet lexicon to determine its suitability for the polarity classification task, we also explore the performance of summaries when used as representatives of full documents in polarity classification tasks. As part of the classification of summaries using the lexicon-based approach, we implement three novel weighting schemes, focusing on last sentences. Our hypothesis is that last sentences capture the sentiment expressed in full reviews, due to review writing styles, where closing remarks are filled with emotionally charged text. We refer to these tests as Test 1, 2 and 3.

Our intuition when implementing the lexicon-based approach is that even though this approach takes advantage of the rich syntactic and semantic information in the SentiWordNet lexicon, it will still not perform with a very impressive accuracy.

We hoped to address the following questions at the end of this implementation:

- How suitable is SentiWordNet for classifying documents into positive and negative

polarities?

- How do we determine the score to assign to entries in the lexicon with multiple senses, and with different parts of speech?

- How well do lexicon-based approaches work in classifying documents according to their polarities, and how comparable are the results obtained from such a method?

In addition to these, we also looked into answering two of our research questions on learning which sentences correlate well with the overall classification of a review, and if these sentences can be used to generate valid representative summaries of the opinions expressed by the text. Also, how do we determine which of these sentences to include in our summaries, and what number of them would make up the summaries.

We explore the SentiWordNet lexicon and also examine its threshold in order to determine if the midpoint of the lexicon could in some way be altered to enable outliers get classified. This was a crucial experiment because the lexicon will be used throughout our work, and we had to know that the threshold set for positive and negative classification was one that we should retain.

We report our results at the end, and compare this to what is obtainable in the literature where pure lexicon-based approaches have been implemented.

4.1 Lexicon-based Approach to Polarity Classification

The lexicon-based approach is considered to be based on the assumption that the contextual sentiment orientation is the sum of the sentiment orientation of each word or phrase, in a document [71]. As its name suggests, it is concerned with the use of lexicons to carry out sentiment classification. Opinion lexicons associate sentiment orientation and words. The use of opinion lexicons in opinion mining stems from the hypothesis that individual words can be considered as a unit of opinion information, and as such, may provide clues to document sentiment and subjectivity [67].

A more detailed analysis and description of the SentiWordNet lexicon has been given in the related work chapter of this work. SentiWordNet, which builds upon WordNet, is said to have more sentiment related features [18]. The aim behind its design was to provide an extension for WordNet, in order to cope with sentiment directed tasks, so that each synset could be associated with a value, concerning the negative, positive or objective connotation [50].

Figure 4.1 gives a general depiction of a lexicon based polarity classification process. In this case, the lexicon is SentiWordNet.

Sentiment-Classification Features A number of features can be extracted from the lexicon depending on the focus of the research. Among these, the most common is the score assigned to each word, which is usually a triple of positive, negative and objective. This score can be used to formulate features to classify the whole document, such as:

- Summing up all the scores, and carrying out a comparison of negative scores against positive scores
- Finding the average of the scores
- Summing up only the scores of certain parts of speech, like the adjectives, adverbs or nouns
- Carrying out a majority vote, where a document is classified as belonging to a certain class based on the number of words classified as that class, that it has.

Dang et al[18] determined the polarity of a word using SentiWordNet, by carrying out a comparison between the positive and negative score of a word in a certain part-of-speech. Ohana and Tierney [67] used a number of features, including sums of scores of certain parts of speech, average of scores, and percentage of negated terms.

In this chapter, we focused on utilizing SentiWordNet scores as features, and determined the semantic orientation of whole texts by comparing the total negative scores against the

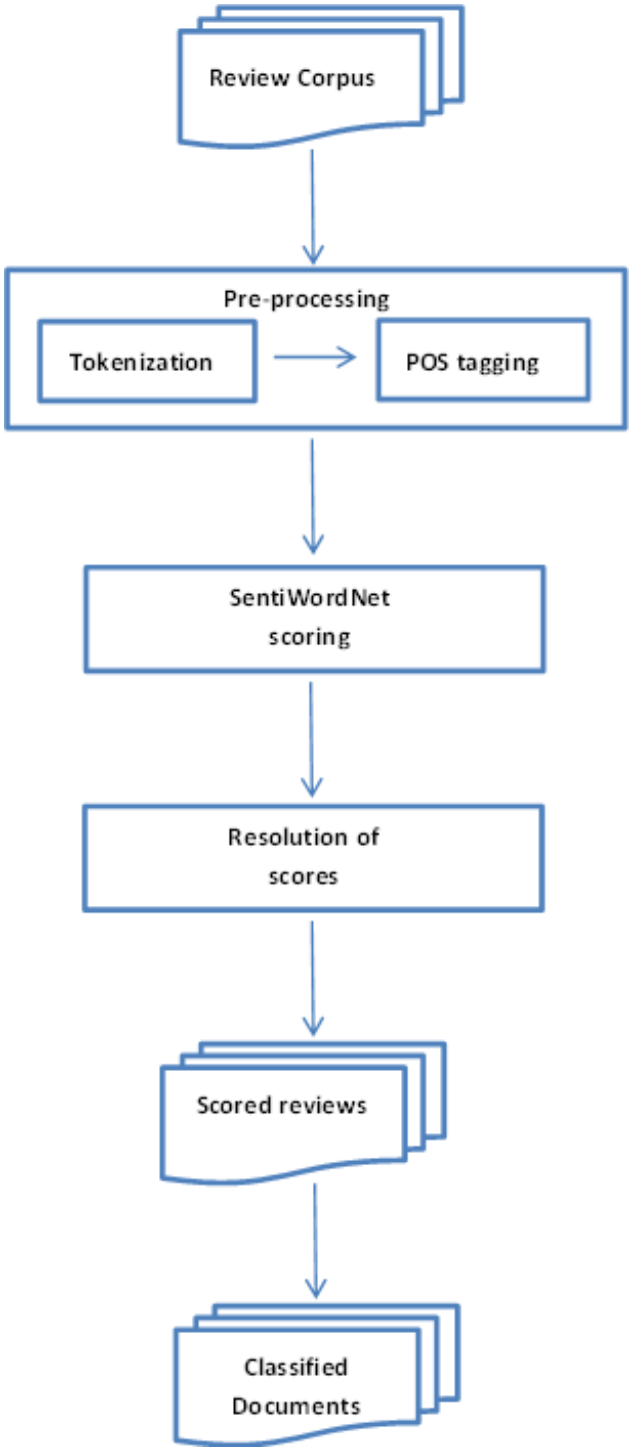

Figure 4.1: General Architecture of Lexicon based classification

total positive scores. The document was essentially treated as a bag-of-words, which is typical of a document-level sentiment classification approach. Additionally, we ran other experiments with a variety of the features we used. These are explained in the following sections.

4.2 Resolving SentiWordNet Scores

The scores in SentiWordNet are represented in triples, which consists of the negative score, the positive score, and the objective score. Due to the fact that each word has a number of senses for each part of speech, deciding the actual triple to assign to a particular word is something that must be considered.

We ran three different variations of this and evaluated the results we obtained. This was important because it would direct our use of the scores of the lexicon for determining the semantic orientation of words for the rest of the work. The three variations we implemented were Primary POSPolarity, POSpolarity and OverallPolarity, and they are described in detail in the following subsections.

4.2.1 OverallPolarity

In the OverallPolarity variation, every sense of a given word in SentiWordNet, and every part of speech was collapsed into one entry for the word. Hence, a word would have a triple of scores, which would be the same for each occurrence of the word in the dataset, irrespective of what the part-of-speech of that occurrence of the word is. This procedure is illustrated in diagram 4.2, and explained in the OverallPolarity algorithm given in Figure4.3. We are only interested in the positive and negative scores, and not the objective score of the word.

The figure above is used to illustrate an example of an entry in SentiWordNet, here, the word "good". It should be noted that this is just a portion of the complete entry for the word "good". The word actually has 21 entries for the adjective sense, meaning it goes up to "good#21, 4 entries in total for the noun sense, meaning it goes up to "good#4", and

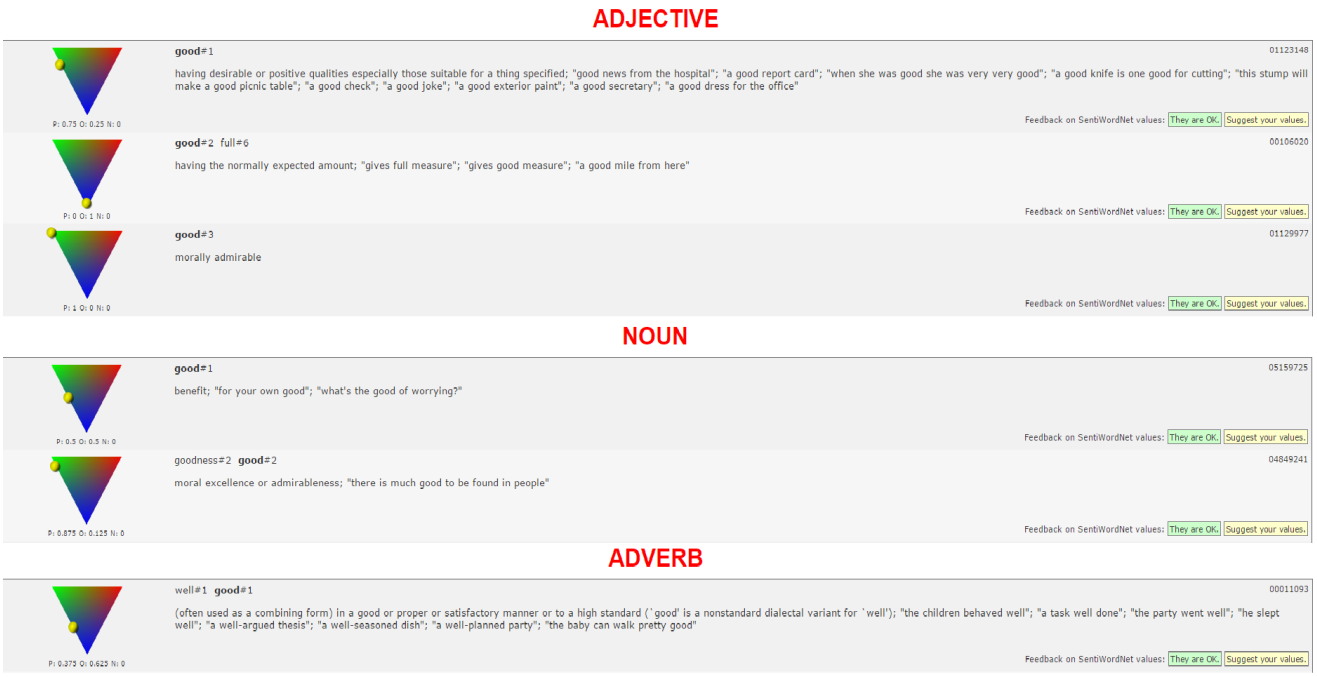

Figure 4.2: An Entry in SentiWordNet

two entries for the adverb sense, going up to "good#2".

As the algorithm shows, if we were to determine the score of the word "good", we would use :

$$PosScore = \frac{\sum(allpositive)}{count} \tag{4.1}$$

$$NegScore = \frac{\sum(allnegative)}{count} \tag{4.2}$$

where,

allpostive: All the positive scores of the word "good", from all the senses in all the parts of speech of the word in SentiWordNet

allnegative: All the negative scores of the word "good", from all the senses in all the parts of speech of the word in SentiWordNet

count: The total number of synsets the word has in SentiWordNet, across all it's parts-of-speech.

```
Data: The Corpus A
Result : Assigning polarity scores using OverallPolarity

For every word (a) in 'A' do
    SenseNum = 0
    Posscore = 0
    Negscore = 0
For POS = adjective
    synset i = 1 to n
    Sensenum = Sensenum + 1
    Posscore(adjective) = Posscore(i) + Posscore
    Negscore (adjective) = Negscore(i) + Negscore
end
Repeat for POS = adverb, noun and verb

TotalNegScore = Negscore(ad) + Negscore(v)+ Negscore(n)+ Negscore(av)
TotalPosScore = PosScore(ad) + Posscore(v) + Posscore(n) + Posscore(av)
(ad = adjective, n= noun, ad = adverb, v = verb)

Negative score of word = TotalNegScore/SenseNum
Positive score of word = TotalPosScore/SenseNum
```

Figure 4.3: OverallPolarity Algorithm

Hence, for the word "good", which has 21 synsets under the "Adjectival" part-of-speech, 4 synsets under the "Noun" part-of-speech, and 2 synsets under the "Adverbial" part-of-speech, the positive score assigned to good would be the sum of all the positive scores for all 21 adjectival synsets and the positive scores for all 4 noun synsets and those of the 2 adverbial synsets, divided by 27. The same procedure will also be followed for the negative scores, and this average will be stored as the positive and negative score for "good" in this approach, respectively.

Symbolically, the scores for the word "good" will be calculated thus:

$$Pos\,Score = \frac{\sum_{i=1}^{21}(Adj\ senses) + \sum_{i=1}^{4}(noun\ senses) + \sum_{i=1}^{2}(Adv\ senses)}{(21+4+2)} \tag{4.3}$$

$$Neg\,Score = \frac{\sum_{j=1}^{21}(Adj\ senses) + \sum_{j=1}^{4}(noun\ senses) + \sum_{j=1}^{2}(Adv\ senses)}{(21+4+2)} \tag{4.4}$$

where,

$i\epsilon(positive\,scores)$; and $j\epsilon(negative\,scores)$

Adj: Adjectives

Adv:Adverbs

This was how the prior polarity score of terms in the corpus were determined for the overallpolarity category.

Implementation The dataset consisting of 2000 documents, 1000 negative and 1000 positive documents was considered as a bag of words, and each word present in SentiWordNet was looked up and it's positive score and negative score were determined by the approach described in the equations above. The precision and recall of these scores were computed, as well as the accuracy. These measures were defined thus:

$$Precision = \frac{tp}{(tp+fp)} \tag{4.5}$$

$$Recall = \frac{tp}{(tp+fn)} \tag{4.6}$$

$$Accuracy = \frac{tp+tn}{(tp+tn+fp+fn)} \tag{4.7}$$

where:

$tp/True\,Positive$:Document was positive and predicted positive.

$tn/True\,Negative$:Document was negative and predicted negative.

$fp/False\,Positive$: Document was negative but predicted positive.

$fn/False\,Negative$: Document was positive but predicted negative.

The results obtained are shown in Table 4.1.

Table 4.1: Results from OverallPolarity approach

Documents	Precision	Recall	Accuracy
Positive Documents	65.8%	63.8 %	65.0%
Negative Documents	64.5 %	66.5%	

4.2.2 Primary POSPolarity

Determining the semantic orientation of words using this second approach was implemented differently. This is in the sense that the part of speech of the focus word was taken into consideration. This was an attempt at word sense disambiguation. We do not make any claim of carrying out word sense disambiguation in this work because it is a more intense area of sentiment analysis, and would deserve its own seperate research.

An assumption we made in this approach was that the first sense of the word in SentiWordNet was usually the most commonly used sense of the word, and as such, there could be a possibility of achieving a better accuracy by considering this to be the intended sense in the corpus.

This approach can be explained figuratively using Figure 4.4, with the same word, "good" as an example. From the previous section, it was stated that in SentiWordNet, "good" had 21 senses under the adjectival part-of-speech, 4 synsets under the noun part-of-speech, and 2 under the adverbial part-of-speech.

In this approach, only the synset shown in the figure were considered.

Implementation The entire corpus was first tagged using QTag. After the part of speech of each word was determined, the words were looked up in SentiWordNet. Each tagged word present in SentiWordNet was looked up under the tagged part-of-speech, and the positive and negative scores of the first synset of the word in that part-of-speech were returned as the positive and negative scores for every occurrence of that word in the corpus. The algorithm below shows the steps taken in this approach. To determine the semantic orientation of a

Figure 4.4: An Entry in SentiWordNet

```
Data: The Corpus A
Result : SentiWordNet tagged reviews
Tag all reviews using QTag
For every word (a) in 'A' do
if POS(a) = j then
(where 'j' can be an adverb, verb, noun or adjective)
     Posscore(a) = (Posscore(sense#1(j))
     Negscore(a)  = (Negscore(sense#1)(j)
```

Figure 4.5: PrimaryPOSPolarity Algorithm

word, it's negative and positive scores were compared, and the orientation with the higher score was assigned to the word. To determine the orientation of the document, the sum of the scores was computed. When all the words in the corpus have been scored, or ignored, the sum of the negative scores of all the words was computed, and compared to that of the positive scores of the words. If the sum of the negative scores was higher than the positive scores, then the document was classified as negative. Otherwise, it was classified as positive.

We will demonstrate the procedure in this approach using the same word used in the previous section, "good". From Figure 4.4, it can be seen, as has been previously established, that "good" has three parts-of-speech in SentiWordNet. The scores for occurrences of "good" will hence be assigned this way:

If "good" is tagged as an adjective: $PosScore = 0.75$, while $NegScore = 0$ If "good" is tagged as a noun: $PosScore = 0.5$, and $NegScore = 0$ If "good" is tagged as an adverb: $PosScore = 0.375$ and $NegScore = 0$.

Figure 4.6: POSpolarity approach - An Entry in SentiWordNet

To evaluate this classification procedure, we used the same evaluation measures explained in subsection 4.2.1, precision, recall and accuracy, defined as shown in equations 4.5,4.6 and 4.7, respectively. The results obtained from this are shown in Table 4.2.

Table 4.2: Results from PrimaryPOSPolarity approach

Documents	Precision	Recall	Accuracy
Positive Documents	56.2%	89.9 %	60.0%
Negative Documents	74.0 %	29.2%	

4.2.3 POSPolarity

In this approach, we utilized both the parts-of-speech of the words in the corpus, as well as the senses. Generally, to assign a score to a word, the synsets in its tagged part-of-speech were all considered. The negative scores of all the synsets in that part-of-speech were summed up and divided by the number of synsets in that part-of-speech, and the resulting average was assigned as the negative score of the word in that part-of-speech. The same procedure was repeated for the positive scores. This approach is illustrated using the synsets from the noun part-of-speech of the word ”good” in Figure 4.6below.

From the figure, ”good” has four synsets under the noun part-of-speech. To determine

the positive score of the word, as well as its negative score, the following formula is used:

$$Positive\,Score = \frac{\sum_{(i\epsilon noun)=1)}^{4}(positive\,scores)}{Number\,of\,synsets\,under\,good} \tag{4.8}$$

$$Negative\,Score = \frac{\sum_{(i\epsilon noun)=1)}^{4}(negative\,scores)}{Number\,of\,synsets\,under\,good} \tag{4.9}$$

which will basically give the positive score for "good" when it occurs as a noun as $((0.5 + 0.875 + 0.625 + 0)/4)$ which will be 0.5, while the negative score will be 0.

Implementation As was implemented in the primaryPOSpolarity approach, the words in the corpus were tagged using OTag, and were looked up in SentiWordNet. If the word was present in SentiWordNet, its polarity score was computed, and it was assigned the positive and negative score. The words not found in SentiWordNet were ignored. The word was then determined to be either positive or negative by comparing the values of the two scores. If the positive score was higher, the word was considered a positive word, and vice versa. When all the words in the corpus have been scored, or ignored, the sum of the negative scores of all the words was computed, and compared to that of the positive scores of the documents. If the sum of the negative scores was higher than the positive scores, then the document was classified as negative. Otherwise, it was classified as positive.

The POSpolarity approach is shown in the algorithm below.

To evaluate this classification procedure, we again used the same evaluation measures explained in subsection 4.2.1, precision, recall and accuracy, defined as shown in equations 4.5,4.6 and 4.7, respectively. The results obtained from this are shown in Table 4.2.

The results from this evaluation are shown in Table 4.3.

Table 4.3: Results from POSPolarity Approach

Documents	Precision	Recall	Accuracy
Positive Documents	57.5%	86.0 %	61.0%
Negative Documents	71.5 %	35.6%	

```
Data: The Corpus A
Result : Assigning polarity scores using OverallPolarity

For every word (a) in 'A' do
    SenseNum = 0
    Posscore = 0
    Negscore = 0
For POS = adjective
    synset i = 1 to n
    Sensenum = Sensenum + 1
    Posscore(adjective) = Posscore(i) + Posscore
    Negscore (adjective) = Negscore(i) + Negscore
end
   Negative score of word = NegScore/SenseNum
   Positive score of word = PosScore/SenseNum

Repeat for POS = adverb, noun and verb
```

Figure 4.7: POSPolarity Algorithm

4.3 Error Analysis of Experiments

The results we obtained from this purely lexicon-based approach is in line with what has been obtainable in the literature. In their approach which was also a purely lexicon-based approach using WordNet, [4] obtained a highest accuracy score of 60.4%, and stated that this had been confirmed to be in accordance with previous findings, which report that it is difficult to surpass the accuracy of 65% using purely lexicon-based approaches.

We based our classification fully on SentiWordNet scores, so this level of accuracy was expected. Zhang[113] implement a lexicon-based approach where they utilize SentiWordNet scores directly and report an accuracy of 64.25%, which is in line with what we obtained.

To trace the factors which influence this classification approach and which have potentially led to these results, we carry out an error analysis.

4.3.1 Positive Reviews

1. The positive reviews which were classified as negative are mostly reviews which contain an extensive description of the plot, where the movie may have been a sad movie, or the plot was a sad one, probably about grief or loss. The reviews could also be reviews of movies in which there were negative characters, which the reviewer writes about. Also, positive reviews on horror movies and some thrillers were misclassified as negative.

2. Another source of the misclassification of positive reviews seemed to have been due to comparisons. This affected the reviews in two ways. The first way was that in a positive review, the reviewer could be scornfully referring to another movie which did not live up to expectations, and which could not be compared to this other movies quality, and these negative words could lead to misclassification. The second way could be a previous part of the movie, which had fallen short of expectations, which the reviewer speaks about, or might be other movies of the actors, or the directors, which did not turn out so well.

3. The third source of this misclassification seemed to be as a result of exceeding expectations. In this case, the reviewer starts by expressing his expected disappointment, and why he had expected to be disappointed by this movie, and could use negative words to express what he had been expecting, but then goes on to admit being wrong, as the movie turned out well. The negative words he uses to express what he had expected from the movie could be a source of the misclassification of the review.

4. Some of the positive reviews which were classified as negative may have been classified as weakly positive, had the degree of polarity been taken into account.

5. It appeared as though the positive reviews classified as positive were basically classified this way because they had a lot of positive words in them.

4.3.2 Negative Reviews

1. Sarcasm :- Most times the reviewers use irony, or worse sarcasm to portray their displeasure with a certain movie.

2. The reviewer could also list what is expected of that genre of movies, and then conclude by saying ; movie A, offers none of this. For example ; An action movie is supposed to be filled with thrills and drills and great action with nice effects, movie B offers none of this. This probably leads to the an increase in the number of positive terms, thereby, bringing about a misclassification of the review.

3. Thwarted expectations seems to be greatly responsible for this as well. The reviewer could start off by outlining what they had expected, before going on to express their disappointment at the movie. This also leads to a high presence of positive words.

4. Another thing which could be responsible is when the reviewer starts off by explaining that though this movie could be appreciated by another audience, they do not share the same sentiment. For example; instinct is the kind of movie inexperienced movie goers will undoubtedly label as powerful or touching. I have a name for it myself; gross.

5. Also, some reviewers could express positive sentiment about some aspects of the movie, like the producer for instance, but their overall sentiment for the movie would be negative.

6. Negation seems to be a problem especially in misclassified negative reviews.

7. Taking note of conjunctions can also help achieve better results from classifying negative reviews.

As a whole, it appeared summarization could help correctly classify negative reviews, more than it would positive reviews. We could infer from this analysis that people seem to want to re-emphasize their sentiments at the end of the reviews, when they are being

negative, and most of the negative reviews end with negative sentiments, directed towards the movies, expressed in the sentences. Also, a number of the negative reviews had negative opening sentences.

From these observations, we make the hypothesis that summarization could be a possible remedy to this, and could perhaps improve the classification accuracy.

This leads us to one of our research questions, given in chapter one of this work, which was:

- Can we learn which sentences correlate well with the overall sentiment classification? Can such sentences be extracted to generate valid representative summaries of the opinions expressed within a body of text?

In the next section, we elaborate on the experiments that we carried out on summarization, using the lexical approach.

4.4 Summarization and Lexical Approach Polarity Classification

4.5 Introduction

Summarization in sentiment analysis has focused on extractive subjective extracts as summaries [13]. Aspect-opinion pairs have also been extracted as summaries [10]. The approaches to summarization implemented in this work are similar to those carried out in [13], and are position based summarization approaches, which include the opening and closing sentences of documents, as well as other variations. We explored summarization as a possible pre-process to polarity classification, that is, to discover if there were segments of a document which could be considered opinion segments, and hence used to represent the whole document.

We steered away from the approach of classifying subjective extracts, but instead utilized the documents in the form in which they were in.

The work presented here was also submitted in a paper titled " Sentiment Classification Using Summaries: A Comparative Investigation of Lexical and Statistical Approaches", and was accepted at the Computer Science and Electronic Engineering Conference (CEEC) 2014.

SentiWordNet version 3.0 was used as the lexicon to determine the semantic orientations of the constituent words in the derived summaries. In determining the approach with which to score the words, we decided to adopt the OverallPolarity approach as it had achieved the highest accuracy in comparison with the other two approaches.

4.5.1 Position Based Summarization Approaches

According to [13], there is an empirical observation that reviewers tend to summarize their overall feeling in a sentence or in a short paragraph, placed either in the beginning or at the end of the review. Ohana and Tierney [67] conducted experiments aimed at proving the existence of areas within documents which tend to carry more opinion context, and specifically focus on the closing remarks of the author at the end of the document. Working on this assumption, we select sentences from different positions, first taking the first sentence, as has been implemented in traditional summarization approaches, then we select just the last sentence, then the last 3 sentences and the first 3 sentences. We have designed these experiments in a similar manner to the work of [13], which we have specified that our work bears the most similarity to. Only the polarity scores derived from SentiWordNet for the words from the sentences were considered here.

The position based summaries we implemented were the first and last sentences of the documents to represent the whole, and also the N-closing and N-opening sentences approach. We did not consider sentences in the middle of these documents as a separate test because from our observation, most times, the bodies of movie reviews focus on the movie plot, than on sentiments. This however, is a test we would carry out in the future, to test if determine the validity of this observation.

Table 4.4: First and Last Sentence Approach Results

		Precision	Recall	Accuracy
First Sentence	Positive Reviews	50%	52%	50%
	Negative Reviews	50%	48%	
Last Sentence	Positive Reviews	55%	59%	55%
	Negative Reviews	55%	47%	

4.5.1.1 First and Last Sentence Approach

In this set of tests, the first and the last sentence of the negative, as well as positive reviews were considered. This gave 1000 first sentences and 1000 last sentences. These sentences were used to represent each of the documents which they were taken from. The purpose of this experiment was to assess how efficient it would be to detect the polarity of a document, using only the first sentence of the review as a summary of the entire review, and also, using just the last sentence of the review as a summary of the entire review. Each document was first represented by just its first sentence, and the scores of each of the words in the sentence was derived from SentiWordNet using the overallPolarity approach. The polarity scores of the words in this sentence were then summed up. If the positive score was higher than the negative score, then the sentence, and hence the document was classified as positive. Otherwise, it was classified as negative. The same experiment was repeated using just the last sentences. The results from the experiment are presented in Table 4.4.

We observed that the accuracy obtained from these two types of summaries were not very impressive. The accuracies are lower than those that were obtained from the approaches in section 4.2. The precision and recall values have also been affected and are lower than those reported in that section. It appears that these summaries are not appropriate representations of the full documents, and they do not appear to solve the problems outlined in the error analysis performed on the classification results in the previous section.

4.5.1.2 N-Closing and N-Opening Sentences Approach

the next type of summaries to be tested are the N-block summaries. We decided to use '3' as the value of our 'N' as this was the value used in [13], and hence we would have some source with which to compare our results with.

For the closing sentences, we decided to take the last-3 sentences of the each review, similar to that tried by [13], who carried out extractive summarization in their work and stated that based on empirical observation, reviewers tended to summarize their overall feeling in a sentence or a paragraph, placed at the beginning or end of the review.

We went on to implement three variations of the last sentences summarization. We call these variations Test 1, Test 2 and Test 3. These three tests have never been tried before, to the best of our knowledge, and the aim for attempting them here is to discover if there are certain sentences within the last sentence segment of a review which are more definitive, where sentiment polarity is concerned.

Test 1 In Test 1, we took the average of the scores of the last three sentences. The positive scores of the three last sentences were added and divided by 3, and the same was done for the negative scores. The resulting values were assigned as the positive and negative score of the document, respectively. The higher polarity score of the two was assigned as the polarity of the document. The motivation behind this design of Test 1 was to enable us obtain the overall sentiment of that section of the review. Our aim was to obtain the average negativity or positivity expressed in these section of the review.

Test 2 In Test 2, we selected what we refer to as the most popular polarity, as the overall polarity for the summary. The three sentences' polarities were inspected, and the polarity occurring the most number of times was selected as the overall polarity. The assumption behind this selection technique was that there would probably be a cluster of the sentiment being conveyed by the review in these sentences, so making a selection based on the most frequently expressed sentiment could lead to predicting the overall polarity of the review.

```
begin
For all sentences in document (i)
    Add all positive scores of words in (i)
    Store result in PosSum
    Posscore of Document (i) = PosSum/3
    Add all negative scores of words in (i)
    Store result in NegSum
    NegScore of Document (i) = NegSum/3

Compare PosScore with NegScore
    If NegScore > PosScore
      (i) is a negative document
    else,
      (i) is a positive document
 end
```

Figure 4.8: Steps in carrying out Test 1

The point to note in this test was that there were few occurrences where one sentence was classified as positive, one as negative, and the third as neutral. In such a case, the solution was to assign the polarity with the highest score as the overall polarity.

Test 3 In Test 3, we selected the highest recorded score among the three sentences as the overall polarity. The positive and negative scores were compared for the three sentences, and the polarity with the highest score was assigned as the overall polarity for the document. The motivation behind this choice was to capture strong opinions which could have been expressed in the closing sentences in order to drive home a point. The results for these tests are shown in Table 4.5.

We repeated the same set of experiments for the first-3 sentences as well. The results obtained from these tests are shown in Table 4.6. Same as with the last-3 sentences, our objective was to detect if the opening section of the review was actually where the reviewers had summarized their overall feeling.

```
begin
For all sentences in document A
   Neg = 0
   Pos = 0

   While sum of NegScores in Sentence j in A != sum of Positive Scores
       if sum of PosScores of words in j > sum of NegScores of words in j
           Sentence j in A is positive
           Pos = Pos + 1
       else
           Sentence j in A is negative
           Neg = Neg + 1
       end
if Neg > Pos
   Polarity of document A = Negative
else
   Polarity of document A = Positive
end
end
```

Figure 4.9: Steps in carrying out Test 2

Table 4.5: Position Based Summary Approach Results -Last 3 Sentences

		Precision	Recall	Accuracy
Test 1	Positive Reviews	59%	67%	60%
	Negative Reviews	61%	53%	
Test 2	Positive Reviews	57%	70%	58%
	Negative Reviews	59%	49%	
Test 3	Positive Reviews	57%	61%	57%
	Negative Reviews	58%	54%	

Table 4.6: Position Based Summary Approach Results -First 3 sentences

		Precision	Recall	Accuracy
Test 1	Positive Reviews	60%	54%	55%
	Negative Reviews	55%	56%	
Test 2	Positive Reviews	48%	56%	53%
	Negative Reviews	53%	51%	
Test 3	Positive Reviews	54%	51%	54%
	Negative Reviews	54%	57%	

```
Given document A with n sentences

begin
Negscore = 0
Posscore = 0

For each sentence in document A
    Negscore = sum of all negative scores of constituent words
    Posscore = sum of all positive scores of constituent words
end
For sentence 1 to n
    compare NegScores
    Temp = max [NegScores]
    compare PosScores
    Temp2 = max[Posscores]
    if Temp > Temp 2
       Document is Negative
    else
       Document is Positive
    end
end
end
```

Figure 4.10: Steps in carrying out Test 3

4.6 Threshold Shifting

This experiment was conducted to test the possibility of some of the misclassification being due to some documents lying close to the boundaries of positive and negative classes. In other words, this was to test the hypothesis that moving the midpoint might help in correcting some of the misclassification of the outliers. This was also to assess if the use of 0 as the midpoint in SentiWordNet was actually correct.

To determine a new threshold to compare against, the total positive scores and total negative scores in the entire document collection were computed. This was divided by the total number of positive and negative words. The resulting score for this was then set as the new threshold. For each document, the negative score was subtracted from the positive score, and the result was divided over the number of words in the document. The result

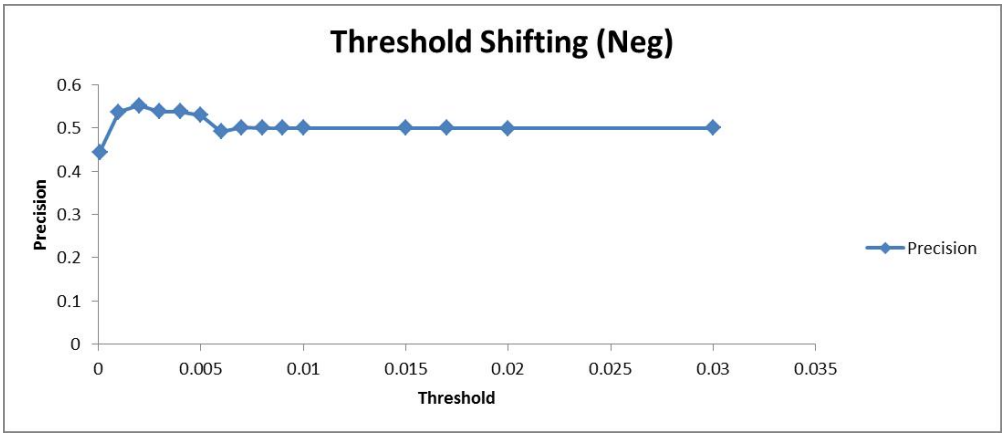

Figure 4.11: The threshold plot for negative values

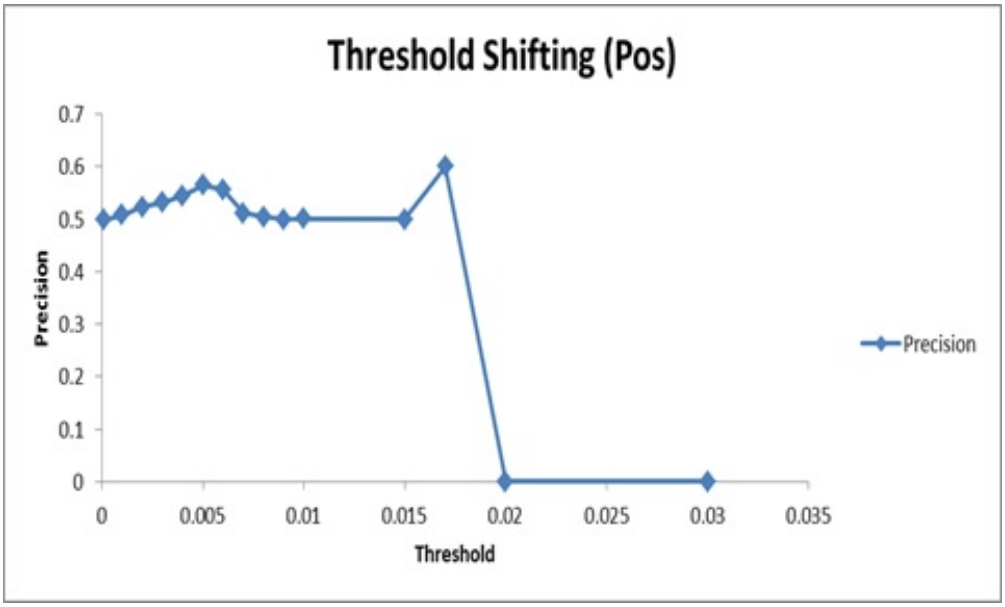

Figure 4.12: The threshold plot for the positive documents

then obtained from this was compared against the new threshold value. If it was greater than the value, then the document was considered positive. If it was less, the document was considered negative. This new threshold value was varied over certain quantities. The plot of the precision value over the different threshold values for the positive and negative values are given in Figures 4.11 and 4.12.

From the results obtained, it was concluded that threshold shifting did not improve the classification results.

4.7 Result Analysis

We compare the results of our method with what is obtainable in the literature, where lexicon-based methods have been used. The lexicon-based approach we have applied is focused solely on the polarity scores of words in SentiWordNet.

Notable lexicon-based approaches which have applied the SentiWordNet lexicon for polarity classification include [67], who used the term counting approach to determine polarity, reading off scores from SentiWordNet, and adding these up to determine polarities. Positive words were then counted, and compared with the number of negative words, to determine the polarity of the document.

Hamouda and Rohaim [34] also utilize SentiWordNet, but attempt some form of basic word sense disambiguation, similar to ours, by first carrying out a prior phase, which they refer to as the SentiWordNet interpretation phase. In this phase, they take the average score according to category, that is, adverbs, adjectives, nouns and verbs. Positive and negative scores of words are calculated by getting the average for its entries according to the categories.

A well known and widely referenced unsupervised lexical approach is the work of [97], who calculated the semantic orientation of phrases using Point-Wise Mutual Information (PMI).

Others include [35] who report work on movie related blogs and also [24] whose work focused on assessing the suitability of SentiWordNet scores for multi-domain sentiment classification.

Annett and Kondrak [4] utilised a number of variations in both the lexicon-based approach and machine learning approach, and found that their best performing lexicon-based system was the system where they first stemmed the words in the blog dataset, before looking them up in a General Inquirer + Yahoo words dictionary. If it was positive, they added this to the list of positive words, and vice versa, and then went on to address thwarted expressions by assigning weights to the words in the dictionary, using the word's minimum

path distance from the pivot words in WordNet.

The accuracies obtained from each of these methods is given in Table 4.7.

Table 4.7: Lexicon-based approaches comparison

Method	Domain	Accuracy (%)
Our lexicon-based approach	Movies	65.0
Turney (2002)[97]	Multidomain	Movies (65.8), Automobiles (84.0), Banks (80.0), Travel destinations (70.5)
Annett and Kondrak (2008)[4]	Movies (blogs)	60.4
Hamouda (2011)[34]	Product domain	Sum on review (67.0), Sentence and average on review (68.6)
Ohana and Tierney (2009)[67]	Movie reviews	65.85
Harb et al (2008)[35]	Movies (blogs)	71.0
Denecke (2009)[24]	Multidomain	Kitchen (58.0) Books (54.0) Electronics (6.05) DVD (59.0) Drugs (52.0) MPQA (40.0)

From the table, we observe that our accuracy of 65% is in line with other lexicon-based approaches, in particular, those that focused on term counting, or lexicon scores of the words. In their work, [4] pointed out that it was difficult to surpass the accuracy level of 65% using purely lexicon-based approaches.The accuracy we obtain here confirms this claim, and this can also be seen in the rest of the pure lexicon-based approaches listed here.

SentiWordNet is well suited as a lexicon for sentiment classification, even for multi-domains, as was concluded by [24] and [67], and its broad area of application justifies our choice to use it as the lexicon in this work, and in our novel hybrid approach.

4.8 Summary

At the beginning of this chapter, we set out to investigate and find answers to the following questions:

- How suitable is SentiWordNet for classifying documents into positive and negative polarities?
- How do we determine the score to assign to entries in the lexicon, with multiple senses, and with different parts of speech.
- How well do lexical approaches work in classifying documents according to their polarities, and how comparable are the results obtained from such a method?
- How do the summaries extracted from the document set perform as representatives of the full document, when classified using the lexicon-based approach?

We ran a number of experiments and at the end we discovered that SentiWordNet as a lexicon is quite suitable for classifying documents according to their semantic orientation, as the results obtained were comparable to other work that have utilized other lexicons, some of which have been manually created from their focus domains . The accuracy obtained from our pure lexical approach is comparable to the accuracy reported in the literature for lexicon-based classification.

Determining the combination or resolution of the scores obtained from SentiWordNet is still a challenging issue, with no one method reported as being the most suitable. We have conducted a variation of experiments and determined that the scores of a word in SentiWordNet, are important across all parts-of-speech and synsets, given that we obtained a better accuracy when we used an average of all the entries of a word in SentiWordNet, to the number of synsets, according to our overallPolarity approach. We feel that this may have been due to the fact that we used the average of scores, and sum total of scores, and not the voting system in determining the polarity of our documents. This observation will be utilized later in the work where we will discuss our hybrid system which also uses SentiWordNet.

Chapter 5

Pure Machine Learning Approaches

5.1 Introduction

In this chapter, we report on the polarity classification experiments that were conducted focusing on purely machine learning approaches. The machine learning algorithm that we used in this work is the SVM Light classifier. The choice to use this classifier was based on the reports in the literature which have shown that it outperforms most other machine learning calssifiers in polarity classification.

In this approach, we experimented with core machine learning approaches, keeping the use of lexicons to the least minimum, and sometimes, leaving it out completely.

Machine learning approaches, as has been mentioned in chapter two of this work rely on labelled data, and this has been pointed out as a drawback of the approach, especially in situations where there is a lack of this data. Machine learning classifiers do not also generalize well to other domains, as a classifier trained on one domain is said to not be transferable to other domains [59][11]. This domain dependency is seen as one of the major drawbacks of the machine learning approach.

A key characteristic of machine learning techniques is that they treat the sentiment classification problem as a simple classification problem, or just any other topic classification problem, and as such, it is assumed that training a classifier on labelled text should be enough

to achieve the classification[11][59][18]. This is one of the hypotheses that we test in this chapter, in an attempt to address some of our research questions.

We also test the reports that the machine learning approach outperform the lexical approach in accuracy by a lot. This is why we keep the use of a lexicon with this approach to a minimum.

We attempt to answer one of our research questions on the possible effects of summarization on polarity classification by classifying extracted summaries with the machine learning approach.

In one of our experiments, word sense disambiguation is implemented at a basic level by incorporating the part of speech of the words in the text in the classifier. In addition to this, we utilized other more advanced features like ngrams, in an attempt to incorporate some context information in our classifier.

5.2 General Machine Learning Approach

There is a standard bag-of-features framework for implementing machine learning algorithms, as highlighted by [74]. Let $\{f_i, ..., f_n\}$ be a predefined set of 'n' features which appear in a document. These can be unigrams or bigrams. Let $m_i(d)$ denote the number of times which f_i occurs in document 'd'. Then, each document 'd' is represented by the document vector: $\overrightarrow{d} := (m_1(d), m_2(d), ..., n_m(d))$.

We will now go on to elaborate on the various experiments which we conducted in this approach, and how the results we obtained compare with other reports in the literature that have implemented similar approaches.

5.3 Features

The first key component in any machine learning technique is the features. These will make up the elements of the vectors which the classifier works with. We gradually modified the

feature set which we trained the classifier on, in our approach, and the most basic feature was the term frequency. There have been contrasting reports in literature about the approach which would give better classification results, between term presence and term frequency.

Both features are defined below. We also utilize another feature, the normalized frequencies.

5.3.1 Term Presence

Term presence is a weighting scheme which measures the occurrence or non-occurrence of a word in a document or sentence.

Using term presence, and not term frequency has in some cases been found to lead to better performance from classifiers in Sentiment Analysis. This is in contrast to topic classification. A possible explanation for this has been given as being that topic is conveyed mostly by particular content words that tend to be repeated [74].

Terms that appear only once are said to be good indicators of subjectivity [100].

5.3.2 Term Frequency

Term frequency refers to the number of times a term occurs in a document or a sentence. The term frequency has been used as a classification measure for both the lexical and the machine learning approaches, as well as the hybrid approach.

The frequency count weighting scheme is used to assign weights to features, where a feature's frequency is used as the feature value.

In order to ensure that some words which may have been used too frequently in the dataset do not exert more influence, though they may be relatively unimportant to the classification process, we decided to use normalized frequencies as our features.

5.3.3 Normalized Frequencies

We use the normalized frequency of each word in the document. Given a word a in a document 'd', the normalized frequency of 'a' was computed as:

$$f = \frac{n}{N} \tag{5.1}$$

where: f : The normalized frequency

n : The number of times 'a' occurs in 'd'

N : The number of words in 'd'.

The document vector will be : $\overrightarrow{d} := (f_1(d), f_2(d), ..., n_f(d))$.

5.4 The Classifier

The machine learning classifier that we use for our experiments is Joachims' SVM_Light [39]. Support Vector Machines are one of the most popular supervised learning algorithms, not just in the area of Sentiment Analysis, but also in other areas of classification. Support Vector Machines are fast algorithms which perform with state-of-the-art accuracy. They are used for creating feature-vector-based classifiers. Each instance to be classified is basically represented by a vector of real-numbered features.

In the two category classification, the general idea behind it is to find a hyperplane, which is represented by a vector, say $\vec{v}$, that does not only separate the document vectors in one class from the other, but does so by a margin which is as large as possible [74].

The training data is used to generate a high-dimensional space which can be divided by this hyperplane, between positive and negative instances, as the case may be. New instances are subsequently classified by finding their position in space, with respect to the hyperplane.

Symbolically, this search corresponds to a constrained optimisation problem which lets $c_j \in \{1, -1\}$, which correspond to the positive and negative class be the correct class of a

document d_j , the solution will be written as shown:

$$\vec{v} := \sum_j \alpha_j c_j \vec{d_j}, \alpha \geq 0, \tag{5.2}$$

Where α_j's are obtained through solving a dual optimization problem. Those $\vec{d_j}$ where α_j is greater than zero are referred to as 'support vectors', as they happen to be the only document vectors which contribute to $\vec{v}$. Classification of test instances is therefore about determining which side of $\vec{v}$'s hyperplane they fall on [74].

The classifier has a learning model, as well as a classification model. The latter can be used to apply the learned model to new training examples [39]. The classifier also has different kernel settings, which include the polynomial kernel setting, Radial Basis Function (RBF) kernel setting, and the sigmoid kernel setting. We implemented different kernel settings in our experiments to determine which setting would give the highest accuracy.

5.5 Experimental Setup

The same dataset we used for the lexical approach was used here, which is Pang and Lee's Cornell movie review dataset comprising of 1000 positive reviews and 1000 negative reviews. The reviews were tokenized and the normalised frequencies of the words in the dataset were computed. These were then fed into the SVM_Light classifier, each document represented by a vector of normalised frequencies of its constituent words.

We split the dataset in two, the test set was made up of 100 documents, and the training set was made up of 900 documents. We ran the classification and performed 10-fold cross validation. The average accuracy obtained from these is reported in Table 5.1.

5.5.1 Stop words Removal

Stop words are extremely common words which we believe will not contribute to the classification process. The list of these stop words is referred to as a stop list or stop word list.

They are various stop words lists, or stop lists which are used in various areas of natural language processing, like in information retrieval and summarization. Stop lists can also be created as needed by sorting terms based on the total number of times they appear in the document set, and then taking out these most frequent terms as stop words.

As part of the experiments we performed, we took out stop words which could possibly introduce noise into the classification process and re-classified the documents again, without the stopwords. We executed this by using an already available stop words list, prepared for sentiment analysis.

This led to quite an increase in the accuracy we obtained, confirming the hypothesis that leaving such words in the classification process had the potential to introduce noise. The results obtained from this is also shown in Table 5.1. It can be noted from the Table that there is an increase in accuracy of about 5 points.

Table 5.1: Full documents Classification

Test	Precision	Recall	Accuracy
All Words in document collection	78 %	79%	78%
All Words in document collection with stopwords removal	83%	86%	84%

5.5.2 Different Kernel Settings

To evaluate the effect of different kernel settings on the classification results, and subsequently decide on which setting to use, we repeated the classification process without stop words removal. We experimented with the three kernel settings; Polynomial setting, Radial Basis Function (RBF)setting, and the sigmoid kernel setting. We compared the results obtained against the default linear kernel setting.

We did not observe any changes in the results obtained, as shown in Table5.2. We therefore decided to maintain the default linear kernel setting for the subsequent experiments.

Table 5.2: Different Kernel Settings Test

All words in document collection	Precision	Recall	Accuracy
Polynomial kernel setting	78%	79%	79%
Radial basis kernel setting	78%	79%	78%
Sigmoid kernel setting	78%	79%	78%

5.5.3 Excluding Objective Words

Objective sentences or words have been mentioned in the literature to adversely affect the accuracy of the sentiment classification process, due to the belief that they introduce noise. There have been reports of increase in accuracy being achieved due to discarding objective material [66][72], as there have been a number of approaches that have been designed around extracting only the subjective material in text and classifying these only [38][28].

To exclude the objective words in our approach, we designed an experiment in which we classified the documents using only the words that were present in SentiWordNet. Every other word that was not in SentiWordNet was discarded. The results we obtained from this are shown in Table 5.3.

Another variant of the experiment we carried out was to leave out the neutral words in SentiWordNet. We defined neutrality as the words in SentiWordNet which have a zero positive score and zero negative score, which means that they had an objective score of '1'. We do not define neutrality as words whose positive score in SentiWordNet was equal to the negative score.

Table 5.3: SentiWordNet words

Text	Precision	Recall	Accuracy
Only words in SentiWordNet as features	82%	82%	82%
Only words in SentiWordNet with stopwords removal	83%	85%	84%
Only SentiWordNet words without neutral words	81%	85%	82%

From the results we obtained, we observed that using only words which are present in SentiWordNet increases the accuracy from 78%, which is the accuracy obtained from using all the words in the document collection, to 84%, which is quite an improvement. However, we also observed that when we exclude the stop words, the accuracy we obtain from using

only the words from SentiWordNet equals the accuracy we obtained when we excluded stop words from the whole document collection classification as well. Hence, we cannot make a definite statement that the exclusion of objective words does lead to a better accuracy at this point.

5.6 Summarization and Machine Learning Classification

We implemented summarization in the previous chapter, using lexicon-based approaches. We run some experiments on the same sections of text in this approach as well, in order to provide for comparative analysis.

We made the decision not to follow the approach of summarizing by extracting only subjective sentences, as in [13]and [38], but instead, to apply the machine learning algorithm to the same summaries that we applied the lexical approach to. This will also enable us carry out a valid comparison between the two approaches with respect to classifying summarized texts.

Another motivation behind these tests was to assess the efficiency of using machine learning approaches with summaries, against full texts classification. This would inform us as to whether these approaches have picked up on the hypothesis that reviewers tend to summarize their opinions in certain sections of the document, like the closing sentences, or opening sentences [67],[74].

5.6.1 Position-based Approaches

The position-based summarization approaches have already been introduced in the previous chapter. In this approach we classified sentences based on their position in the text, as was implemented in the lexicon-based approach. We selected the N-opening and N-closing sentences to represent the documents.

Implementation As a pre-processing step, we removed stopwords from the documents. We set 'N' to '1' and to '3' as was done in the lexicon-based approach. We selected the first sentence in the 1000 positive reviews and 1000 negative reviews, and the last sentence in the 2000 reviews as well. We then used these sentences as representatives of the full- text. In the next variant of the experiment, we selected the first-3 sentences from the 1000 negative reviews and 1000 positive reviews, as well as the last-3 sentences for the last variation of the experiment.

We computed the normalized frequencies of the words in these summaries and fed these to the SVM_Light classifier as features. The results obtained from the classification of these reviews is shown in Table 5.4.

Table 5.4: Last and First Sentences Classification

Test	Precision	Recall	Accuracy
Full reviews	83%	86%	84%
First Sentence	57%	61%	57%
Last Sentence	60%	69%	62%
First-3 Sentences	58%	64%	58%
Last-3 Sentences	70%	73%	71%

As can be observed from Table5.4, the summaries performed worse than the full documents in terms of classification accuracy. This is in line with what was reported by [13], which is the most similar approach to ours.

From the results obtained, we discovered that position-based summaries appeared not to be effective for sentiment classification, and hence, we would have to implement other approaches to incorporate the conceptual aspect of the documents, and to improve our accuracy of classification.

5.6.2 Open text Summarizer

As has been pointed out earlier in this chapter, the machine learning approach basically works by considering the sentiment classification process as another form of topic-classification. It makes the assumption that applying a machine learning classifier directly to documents for

sentiment-based classification, will be effective, as is the case with topic-based classification.

Working along this line, we made the assumption that summaries created from a topic-based summarization tool would probably be representative enough for documents, in terms of possessing a semantic orientation which is the same as that of the full-text.

To test this intuition, we decided to utilize an open source summarization tool which has been used as a benchmark in the literature and positively appraised for its performance. We used Rotem's Open Text Summarizer (OTS) [85], an automatic text summarization tool which is an open source tool and also a library as well as a command line tool. OTS has been used in [106], and [14], among others.

5.6.2.1 How OTS Works

OTS works by reading a text and deciding which sentences are important and which are not. OTS supports over 25 languages. It is both a library and a command line tool. The program either prints out the summarized text or HTML. If the document is in HTML, the important sentences are highlighted [85].

Grading Words The algorithm performs stemming on words. It then performs a term count, considering only word stems, and sets a hierarchy according to the number of occurrences. The important words are considered to be the top 100 words with the highest count.

Grading Sentences The score of a sentence is obtained by considering all the word's occurrences in the sentence. For every occurrence, you take a score, that is:

$$wordscore * keyVal \tag{5.3}$$

where :

$wordscore$: is the number of global occurrences of the word we are considering; and

$keyVal$: is a factor that can be 1,2 or 3, depending on whether the word we are considering

is one of the top 4 (by global occurrence number) or not.

All repetitions are then added. If the same important word appears 4 times, you add the same score 4 times.

Highlighter This considers sentences according to their scores, and returns the highest scored sentences up till the required summary percentage. The sentences are returned in the order in which they appear in the document.

The Second Grader The 2nd grader multiplies the score of the first line/title by 2 because it obviously considers this line to be very important. Then, the score of each line that starts with a paragraph is multiplied by 1.6.

5.6.2.2 Experimental Setup

The experiment carried out with OTS was performed to evaluate the performance of the classifier on automatically generated summaries. These summaries were used in place of the full texts. Summaries of different ratios were classified. We started by setting the summarization ratio to 20%. This ratio was then varied in order to create summaries of various ratios, to test their performance, and to detect if there was an ideal ratio by which the summaries created were representative enough of the full documents, or if they could give a better classification accuracy than full documents.

The documents were first summarized using OTS, after which the normalized frequencies of the summaries were computed and fed into the classifier and classification was carried out. Stop words were removed. As a prerequisite experiment, we classified the 20% summaries using only the words in the text that were present in SentiWordNet. We then classified again using all the words in the dataset, and found that using only the words present in SentiWordNet yielded an accuracy of 66%, over the 20% ratio summaries, while using the entire document set yielded an accuracy of 68%.

From this result, we decided to carry out subsequent tests considering all words in the documents, other than just words present in SentiWordNet.

The results obtained from classifying the summaries of various ratios are given in Table 5.5.

Table 5.5: Open Text Summarizer

Summarization Ratio	Precision	Recall	Accuracy
10%	63%	67%	64%
20%	67%	72%	68%
40%	73%	77%	74%

We observed that there was a correlation between the accuracy and the ratio of summarization. As the ratio increased, the accuracy also increased. As the summaries tended to the full document, the accuracy also got better, hence, we concluded that using the OTS tool did not create summaries that could be used in a positive way for sentiment classification.

5.7 Word Sense Disambiguation

Word Sense Disambiguation is hardly an aspect of sentiment classification that can be covered in a section of this piece of work. It is an intense area on its own, and would require its own full project. This thesis does not attempt to cover Word Sense Disambiguation, or as it is usually referred to, WSD.

What this section focuses on is part-of-speech tagging, a basic form of WSD. In the next experiments, part-of-speech information was incorporated. Tagging a piece of text according to its part-of-speech has been widely used in the area of sentiment classification, especially in distinguishing different textual classes, like adverbs, adjectives,and verbs, which are considered to be emotion carriers in text. Adjectives are especially considered as this.

There have been some instances in the field where the POS information was added to the features fed to the classifier in different forms. Part-of-speech information has been appended to text in order to generate language patterns, using association rule mining [54].

Implementation The documents in the dataset were tagged using Oliver Mason's QTag, as in the lexical approach. The part-of-speech of each word was appended to the word,

and these words with appended POS were added to the feature set as additional features. For example, given a word like 'House', which could be a noun or verb, the noun form of the word would be tagged like so; 'House_NN', where 'NN' is the part-of-speech tagged by QTag. 'House_NN' was considered as a new feature, and was added to the feature set, as an additional feature to 'House', but not as a replacement.

Two experiments were run on this augmented feature set. In the first experiment, the feature set consisting of the single features, together with the features appended with their Part-Of-Speech was fed to the classifier. In the second experiment, the set of new features, that is, the words with POS appended were used as the only features.

In the former case, a precision of 83%, a recall of 86% and an accuracy of 84% were obtained. In the latter case, a precision of 83%, a recall of 85% and an accuracy of 84% were obtained. From the obtained results, no change in accuracy between the two different variations was observed. We hypothesize that this may be due to a couple of factors, like the fact that we did not perform any feature selection, hence there may have been too much noise in the classification process.

5.8 Validation

The next series of experiments were concerned with applying the classification approach to other domains. One of the key challenges of Sentiment Analysis as has been previously established, is that of cross-domain classification. It is a well known fact in the field that classifiers trained on one domain, or those that work well on one domain tend not to do so well in other domains.

The cross domain validation we carried out at this stage was to achieve two main aims, the first one being to test the hypothesis that a classifier that works well on one domain would not perform equally well in the other, due to the different lexical characteristics of the words used in different domains. The second was to assess how the approach would fare with unbalanced datasets, being that the Cornell reviews movie dataset was a balanced set

of 1000 positive and 1000 negative documents.

Implementation We used the Multi-Domain Sentiment Dataset (MDS) used in [12]. In this dataset which is made up of product reviews from Amazon, we ran experiments on the Beauty, Computer and Video games, and the Video datasets. The video dataset is made up of movie reviews, and would serve the purpose of providing us with the means to compare how well adaptable the machine learning classifier was on classifying movie reviews from another source, and of another structure different from that of the Cornell reviews.

We preprocessed the data by extracting the review text and then tokenizing the reviews and removing stop words. The normalized frequency of the words was computed and this was fed to the classifier as features.

The results obtained are shown in Table 5.6.

Table 5.6: Other Datasets Test

Product	Precision	Recall	Accuracy
Beauty	82%	93%	82%
Computer and Video Games	85%	98%	87%
Video	76%	90%	81%

From the classification accuracy obtained, we observed that the classifier fared well on the other reviews, attaining an accuracy score close to that of the movie reviews. It can be argued that this is due to the similarity of the language used, especially in the Computer and Video games domain and the video domain.

We must clarify that we did not train the classifier on one domain, then attempt to use it to classify documents from another domain, neither did we create a pool of features extracted from all domains, as is often the case in cross domain sentiment classification.

5.9 Higher Order N-grams

Higher order n-grams are believed to add some form of context into the classification process. Bigrams and trigrams have been reported to be good at capturing local dependency [66].

There has however been some contradictory reports about the value they add, and if utilizing them actually benefits or harms the classification accuracy.

Ng et al [66] report an increase in accuracy when bigrams and trigrams were added to the feature set, initially consisting of unigrams only, for classification. Dave et al in [23] also report good performance obtained in classifying reviews using only bigrams and trigrams, while [74] report quite the opposite, that bigrams are not useful features, used alone, or in conjunction with unigrams.

We utilized these higher order ngrams in determining classification accuracy. The results we obtain would inform our decision on incorporating these higher order ngrams in our final framework.

5.9.1 Implementation

We performed preprocessing, which included tokenization and stop words removal on the Cornell Movie reviews dataset, before extracting the n-grams.

We used the Ngrams Statistics Package (NSP) from the OpenNLP toolkit to extract our bigrams and trigrams. We set a threshold of considering only bigrams and trigrams of frequency from '2' and above in the document collection. Numbers were not considered, but punctuations were. We based this choice on the latter being more relevant in terms of capturing contextual information. The normalized frequencies of the selected n-grams were computed, and these were then fed to the SVM_Light classifier as features.

The results obtained from these tests are given in Table5.7.

Table 5.7: Incorporating Higher Order Ngrams

Ngrams	Precision	Recall Accuracy	
Unigrams + Bigrams	77.2%	76.5%	76.9%
Unigrams + Bigrams + Trigrams	77.5%	77.1%	77.3%

From the results we obtained, we observed that adding the higher order ngrams to the feature set has an adverse effect on the classification accuracy. This goes against what we had expected, though it falls in line with what was expressed in [74]. We hypothesize that

this poor result may have been related to our not ensuring that the number of higher order n-grams and that of unigrams were close, as we believe this will ensure that the impact of the unigrams were not undermined by the higher-order ngrams. There may have been an increase in the dimensionality of the feature space due to the number of bigrams and trigrams added [66].

5.9.2 Different Weighting Scheme

As a further test of the effect of higher order ngrams, we utilized a different weighting scheme. We decided on term presence, as according to [45], a good sentiment term should be discriminating and prominent, and the appearance of such a term imposes a greater influence on the judgement of the analysis system. Pang et al [74] had also provided empirical evidence that the use of term presence over term frequency was more effective in data driven sentiment classification task.

We repeat the tests, using unigrams that appear at least '3' times in the dataset, bigrams that appear at least '2' times, and trigrams that appear at least '2' times. We use term presence, with a binary document vector, setting the value of the word in the document vector to '1', if the word appeared in the selected set.

We ran the tests again, and obtained the results in Table 5.8.

Table 5.8: Incorporating Higher Order Ngrams with Term presence

Ngrams	Precision	Recall	Accuracy
Unigrams + Bigrams + Trigrams	89.7%	78.9%	84.9%

We obtain an accuracy of 84.9% for using higher order dependencies, when we utilize term presence as the weighting scheme, against using normalized frequencies. This lends support to the claims made by [74], about the importance of term presence in sentiment classification.

5.10 Result Analysis

We have observed that our best performing system in the machine learning approach is that of using term presence as frequency for features in the SVM Light classifier. We obtained an accuracy of 84.9%, a slight improvement over using unigrams only. We compare this result with other systems in the literature that have utilized the machine learning approach. We compare with the state-of-the-art in machine learning approaches in Table 5.9.

We compare our machine learning approach with two pioneering systems, Pang et al [74], and Pang and Lee [72] which have been widely cited in sentiment classification with machine learning approaches. Pang et al [74] used three different machine learning algorithms, SVM, Naive Bayes (NB) and Maximum Entropy (ME) with unigrams as features. They carried out classification with various feature sets, including appending the part-of-speech of each word to the word itself as a crude form of word sense disambiguation, and also giving the words different weights based on their position in text. They found that using word presence information as a feature yielded better results than frequency, and that position information did not lead to improvements in accuracy.

Pang and Lee [72] presented a novel machine learning system that applied text-categorization to the subjective parts of a document only. The subjective portions were extracted using efficient techniques for finding minimum cuts in graphs. They claim that this facilitates the incorporation of cross-sentence contextual constraints. They also included the degree of proximity between pairs of sentences in their feature set.

Though the work of [72] is presented as a novel machine learning approach, we believe that the system performance is as a result of incorporating context by exploring inter-sentence relationships, and also attempting to include syntactic information in the classification process. This work did not perform machine learning based classification as a bag of features fed to machine classifier, due to the syntactic knowledge sources exploited for the feature set. It is our believe that this led to the high accuracy reported in this work.

Annett and kondrak [4] utilized term presence and term frequency information of un-

igrams, as well as three additional aggregate features; the number of positive words, the number of negative words, and the number of neutral words. They trained and tested three different algorithms on these feature sets, SVM-Light, Naive bayes, and ADTree. They noted that unigram feature representations were the most effective across all their algorithms, which we agree with, based on our results in Table 5.7. They however also found that using frequency as features outperforms presence, which we do not agree with, based on the performance of our machine learning approach.

An unsupervised machine learning approach is implemented by Ng and Dasgupta in [22]. They propose a novel clustering framework with works with user feedback. Due to problems with K-means, they opted for spectral clustering which was proposed by [65] and works by reducing the dimension space of text while retaining as much information as possible about the original space as possible. The data points in this low-dimensional space are subsequently clustered. The SVM classifier is trained on the partitions where data points in the same cluster belong to the same class, and used to classify their documents. Their system is tested on mulitple domains, utilizing frequency as presence.

Table 5.9: Machine Learning approaches comparison

Method	Domain	Features	Accuracy (%)
Our Machine-Learning approach	Movies	unigrams, bigrams, trigrams	84.9
Pang et al (2002)[74]	Movies	unigrams, POS, Position information	SVM (82.9), NB (81.5),M.E (81)
Pang and Lee (2004)[72]	Movies	Minimum graph cuts	SVM (87.15), NB (86.4)
Ng and Dasgupta (2009)[22]	Multidomain	unigrams	Movies(70.9), Kitchen (69.7), Book (69.5), DVD (70.8), Electronics (65.8)
Annett and Kondrak (2008)[4]	Movies (blogs)	unigrams,count of words	SVM Light (77.4), NB (77.5), ADTree (69.3)

Looking at Table 5.9, machine learning techniques clearly outperform their lexicon-based counterparts. Our machine learning approach achieves an accuracy that is on par with other machine learning approaches, attaining a higher accuracy than Pang et al [74], on the same movie reviews dataset that was developed from their work.

We see that some of the reported work in the literature that used term frequency as features outperform those that used term presence, and the reverse has also been reported. We found term presence to perform better than normalized frequencies in our implementation, especially when higher order ngrams are added to the feature set. Unigrams have been reported to give good results where they have been used as the chosen features. In our approach, we also found that unigrams gave impressive results when used on their own, especially when normalized frequencies were used to determine the feature set. It is our intuition that the adverse effect reported when the normalized frequency features are used with unigrams and the higher order ngrams, is due to the unigrams being drowned out by the number of these higher order ngrams. We believe that this can be addressed using an efficient feature selection technique.

It is also worthy to stress that the accuracy obtained by [72] shows that perhaps adding more linguistic and syntactic information to a machine learning classifier leads to better accuracy than using just vectors derived from bag of words information. This gives credence to the intuition that our novel hybrid approach which works on a feature set enriched with linguistic knowledge sources will also lead to better results than pure machine learning or pure-lexicon based approaches.

5.11 Summary

In this chapter, we utilized an approach for classification based on pure machine learning techniques. We carried out a number of experiments on various feature sets to determine the features most capable of improving the classification accuracy. In an attempt to incorporate context information, we incorporated higher-order ngrams; bigrams and trigrams to the

feature set used for classification.

We observed an increase in accuracy when stop words were removed, and we put this down to the elimination of noise from the classifier. We also observed that there was a decrease in accuracy when we introduced higher order ngrams into the classifier. This was however not the case when we used a different weighting scheme, term presence, rather than normalized frequencies. We observed a significant increase in the accuracy, with p-value of 0.00512, at $\alpha = 0.05$. We are inclined to conclude that the term presence weighting scheme is better than the normalized frequency scheme, in sentiment classification.

Though ngrams are said to capture short range dependencies, we still need other techniques to capture long range dependencies, and to incorporate context in the classification approach.

We have also not considered opinion aspects in this approach, which is implemented by still classifying the documents as a whole, irrespective of constituent opinions directed towards the aspects of the reviewed entity. This approach still treats the documents as a standard bag of words, and we aim to introduce more syntactic structure into our design. We have however shown in this chapter that the learning based approaches perform better in terms of classification accuracy, in comparison to lexicon-based approaches.

We need a more detailed technique that takes the lexical properties of the text into consideration, and is aspect focused. We also need better feature selection techniques and a bit of word sense disambiguation in our approach. To be able to achieve this aim, we need to find a way to incorporate lexical information, preferably from a lexicon, into our machine learning process. We hypothesize that such an approach will have the added advantage of a high accuracy, as well as taking the lexical properties of the text into consideration. We believe our novel hybrid approach addresses all of these.

Chapter 6

Hybrid Approach to Sentiment Classification

In this Chapter, we provide the description and implementation of our novel hybrid approach which integrates lexicon-based and learning based approaches.

The two approaches described in Chapters Four and Five, being the purely lexical and purely learning based/machine learning approaches respectively, have their benefits and drawbacks as have been extensively covered in Chapter Two of this work.

As a recap, the lexicon-based method, though having the advantage of producing explainable and readable results, also suffers from a low accuracy, which is evident in its performance in Chapter Four of this work. It has the limitation of determining the sentiment polarity of words based on the sentiments of a set of words which appear in the body of text. The major challenge in this is that the polarity of sentiment bearing terms is domain and context dependent [82]. The purely machine learning based methods have the advantage of high accuracies, but how this is obtained is mostly unclear. It may therefore be a bit challenging to pin-point the exact feature which produces, or enhances this accuracy. The method treats the sentiment classification problem as just another classification problem [59][18], and also suffers from dependency on writing style and struggles with the domain dependency, where sentiment polarity classification is concerned [59][11].

There is however a desire to understand the effect of linguistic knowledge sources on the classification process.

Our approach exploits various linguistic knowledge sources, and incorporates this knowledge in the learning based classification process, to obtain a classifier trained on contextual and syntactic information. The aim is to maintain the high accuracy of learning based approaches, as well as their efficiency.

The main focus of this approach is to determine the sentiment expressed about the aspects of a product or service in a sentence or document. We focus more on this, than we do on obtaining the overall sentiment of the document as a whole. The goal is to take into account all the individual sentiments about each important aspect, when determining the sentiment of the entire body of text.

We therefore design our approach as a aspect-focused, rather than a general bag-of-words method. In addressing this, we utilize syntactic information obtained from dependency trees, and extract potential relationships between aspects and sentiments using transitive relations. This approach of utilizing transitive relations extracted from the dependency tree has not been used in the literature, to the best of our knowledge.

We further incorporate and generalize these features created from dependency tree relations by generating composite features through wildcarding. To address the domain dependency of the polarity of sentiment bearing terms, we develop a semi-automatic dictionary based approach and utilize this for generating a domain specific lexicon. Further syntactic and semantic information is added to the composite features, to enable us assess their impact on the classification results. The aim of this is to obtain document classification, with respect to domain specific terms and contextual polarity.

This hybrid classification approach was designed based on our intuition that a system such as this, which is aspect focused and incorporates linguistic knowledge in a learning based technique is more beneficial to addressing the challenges of sentiment classification, and performs a more informed classification, in which the important features can be easily determined. Our hypothesis is that this approach will extract more relevant relations which

exist between aspects and sentiment terms, and that by taking account of these, our classifier should perform a more informed classification, which should positively influence its accuracy. We also hypothesized that generalizing features using a domain specific lexicon would lead to less sparse features, as opposed to using generic lexicons. It should also ensure that we have taken into account the contextual polarity of the text, other than the prior polarity, which is what is obtainable from generic lexicons. As such, our approach is not only aspect-focused, but also context rich.

6.1 Overview of the System

This method is made up of the following steps:

- Pre-processing
- Target identification
- Feature extraction
- Relation extraction
- Feature selection
- Polarity classification
- Validation from Other domains

We depict these steps in Figure 6.1, and provide a detailed breakdown of the composition of each step in the next section.

6.2 Components of the system

The stages in developing this novel approach have different compositions, which we show in Figure 6.2. This diagram is not a flow diagram, meaning each stage does not follow from the

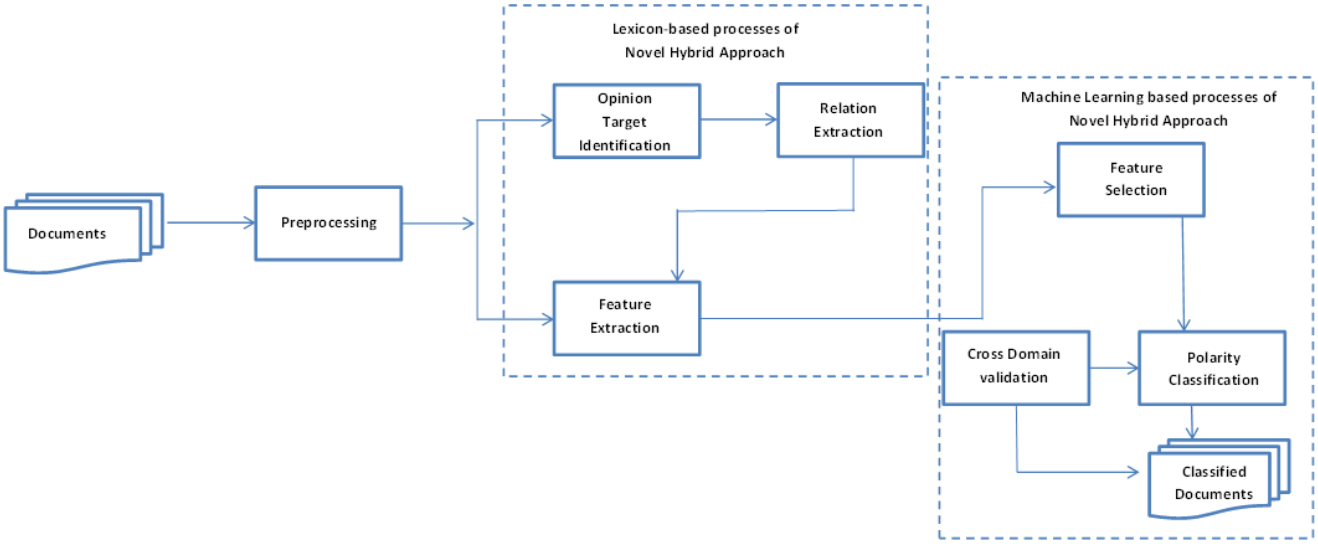

Figure 6.1: Overview of Novel Hybrid Approach Classification

other, but we use this for simplicity, to explain the components of the stages in developing this approach.

In Figure 6.2, we show the various knowledge sources which have been incorporated into the classification process. These will further be explained in the course of this Chapter. The features that we have utilized, as well as the process of the selection of these features make up for the novelty in this work.

The approach works by extracting these knowledge sources from the corpus, and then performing some feature selection, as well as relation extraction, and then these features used to create the feature vectors that the learning classifier is trained on. The SVM_Light classifier is the learning classifier we use in this work, and it is trained and tested on the feature vectors, to produce a classification of documents into two classes, the positive and negative classes.

6.2.1 Feature Extraction

The previous chapters, especially in Chapter Five, shows the use of higher order n-grams, in addition to unigrams. We showed the effect of unigrams in classification and compared these with higher order ngrams,which were bigrams and trigrams.

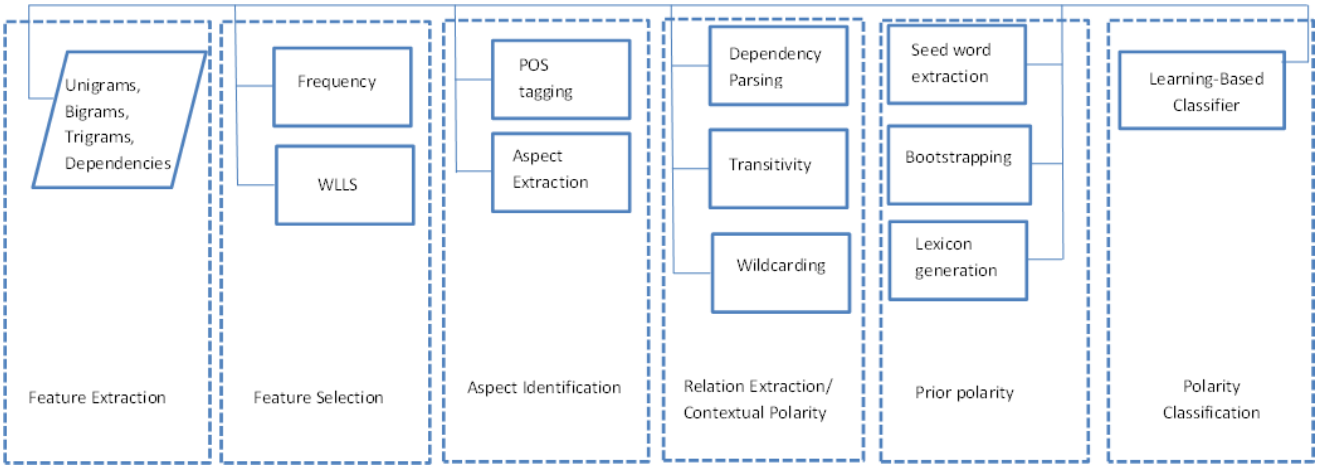

Figure 6.2: Composition of stages in development of Our Hybrid Approach

It has been stated in the literature that bigrams and trigrams capture short range dependencies [87], as well as provide a simple way of capturing context[66], hence, we utilize them in this approach as well, in addition to the other features.

Bigrams and Trigrams are extracted from the corpus using the Ngrams package described in 5.9.1. Punctuations and numbers are not removed, but are left in to provide context. These features are used in the first sets of experiments.

Unigrams have been used as major emotion indicators [87], especially adjectives, whose existence in text has been considered as a sign of subjectivity [36][97].

Though quite common, adjectives have not been seen as the only emotion carrying words across the literature. There have also been adverbs, verbs, as well as nouns [28].

In this approach, we have used adjectives, adverbs and verbs as emotion carrying words.

We perform document level sentiment classification, and use dependency relations to generate transitive relations between aspects and other words in the sentence. Dependency relations are known to capture non-local, long range dependencies amongst the constituents of a sentence [61][66]. Dependency trees are also used to capture context information in the corpus. In trying to capture the deeper linguistic constructs which impact sentiment, it is vital to incorporate the semantic parse of a sentence, which is achieved through dependency parsing and extraction of the dependency relations which exist between the words of a sentence. Dependency relations also in finding the relations between sentiment and aspect

[28]. The dependency relations are extracted using the Stanford Dependency Parser.

6.2.2 Feature Selection

Feature selection is an important aspect of sentiment classification, as irrelevant features introduce noise to the classification process. Not all features from the corpus are necessary to learn the concept of interest, and many of these may end up being redundant, and hence, introducing noise [18]. Ng et al [66] in their work on examining the role played by linguistic knowledge sources mentioned were of the opinion that while not using feature selection does not hurt classification performance when using only unigrams, it is critical when bigrams and trigrams are used in conjunction with unigrams.

In this approach, one of our main motivations is to take advantage of the high accuracy of the machine learning approach, and this accuracy is only achieved from using a representative collection of labelled training data, and through careful selection of the features [11]. It is based on these reasons that we decided to limit the number of features we retained in our classification process.

In the set of extracted n-grams, we only utilized n-grams whose frequency was at least '5' and above, and for the bigrams and trigrams, we implemented two variations, one,using a frequency of from '2' and above, and the other, from '5' and above. We set the same frequency level for dependency relations as well. These numbers are empirically determined, and were chosen after a series of tests, where they gave better results than the values.

WLLS is a scoring scheme which has the merit of capturing relevancy, with respect to each class. The log ratio hence gives a low score to entities which are uniformly distributed over all classes and gives a higher score to those which are more specific to classes. WLLS has been used previously in the area as a selection criterion. Ng et al[66] used it and achieved good results in terms of class distinction, and in obtaining informative features.

For the selection criterion, and in an attempt to extract informative features which are distinctive when it comes to class, we computed the Weighted Log-likelihood Score (WLLS) of each ngram and also of the dependency relations.

After the assignment of the scores, the ngrams and dependency relations are ordered in descending order of their scores, such that those with higher scores are at the top.

We selected a set number of unigrams, bigrams, trigrams and dependency relations with the top WLLS, to make up our feature. An element of '1' was added to the feature vector to represent the presence of this feature in a document, and '0' was added to represent an absence. The machine classifier was then trained on a training set of these vectors, and used to classify the documents.

6.2.3 WLLS - Weighted Log Likelihood Score

The weighted log-likelihood scheme was proposed by Vincent et al in [66]. In this scoring scheme,every unigram, bigram and also trigram is assigned a weighted log-likelihood score with respect to each class of emotion. The scoring scheme captures the relevance of an ngram with respect to each class. A low score is given to ngrams which are uniformly distributed over all classes and high scores are given to ngrams which are specific to each class.

WLLS is defined as:

$$WLLS(w_i, c_j) = P(w_i \mid c_j) log \frac{P(w_i \mid c_j)}{P(w_i \mid \neg c_j)} \tag{6.1}$$

where,

w_i : the unigram or bigram whose score we wish to evaluate

c_j : the class (sentiment) with respect to whom the score is evaluated

$P(w_i \mid c_j)$: the ratio of count of w_i in class c_j to the count of all words in class c_j

$P(w_i \mid \neg c_j)$: the ratio of count of w_i in class $\neg c_j$ to the count of all words in class $\neg c_j$.

6.3 Aspect Identification

It has been established that the major goal of our hybrid approach is to account for the sentiments expressed towards the aspects which make up an entity. The aspects are extracted

in the relation extraction stage. The goal is to avoid including a lot of objective text which might end up clogging the classification process with noise. We identified aspects first and then mined the relations around them in an attempt to include more organized and meaningful features. We worked with the assumption that the sentiment words in a sentence are in reference to a certain aspect of the object and mining the relations between them and the aspect would enable us extract valid feature-aspect pairs.

In an attempt to identify and extract these relations, [66] extracted relation pairs consisting of Adjective-Noun pairs, Subject-Verb pairs, and Verb-Object pairs using the same movie reviews dataset as that used in ours.

We extract frequently occurring nouns in the corpus, but exclude proper nouns to avoid considering aspects like the movie name, as well as the actor names and direction or other crew names.

In addition to this, we set a support for the selected aspects, where a noun can only be considered a possible aspect if it has a support of at least five (5) documents in the dataset where it appears. This number is empirically selected and can be varied.

An approach which is most related to ours, in that it carries out relation extraction around identified nouns and noun phrases is the work of [38]. They however work with customer reviews of products, and use an association miner on the selected nouns and noun phrases. They also carry out two forms of pruning where in one, they remove single words as potential features. We consider nouns as part of our aspects, and extract transitive relations around our aspects.

There has also been a feature-opinion pair mining in the literature, where the suspected aspects are manually tagged by the researchers [115].

What we do first is to tag the documents in the corpus using QTag, and then we extract the nouns that have the highest frequency. We set the number arbitrarily. We then compute the support of the selected nouns and drop those whose support falls short of our required limit. This gives our final list of aspects, around which we extract the relations, to generate our transitive dependencies.

6.4 Relation Extraction/ Contextual Polarity

In this step, the aim is to extract the relations which exist around our selected aspects. We also determine the contextual polarity in this step.

Contextual polarity is captured based on the extracted dependency relations, as well as the transitive dependencies. The transitive dependencies also provide a means to actually capture other relations between words, and possibly, relations between aspects and sentiment.

We explain the experiments in this step in the subsections below.

6.4.1 Transitivity

In order to explain the concept of transitivity, we have to set the context by introducing it through dependency relations.

Dependency relations have been defined as essentially being a set of triplets or triples, each composed of a grammatical relation, and the pair of words from the sentence, among which the grammatical relation holds (rel_i, w_j, w_k), where rel_i denotes the dependency relation between the two words, w_j and w_k [40]. In such a relation, the head word is w_j, and the word it modifies is w_k.

The motivation behind our use of transitive relations is that based on the transitivity of relations, different words in a sentence may be related [28].

One of the aims of this work is to discover the relations between words, and add these to our feature set. This enables us carry out our deep syntactic parsing. Our hypothesis is that discovering these implicit features will help uncover hidden relationships which will positively affect the classification process.

An example of transitivity can be seen in a sentence such as: "this movie is truly great". It is possible that we may not be able to extract the pair (movie, great) from bigrams or the regular lexicalized dependency relations. To extract this relationship, we utilize transitivity.

In our approach, we carry out a systemic method to extract these transitive relations. We first locate the potential aspect in a sentence, then we extract the transitive relations around

the term, up to a depth of '4'. This value was chosen based on a similar experiment by [28], who experimented with searching the dependency tree and extracting relations with no limit, that is, all the relations, and comparing the results to those obtained from extracting the relations up to the depth of '4'. They found that better results were obtained when the depth was set to '4', as opposed to 'all', and this held for other values they experimented with.

Our approach is similar to theirs in that we use transitive relations on the Cornell Movie Review Dataset [1], but differs in that we extract the relations around aspects, while they started their search from subjective words in the tree, and worked back to the aspects. Also, we carry out polarity classification, while they only compared the sentiment expressed by their extracted relations with the sentiment expressed by the full sentence. They also report extracting a lot of redundant relations, which we hope to avoid with our aspect-focused approach.

We implement transitivity by first identifying a sentence that has one of the aspects that we have extracted, then, if there is an emotion carrying word present in the sentence, we extract the transitive relations in that sentence up to the depth of '4'.

We abstractly represent these steps in algorithm 6.3.

6.4.2 Wildcarding

One of the prevalent problems which is encountered where higher order ngrams and dependencies are used is data sparseness [87], in that the feature vectors of some of the documents, particularly short ones have zeroes in them [66].

Wildcarding is a technique in which some words in a dependency tree or subtree are replaced by a generic node, which can match any term.

We hope to tackle this problem of sparseness by wildcarding of very specific terms. In addition to this, some extracted relations may be specific to certain documents, and hence not generalize properly as features. We hypothesize that wildcarding should help create more

[1] `https://www.cs.cornell.edu/people/pabo/movie-review-data/`

```
Data:   The Dependency graphs for sentences in Document D and
        the list of selected aspects (A)
Result: List of transitive dependency relations T
        List of transitive dependency relations with emotive words (TW)

For Sentence i in D do
   if aspect A(j) in Sentence i then
      for dependency relation (j, word(i)) do
          if (word (i), b) then
             Extract Transitive relation (j,b)
             Extract all transitive relations to the depth of '4'
             Add to T
             if b is an adjective or an adverb or a verb then
                  Also add to TW
             end
          end
       end
   end
end
```

Figure 6.3: Transitive relations Algorithm

similarity between the extracted features, and hence generate features which generalize over the dataset.

We perform a number of wildcard variations over dependency relations, and ngrams. The variations are in some ways similar to what has been obtainable in the literature, and in other ways, different.

Joshi and Penstein-Rose [40] use the term "backing-off" for their form of wildcarding in their work with on detecting sentiments in product reviews. They replace the head word in a dependency relation with its part of speech tag, and retain the modifier word, and then also trying a variant where there replace the modifier word with the part of speech tag, and retain the head word.

In a similar attempt, [6] present some form of wildcarding to create annotation graphs and extract subgraph features from these. They work on sentence classification. Pak and Parpubek [70] also perform subgraph wildcarding. Their aim was to wildcard subjects and objects, but they did not wildcard adjectives as verbs, as they believed that these usually

possess sentiment.

We wildcarded these emotive words because while people may not use the same words to express their sentiments, in terms of classifying the negative and positive sentiments, we only need to be aware of the existence of a word depicting this sentiment, not necessarily the term used.

To the best of our knowledge, no one has implemented the different variations of wildcarding that we report in this work for sentiment classification at the document level.

These variations are as follows:

Part of Speech-Word pair In this variation, we replace one of the words in the dependency relations with their part of speech, while retaining the other word. We do this in turn for the head word in the dependency relation, as well as for the modifier word. Hence, there will essentially be pairs such as: $(POS, word)$ and $(word, POS)$.

The motivation behind this is to extract relations where there could be an emotive word in a pair with a word.

Word-Polarity pair In this variation, we replace one of the words in the relationship pair with its prior polarity. We will do this in turn, and carry out texts with features that have been wildcarded this way. Hence, we would be extracting features such as: $(word, POL)$ and $(POL, word)$.

Part of Speech -Polarity pair In this variation of wildcarding, we fully replace the constituent words of the relation pair. We extract relations such as : (POL, POS) and (POS, POL). That is, we replace one component of the pair with its polarity, and the other with its part of speech. We repeat this one more run, but swap the order.

Appending Polarities In this variant of our experiment, we replace a member of the relationship pair with a composition of its part of speech tag, and polarity. Hence, we extract relations such as : $(POL\_POS, word)$ and $(word, POL\_POS)$.

These composite features created by wildcarding are added to the features list of the dependencies, and WLLS will be computed again. They will be sorted in ascending order, and the top features with top WLLS scores will be utilized as features for the classifier.

The goal of this approach is to generate features which are a bit more complex than plain ngrams and dependency relations, and to carry out some basic form of word sense disambiguation through the incorporation of part-of-speech tags. Contextual polarity is also one of the gains that is achieved by some of these forms of wildcarding. Also, it is worth mentioning that this enables us incorporate syntactic and semantic features, as part of our lexical knowledge sources, in the learning based process.

6.5 Prior Polarity

The prior polarity of a term is its semantic orientation irrespective of context. The prior polarity of terms can be obtained from lexicons, such as WordNet, SentiWordNet [27], as well as the subjectivity lexicon [94].

6.6 Lexicon Generation

We have stressed that domain dependency is one of the issues in opinion mining. It is a common problem that a word may have completely different sentiment orientation in different domains [59] [75]. In their study, [68] found that to achieve good sentiment analysis results, it is important to build a domain-specific lexicon, which is related to both the entities/aspects and their sentiment expressions. Rastogi [82] stated that incorporating the information which is in domain specific lexicons can bring about a drastic improvement in accuracy for sentiment analysis.

These lexicons are sometimes manually created, either by hand annotation[66] [75], using web searches [82], manually coming up with a seed list based on common words, and then growing this list from a lexicon, like WordNet through bootstrapping [38][115].

While determining these polarity scores can be done manually, this functions based on human intuition, and hence is subject to bias and can be influenced by the individual's education, as well as cultural background [32]. In addition to being subject to bias, they are also constrained to a small number of terms and are time consuming to create [67]. Because they rely on human intervention, they are costly, and as such, are unsuitable for processing of large volumes of data [32].

Due to some of the above issues, we propose a novel semi-automatic approach to generate a domain specific lexicon, based on the words present in our corpus and how they are used.

6.6.1 Seed Word Extraction

One of the key aspects to developing a domain specific lexicon is the seed word selection process. Our intention is to determine this list semi-automatically, without any interference from human subjects.

To achieve this aim, we have used WLLS, due to its ability to capture terms' relevancy with respect to a certain class.

Out motivation is to create an approach where the emotive words are classified based on the sentiment that they express in the particular domain in focus. WLLS has been shown to work well as a good tool to select informative features.

We extract the top 100 emotive positive words and top 100 emotive negative words. By emotive words, we mean words whose part of speech tag is adjective, adverb, or verb. The figure '100' is arbitrarily decided,and is also based on the choice of the count of emotive seed words number which was made in [115].

These top 200 words form our seed words list which will be used in the bootstrapping approach to generate our domain specific lexicon.

6.6.2 Bootstrapping

Our bootstrapping approach uses the SentiWordNet lexicon. This approach involves extracting the synonyms of the emotive seed words from SentiWordNet, and hence, populating the domain specific lexicon.

Hu and Liu [38] carried out a similar approach in their work on mining and summarizing product reviews. They determined the semantic orientation of adjectives by using a simple, but effective method which utilized the adjective synonyms and antonyms set in WordNet. They stated that adjectives share the same orientation as their synonyms. Mining WordNet for a domain specific lexicon is also used in [115] and [28].

Our approach differs from the above in that we use SentiWordNet for our approach, as it is a lexicon that was generated for sentiment analysis. We also use the subjectivity lexicon as a supporting lexicon. Fei et al [28] who generated two sets of lexicons, one generated from the first sense of a seed word in WordNet and another based on all the senses, did not carry out sentiment classification.

Our bootstrapping approach is also different from what has been reported in the literature, to the best of our knowledge.

In our approach, the bootstrapping approach is carried out by looking up the words in the seed list on SentiWordNet, and checking all its sysnsets under each part of speech. The score of the word in each synset is examined. If in the score triple, the positive score is the highest, then the synonyms in the synset with #1 and #2 are selected and added to the seed list. The other entries from #3 are ignored. The same process is repeated for the negative words. This will give rise to two lists, one of positive words, and the other of negative words. Any words which are not found in SentiWordNet, but exist in the dataset as an emotive word will be looked up in the subjectivity lexicon.

Our lexicon generation approach is depicted in Algorithm 6.4.

```
Data:   List of top 100 Positive emotive words (A) and List of top
        100 Negative emotive words B
Result: Domain Specific Lexicon

 Posscore is positive score, Negscore is negative score and
       Objscore is objective score

For word i in A do
    if i in SentiWordNet then
         if POS(i) = Adjective or Adverb or Verb then
             for synsets 1 To n
                 if Posscore > Negscore AND Posscore > Objscore
                      select word#1 AND word #2
                      add to List A
                      move to next word
                 end
             end
         end
     else
         add word to Temp

For word j in B do
    if j in SentiWordNet then
         if POS(j) = Adjective or Adverb or Verb then
             for synsets 1 To n
                 if Negscore > posscore AND Negscore > Objscore
                      select word#1 AND word #2
                      add to List B
                      move to next word
                 end
             end
         end
     else
         add word to Temp2

Look up words in Temp and Temp2 in the subjectivity lexicon
     if word in Lexicon
        if word is positive
           add to List A
        if word is negative
           add to List B
     else
        discard word
     end
end
```

Figure 6.4: Domain Specific Lexicon Generation Algorithm

6.7 Experimental Setup and Evaluation

We report the experiments carried out in our approach in this section. As a form of preparation for the testing, we carried out preprocessing on our datasets, which included tokenization and stop words removal.

We used SVM Light classifier as our machine learning algorithm, and implemented feature selection as one of the tests.

For evaluation, we used precision, recall and accuracy values, computed over 10-fold cross validation tests. We used review datasets across different domains, and of similar, as well as varying sizes, for the implementation of our approach, as well as for validation tests.

We use ngrams, dependency relations and part-of-speech information, and report here on the components of each test and the results obtained.

For the extraction of ngrams, we used the Ngrams Statistics package from the OpenNLP toolkit [1]. This tool was used to extract the bigrams and trigrams which we also used in Chapter Five of this work.

For part-of-speech tagging, we have used QTag [96], and to extract dependencies, we used the Stanford Dependency parser [2].

6.8 Datasets

The primary dataset we use in this work is a dataset of movie reviews. We chose this domain because movie reviews are easily obtainable and also provide a challenge in terms of sentiment classification [94]. This is due to the extensive plot descriptions which are included in the reviews. In a product review for instance, a number of negative words could more easily imply a bad product than would a number of negative words in a movie review. This could simply the description of a plot in the review, of say, a horror movie , whose overall sentiment expression could still be positive. In providing some explanation to this,

[1] `http://search.cpan.org/~tpederse/Text-NSP-1.27/lib/Text/NSP.pm`
[2] `http://nlp.stanford.edu/software/lex-parser.shtml`

[97] explained that movie reviews have two aspects, the elements; which includes the actors and the events, as well as the style and art of the movie; a unified whole.

We use the same Cornell Movie reviews dataset [74] which we have used in Chapter Four, and Chapter Five of this work. This dataset has been widely used in document level sentiment analysis, and in some sentence level sentiment analysis. It comprises of 1000 negative and 1000 positive movie reviews, and is hence a balanced dataset.

The second dataset which we also use in this work is used for validation of our approach, and it is the Multi-Domain Sentiment Dataset [1]. Used in [12], this dataset consists of product reviews which were taken from Amazon.com, and comprise of reviews from various product types or domains.

6.9 Baseline

The baseline that we will use is the higher-order ngrams classifier in Chapter Five. We re-write the figures in Table6.1.

Table 6.1: Higher Order Ngrams classifier

Ngrams	Precision	Recall Accuracy	
Unigrams + Bigrams + Trigrams	89.7%	78.9%	84.9%

6.10 Experiments

6.10.1 Feature Selection

After the extraction of our ngrams, we select the top ranking features using the WLLS scoring scheme. We select the top 5000 unigrams, 2500 bigrams and 2500 trigrams . These figures were determined after iterative testing with various figures. We only select unigrams which occur at least '3' times in the text, and select bigrams and trigrams with at least an occurrence of "2" in the dataset.

[1] `http://www.cs.jhu.edu/~mdredze/datasets/sentiment/`

Using these selected features, we use term presence as frequency, with the SVM Light classifier and run a 10-fold cross validation classification, and show the results in Table 6.2.

Table 6.2: Higher Order Ngrams with Feature Selection

Features	Precision	Recall	Accuracy
Unigrams + Bigrams	86.6%	85.3%	86%
Unigrams + Bigrams + Trigrams	87.5%	85.8%	86.7%

We observe that there is an improvement to the accuracy when WLLS is used to select the top ranking features to be used for classification. The results reported are the average values obtained from 10-fold cross validation. We put this down to the efficiency of feature selection.

6.10.2 Incorporating Dependencies

As part of our approach to incorporate more linguistic features, we incorporate the extracted dependencies in our feature set, and rerun the classifier on this. We select the top 1500 dependencies, using iterative tests as well to determine the best figure,and as a further filtering step, we only retain dependency relations that occur at least "2" times in the dataset.

We run various variations with unigrams, bigrams and trigrams, with the extracted dependencies. We present the results from these tests in Table 6.3.

Table 6.3: Higher Order Ngrams with dependency relations

Features	Precision	Recall	Accuracy
Unigrams + Dependencies	87.9%	85.8%	86.9%
Unigrams + Bigrams + Dependencies	88%	85.5%	86.9%
Unigrams + Bigrams + Trigrams + Dependencies	88%	85.5%	87%

6.10.3 Transitive Dependencies

To incorporate aspect focused sentiment into our learning-based classifier, we utilize transitive dependencies. These dependencies are extracted around selected top nouns in the dataset.

To determine these popular nouns, we extract frequent nouns from the dataset, with a support value of '5' documents. This means that even if a noun has a high frequency in the dataset, if it does not occur across at least '5' documents, then we consider that it may not be an aspect. Hu and Liu [38] used a p-support value of '5' sentences for their product reviews, and reported non-opinion targets being selected, so we are using a different approach to that. We iteratively tested different support values before deciding on '5' documents. The intention here is to select an aspect which is an aspect of the dataset as a whole, and not just an aspect of a certain document.

We selected a variety of top nouns frequencies, including 100, 75, 50, 20 and 5. We then extracted transitive dependency relations around these potential targets, to a depth of '4'. We found that the best result came from using the top '50' nouns, so we retained these.

We ran our classifier again combining the ngrams with these transitive dependencies. We present the results obtained in Table6.4.

Table 6.4: Transitive Dependency relations with Ngrams

Features	Precision	Recall	Accuracy
Unigrams + TDeps	88.7%	85.9%	87.5%
Unigrams + Bigrams + TDeps	88.4%	85.9%	87.3%
Unigrams + Bigrams + Trigrams + TDeps	88.9%	85%	87.1%

6.10.4 Composite features

This section presents the results of our experiments on testing the effects of including the part-of-speech information as well as the prior polarity information in the sentiment classifier. Since part-of-speech offers a basic form of word sense disambiguation, we test to see what effect this information would have over the classification process, in comparison to the baseline. We also conducted these tests to identify the contribution that having more generalized dependency features would make.

6.10.4.1 Part-of-speech, word pair

Here, we first wildcarded the dependent word in the dependency relation pair, by replacing it with its part-of-speech. We selected the dependencies with the top WLLS score, using the same scores that we have been using so far. We classified the documents again, and present the result from this in Table 6.5. Next, we wildcarded the head word in the dependency relation pair, replacing it with its part-of-speech tag, and repeated the same tests. These results are also presented in Table 6.5.

Table 6.5: Part-of-speech - Word pair

Features	Precision	Recall	Accuracy
Unigrams + Deps (head, POS (dep))	86.7%	84.8%	85.9%
Unigrams + Bigrams + Deps (head, POS (dep))	86.8%	84.8%	85.9%
Unigrams + Bigrams + Trigrams + Deps (head, POS (dep))	87.5%	83.4%	86.6%
Unigrams + Deps (POS(head),dep)	86.7%	85.2%	86%
Unigrams + Bigrams + Deps (POS(head),dep)	87%	85.2%	86.2%
Unigrams + Bigrams + Trigrams + Deps (POS(head), dep)	87.7%	85.5%	86.7%

6.10.4.2 Word, Polarity pair

This composite feature is an attempt to incorporate the polarity information of terms into the classifier, and assess the effects of these features. Additionally, we also assess the performance of our domain specific lexicon, in comparison to SentiWordNet, in terms of prior polarity information. This wildcarding scheme is also aimed at incorporating contextual polarity, as this prior polarity information is added to extracted dependency pairs.

We perform two variations of tests, one involving wildcarding the head word in the dependency relation pair, and the other involving wildcarding the dependent word. We report the results of these tests in two tables, Table 6.6 and Table 6.7. Table 6.7 holds the results of wildcarding the head word, and using the SentiWordNet lexicon to determine the polarity values, and also, using the domain specific lexicon to do same.

Table 6.6 shows the results obtained from wildcarding the dependent word , and also determining the polarity using the SentiWordNet lexicon and the domain specific lexicon.

We represent the SentiWordNet results using (SWN), and the domain specific lexicon results using (SYN).

Table 6.6: Word - Polarity pair $(head, POL(dep))$

Features	Precision	Recall	Accuracy
Unigrams + Deps (SWN))	87%	84.9%	86.1%
Unigrams + Bigrams + Deps (SWN)	86.8%	85.1%	86%
Unigrams + Bigrams + Trigrams + Deps (SWN))	87.5%	85.4%	86.6%
Unigrams + Deps (SYN)	86.6%	85.1%	86%
Unigrams + Bigrams + Deps (SYN)	86.6%	85.2%	86%
Unigrams + Bigrams + Trigrams + Deps (SYN)	87.8%	85.6%	86.9%

Table 6.7: Word - Polarity pair $(POL(head), dep)$

Features	Precision	Recall	Accuracy
Unigrams + Deps (SWN))	86.8%	85.4%	86.2%
Unigrams + Bigrams + Deps (SWN)	86.8%	85.4%	86.2%
Unigrams + Bigrams + Trigrams + Deps (SWN))	87.6%	85.6%	86.7%
Unigrams + Deps (SYN)	86.9%	85.7%	86.4%
Unigrams + Bigrams + Deps (SYN)	87%	85.4%	86.3%
Unigrams + Bigrams + Trigrams + Deps (SYN)	87.9%	85.7%	86.9%

6.10.4.3 Part-of-speech, Polarity pair

This composite feature was created by wildcarding both components of the dependency relations, that is, the head word and the dependent. We present the result of this test in two tables, Table 6.9 and Table 6.8.

In Table 6.8 we show the results of replacing the head word of the dependency relation with its part-of-speech, and also replacing the dependent word with its polarity value. We show the results obtained from using SentiWordNet lexicon (SWN) to determine the polarity value, and using our domain specific lexicon (SYN) in determining the polarity value.

In Table 6.9, we show the results obtained from replacing the head word with its polarity value, and the dependent word with its part-of-speech tag. Again, the results are presented from using the SentiWordNet lexicon to determine the polarity value and also, from using the domain specific lexicon. Same as the previous subsection, we use (SWN) for SentiWordNet, and (SYN) for the domain specific lexicon.

Table 6.8: POS of head word - Polarity of dependent

Features	Precision	Recall	Accuracy
Unigrams + Deps (SWN))	86.4%	85.2%	86%
Unigrams + Bigrams + Deps (SWN)	87%	85.3%	86.3%
Unigrams + Bigrams + Trigrams + Deps (SWN))	87.7%	85.5%	86.7%
Unigrams + Deps (SYN)	87%	85.1%	86.2%
Unigrams + Bigrams + Deps (SYN)	85.3%	85.1%	86.3%
Unigrams + Bigrams + Trigrams + Deps (SYN)	88.1%	85.5%	86.9%

Table 6.9: Polarity of head word- POS of dependent

Features	Precision	Recall	Accuracy
Unigrams + Deps (SWN))	87.2%	85.5%	86.4%
Unigrams + Bigrams + Deps (SWN)	87.1%	85.2%	86.2%
Unigrams + Bigrams + Trigrams + Deps (SWN))	88%	85.4%	86.9%
Unigrams + Deps (SYN)	87.1%	85.5%	86.4%
Unigrams + Bigrams + Deps (SYN)	87.4%	85.2%	86.4%
Unigrams + Bigrams + Trigrams + Deps (SYN)	87.7%	85.5%	86.7%

6.10.4.4 Appending the polarities

We split this into two sections, one section containing appending the polarity to the part of speech tag with one component of the dependency relation pair, while the other section covers appending the polarity with the part of speech tag of one component of the dependency relation pair, and replacing the other component with the part of speech tag only.

Polarity-part of speech, Word pair We created this composite feature by replacing the head word in the dependency pair with its polarity value appended with the part of speech tag, and do the same for the dependent word in the pair as well. We represent these pairs with (POL_POS of head, dependent), and (head, POL_POS of dep). As we have done previously, we use both SentiWordNet and the domain specific lexicon to determine the polarity information, and report these in Tables 6.10 and 6.11.

Polarity-Part of speech, Part of speech of word pair This composite feature was created by replacing the head word of the dependency pair with the polarity value, appended with the part-of-speech tag, and replacing the dependent word with its part of speech. Then,

Table 6.10: *POL_POS* of head word - dependent word

Features	Precision	Recall	Accuracy
Unigrams + Deps (SWN))	87%	85.6%	86.1%
Unigrams + Bigrams + Deps (SWN)	87.1%	85.7%	86.4%
Unigrams + Bigrams + Trigrams + Deps (SWN))	87.6%	86%	86.9%
Unigrams + Deps (SYN)	86.5%	85.1%	85.9%
Unigrams + Bigrams + Deps (SYN)	86.9%	85.1%	86.1%
Unigrams + Bigrams + Trigrams + Deps (SYN)	87.4%	85.5%	86.8%

Table 6.11: Head-*POL_POS* of dependent word

Features	Precision	Recall	Accuracy
Unigrams + Deps (SWN))	86.9%	84.5%	86%
Unigrams + Bigrams + Deps (SWN)	87%	84.8%	86%
Unigrams + Bigrams + Trigrams + Deps (SWN))	87.8%	85.5%	86.8%
Unigrams + Deps (SYN)	86.8%	85%	86%
Unigrams + Bigrams + Deps (SYN)	86.7%	84.9%	85.9%
Unigrams + Bigrams + Trigrams + Deps (SYN)	87.5%	84.5%	86.6%

the same was done for the dependent word, by replacing the dependent word with its polarity value appended with the part-of-speech tag, and replacing the head word with its part-of-speech tag.

We represent these pairs with (*POL_POS* of head, POS of dependent), and (POS of head, *POL_POS* of dependent).

We present the results obtained from these tests in Tables 6.12 and 6.13, using the SentiWordNet lexicon, and the domain specific lexicon.

Table 6.12: *POL_POS* of head word - POS (dependent word)

Features	Precision	Recall	Accuracy
Unigrams + Deps (SWN))	86.8%	85.5%	86.2%
Unigrams + Bigrams + Deps (SWN)	87%	85.5%	86%
Unigrams + Bigrams + Trigrams + Deps (SWN))	88.1%	85.4%	86.9%
Unigrams + Deps (SYN)	86.9%	85.5%	86.3%
Unigrams + Bigrams + Deps (SYN)	87.4%	85.2%	86.4%
Unigrams + Bigrams + Trigrams + Deps (SYN)	87.7%	85.5%	86.7%

Table 6.13: POS of (head)-*POL_POS* of dependent word

Features	Precision	Recall	Accuracy
Unigrams + Deps (SWN))	86.8%	85%	86%
Unigrams + Bigrams + Deps (SWN)	87%	85%	86%
Unigrams + Bigrams + Trigrams + Deps (SWN))	87.8%	85.4%	86.7%
Unigrams + Deps (SYN)	86.9%	85.2%	86.1%
Unigrams + Bigrams + Deps (SYN)	87.1%	85.3%	86.3%
Unigrams + Bigrams + Trigrams + Deps (SYN)	87.9%	85.7%	86.9%

6.11 Validation from other datasets

As part of an evaluation of the performance of our approach, we run it on other datasets. The datasets we choose are product reviews, which were collected from Amazon.com, and which form part of the Multi-Domain Sentiment Datasets[1].

We select two datasets. One, is an unbalanced dataset and serves two purposes, it will help us validate the performance of our approach on a product dataset, as well as on an unbalanced dataset.

The datasets are the books dataset, and the computer and video games dataset. The books dataset contains 1000 negative and 1000 positive reviews, while the computer and video games dataset contains 1000 positive reviews, and 457 negative reviews.

We evaluate the following;

- The performance of higher order ngrams on the datasets
- The role placed by feature selection
- The system performance when transitive dependencies are introduced

In our tests on the movie reviews dataset, we had assessed the performance of unigrams, bigrams and trigrams, when used together, and then we had assessed how this was affected by feature selection. We run the test on the books dataset and the computer and video games, using unigrams, bigrams and trigrams, then implement feature selection, using the same values as we used in the movie reviews. Hence, we used 5000 unigrams, with a minimum

[1] `https://www.cs.jhu.edu/~mdredze/datasets/sentiment/`

occurrence of '3' times in the dataset, 2500 bigrams, with a minimum occurrence of '2'times in the dataset, and 2500 trigrams with a minimum occurrence of '2' times in the dataset.

We extract dependencies using the Stanford Dependency parser, as was done with the movies dataset, and select 1500 dependencies, with a minimum occurrence of '2' in the dataset. We then extract transitive dependencies, leaving the feature number at around '50' features.

We use unigrams and transitive dependencies as the only test with the transitive dependencies, as this gave the highest level of accuracy in the experiments on the movie reviews domain.

The results for the run on the books dataset is given in Table 6.14, and the results for the run on the computer and video games dataset is given in Table 6.15. We refer to feature selection in the tables with "FS".

Table 6.14: Validation with Books dataset

Features	Precision	Recall	Accuracy
Unigrams + Bigrams + Trigrams(No FS)	75.2%	85.8%	78.6%
Unigrams + Bigrams + Trigrams (with FS)	76.9%	84.4%	79.4%
Unigrams + Bigrams + Trigrams + Deps(with FS)	77.2%	84.6%	79.6%
Unigrams + Transitive Deps(with FS)	76.6%	83.8%	79%

Table 6.15: Validation with Computer and Video games dataset

Features	Precision	Recall	Accuracy
Unigrams + Bigrams + Trigrams(No FS)	85.3%	98.7%	87.3%
Unigrams + Bigrams + Trigrams (with FS)	77.4%	99.8%	79.8%
Unigrams + Bigrams + Trigrams + Deps(with FS)	76.1%	99.8%	78.3%
Unigrams + Transitive Deps(with FS)	82.4%	99.2%	84.8%

A close look at the table shows that the computer and video games dataset had quite a drop in accuracy when feature selection was applied. This was not the case with the books dataset. We note that both datasets have different properties, and writing style. The computer and video games dataset was also unbalanced, with quite a large number of positive reviews, in comparison with negative reviews.

6.12 Result Analysis

We analyze the results obtained from our approach, and compare them directly with the other works in the literature that have proposed and implemented various hybrid approaches. Table 6.16 shows the various accuracies that have been reported in the literature, which we will compare ours with. Some of the reported work have derived new generalizing features through wildcarding (features we refer to as composite features), while others have also focused on determining the polarity classification of reviews as a composition of the sentiments expressed with relation to each aspect of the review.

In their hybrid approach which involved sentiment classification in tweets, by interpolating sentiment lexicon score into unigram vectors, [41] arrived at the conclusion that the add-on lexicon for out-of-vocabulary words, used in addition to SentiWordNet led to an improvement in the classification accuracy, on average, compared to using the original public lexicon. One of the reasons why we utilised our domain specific lexicon was to provide contextual polarity information, which also could not be found from generic lexicons. From the results we obtained from comparing the performance of the classification with semantic orientation of words obtained from our domain specific lexicon, against those obtained from SentiWordNet, we did not observe a big change in the accuracy. These results are shown in Tables 6.6 to 6.13. Though from the results we can see that there are instances where we obtain a slightly better result from using our domain specific lexicon, these differences are however not statistically significant.

Gezici et al [33] also focused on classifying documents through context information from different types of sentences, with the belief that this will lead to better differentiation of reviews with different polarities. The TF-IDF scores and weighted TF-IDF scores of the words in these sentences were used as features for classifiers. Like [41], they worked on the intuition that sentiment classification based on SentiWordNet only lacked context information. They report on the importance of sentences in bridging the gap between word polarity determination and whole document classification.

Table 6.16: Hybrid approaches comparison

Method	Domain	Aspects	Wildcarding	Accuracy (%)
Our Novel Hybrid approach	Movies	Yes	Yes	87.5
Aravindan and Ekbal (2014)[5]	Products	Yes	Yes	79.67
Pak and Paroubek (2011)[70]	Movies	Yes	Yes	85.1
Kaewpitakkun et al (2002)[41]	Tweets	No	No	81.2
Gezici et al (2012)[33]	Hotel reviews	No	No	81.45
Kennedy and Inkpen (2006)[43]	Movie reviews	No	Yes	86.2
Mullen and Collier (2004)[62]	Movie and Record reviews	Yes	No	Movies (86.0), Records (87.0)
Mudinas et al (2012)[60]	Software and Movie reviews	Yes	Yes	Software reviews (89.6), Movie reviews (82.30)
Matsumoto et al (2005)[55]	Movie reviews	No	No	Dataset 1 (93.7), Dataset 2 (87.3)
Joshi and Rose (2009)[40]	Product reviews	Yes	Yes	67.9
Arora et al (2010)[6]	Movie reviews	Yes	Yes	76.93
Hu and Liu (2004)[38]	Product reviews	Yes	No	84.2
Ng et al (2006)[66]	Movie reviews	No	Yes	90.5

Matsumoto et al [55] report a state-of-the-art accuracy from exploiting the syntactic relations between words in sentences for document sentiment classification, through the use of word subsequences and dependency subtrees as features for an SVM classifier with extensive feature selection. These features were used in combination with unigrams and bigrams, and concluded that feature selection played a big role in the good classification results obtained.

We also found this to be true, as we reported a statistically significant improvement in accuracy when feature selection was performed, as opposed to not having feature selection.

These results are shown in Tables 6.1 and 6.2. They reported observing improvements where bigrams were included in the feature set, as opposed to using just unigrams. We found that our best performing system was the combination of unigrams with transitive dependencies. While they did not adversely affect the classification accuracy, bigrams did not improve the accuracy of the classification as dependency relations did.

The highest improvement in accuracy as reported by [55] was obtained from addition of the dependency subtrees, as the word subsequences did not bring about any improvement. They however did not perform aspect based classification, which we have done in our approach. We believe that this high accuracy was obtained based on the density of the feature set offered by the dependency subtrees.

Also not performing aspect based classification, [66]'s work on examining the role of linguistic sources in automatic sentiment classification gave quite a high accuracy. They reported not observing improvements from adding dependency relations to a unigram + bigram classifier, which contradicts what we found. The dependency relations they selected were adjective-noun relations, verb-object relations, and subject-verb relations. They also performed feature selection with WLLR and report that not performing feature selection hurts the classifier performance, which we found to be true. They report that adding bigrams and trigrams to a unigram only classifier significantly improves performance, but we did not observe this in our work, as shown in Table 6.3. They concluded that dependency relations are only somewhat useful for the task of polarity classification when bigrams and trigrams are not used. We are inclined to support this, as we found that we obtained a better classification accuracy when unigrams only were used with our transitive dependencies.

In order to make a more pointed analysis of the aspect-based systems and the systems that utilised wildcarding in order to obtain more generic features, we analyse these separately below.

Aspect-based classification One of the contributions our hybrid approach makes is that it classifies documents by taking into account the sentiments expressed on the different

aspects in the document. This is done through the novel use of transitive dependencies for document polarity classification. Some of the work listed in Table 6.16 have presented hybrid approaches which have focused on taking the sentiment expressed about each aspect into consideration.

Aravindan and Ekbal [5] worked on classifying product reviews, using association rule mining to identify the most characteristic features of a product.Nouns and noun phrases are used as the default features, and they report encountering a lot of redundant features. To incorporate contextual information, the three preceeding and next three words surrounding the target phrase are selected as features.

To include contextual information in our approach, we exploited the syntactic relations that exists between sentences, as we believe this is a more structured approach than arbitrarily selecting neighbouring words. We believe that the redundant features were as a result of adding the neighbouring words. We also believe that using transitive relations is more structured, and incorporates ordered syntactic information and relations. We believe that this is why our results are better than what was obtained in [5].

Pak and Paroubek [70] exploited the syntactic structure of a sentence in the sentiment classification of sentences. They used subgraphs from the dependency tree of a parsed sentence as features for an SVM classifier. Reporting their highest accuracy to have been obtained from a modified TF-IDF weighting of the features, they also tried binary weighting scheme and frequency counting scheme, and report that the binary weighting scheme outperformed the frequency scheme.

From our experiments, we have found that the binary weights (where '1' is entered in the feature vector if a feature is present, and '0' is entered otherwise)outperforms the frequency weighting scheme. We have however found that the binary weights also outperform TF-IDF. They also report using a subgraph of size '1'. We extracted transitive relations to the depth of '4', a number decided based on [29]'s results, and based on our observation that using a depth of '1' adversely affected the classification accuracy, extracting many non relevant features. We note that we report an accuracy higher than what was obtained in their work.

Using sentiment associated with different aspects within a movie review, [62] utilized phrases as well as 'secondary' information sources, which included topic relation as well as proximity relation. They assigned semantic values to these and in turn incorporated these as features for SVM modelling. Different SVMs were trained on the various sources and combined with SVMs based on ngrams. They report obtaining their best results from the combination of SVMs trained on these different sources and lemmatized unigrams and unigrams. This report is in line with our observation that combining unigrams with transitive dependencies gave our best performing system.

pSenti [60] is an established aspect-focused hybrid sentiment classification system, which integrates both lexicon-based and learning based approaches into opinion mining. Aspects considered are nouns and noun phrases. pSenti is claimed to detect and measure sentiment at the concept level and provide structured and readable aspect-oriented outputs due to the built-in sentiment lexicon and linguistic rules. To tackle domain dependency, [60]exclude domain specific aspect words from the machine learning step. In our approach, the sentiment classification is centred around the aspect words, and to reduce domain dependency we do not utilize any external domain specific sources. We believe that ours is a better technique because these aspect words will change from one domain to another, and so given that sentiment bearing terms are domain specific, it is better to develop a framework that each dataset can be fit into, rather than attempt to carry out cross-domain sentiment classification, especially aspect-focused classification. Their system performed slightly worse than their purely learning based system, with the accuracy falling from 82.3% to 86.85%.

The impressive performance of unigrams, when used in combination with other syntactic features or patterns was further confirmed in the work of [6] who report their best result to be from a combination of genetic programming with unigram and subgraph features. They found that the addition of the subgraph features to unigrams without feature selection leads to decrease in performance.

Composite features for feature generalization Wildcarding has been used in hybrid approaches to enable feature generalization and tackle sparsity in some cases. We used wildcarding to generate composite features, not just to obtain less sparse feature vectors, but also to tackle domain specificity. A number of variations of composite features were tested, and the results given in this Chapter. Table 6.17 brings together the best results we obtained from utilizing a combination of composite features, ngrams and transitive dependencies. The polarities were also obtained from our domain specific lexicon, and we compared these to SentiWordNet. Though the differences we obtained were not statistically significant when compared with each other, there was a statistically significant difference between the obtained results and our baseline score of 84.9%.

Table 6.17: Best Results using Composite features

Features	Composite feature	Accuracy
Uni + Bi + Tri + Dep	(POS(head), dep)	86.7%
Uni + Bi + Tri + Dep(SYN)	(head,POL(dep))	86.9%
Uni + Bi + Tri + Dep(SWN)	(head,POL(dep))	86.6%
Uni + Bi + Tri + Dep(SYN)	(POL(head),dep)	86.9%
Uni + Bi + Tri + Dep(SWN)	(POL(head),dep)	86.7%
Uni + Bi + Tri + Dep(SYN)	(POS(head),POL(dep))	86.9%
Uni + Bi + Tri + Dep(SWN)	(POS(head),POL(dep))	86.7%
Uni + Bi + Tri + Dep(SWN)	(POL(head),POS(dep))	86.9%
Uni + Bi + Tri + Dep(SYN)	(POL(head),POS(dep))	86.7%
Uni + Bi + Tri + Dep(SWN)	(POL_POS(head),dep)	86.9%
Uni + Bi + Tri + Dep(SYN)	(POL_POS(head),dep)	86.8%
Uni + Bi + Tri + Dep(SWN)	(head,POL_POS(dep))	86.8%
Uni + Bi + Tri + Dep(SYN)	(head,POL_POS(dep))	86.6%
Uni + Bi + Tri + Dep(SWN)	(POL_POS(head),POS(dep))	86.9%
Uni + Bi + Tri + Dep(SYN)	(POL_POS(head),POS(dep))	86.7%
Uni + Bi + Tri + Dep(SYN)	(POS(head),POL_POS(dep))	86.9%
Uni + Bi + Tri + Dep (SWN)	(POS(head),POL_POS(dep))	86.7%

Overall, the highest accuracy we obtained using composite features was 86.9%. In some of the tests, our domain specific lexicon showed a slight gain over using the more generic SentiWordNet lexicon, as shown in Table 6.17. We also explored the effects of wildcarding the head word, against wildcarding the dependent word in the dependency relation. We

found that wildcarding the head word by replacing it with its part-of-speech gave a slightly better result than wildcarding the dependent word.

Vincent et al [66] who reported a state-of-the-art accuracy through examining the contribution of linguistic knowledge sources combined with a machine learning classifier manually incorporated the polarity information in one of the tested feature sets. They hand annotated each adjective in bigrams, trigrams and dependency relations with their polarity, added these to the ngrams and dependency relations again and extracted those with high WLLR scores as features. They reported an increase in accuracy from this annotation, and put this down to the having less sparse features than ngrams for the classification. From examination of our feature vectors, we observed that the features are less sparse than they were without wildcarding, though we did not observe a statistically significant change in the accuracy. Our observations are in line with what is reported in this work, in terms of a reduction is feature vector sparsity through wildcarding.

Arora et al [6] used annotation graph representation, and combined unigrams with part-of-speech and the dependency relation type as labels for the nodes in the dependency tree. They reported using the extra edges which represented the dependency relation labels as an alternative to putting wildcards on words. They found that a combination of these subgraph features with unigrams and genetic programming gave their best result. We found that the combination of ngrams with wildcarded dependency relations gave a higher accuracy than using just unigrams with the wildcarded features. Higher order ngrams appear to contribute to the classification process together with the wildcarded dependency relations. We hypothesize that subgraphs on their own already introduce a lot of syntactic information to the feature set, and hence, the information provided by higher order ngrams are already incorporated in the extracted subgraphs.

Also wildcarding subgraphs of the dependency tree, [70] replaced the aspect words in their subgraphs with wildcards. They did not wildcard verbs and adjectives, as they believe that these are sentiment bearing terms and should not be wildcarded. We wildcard the sentiment bearing terms in our work, replacing them with their polarity information and

believe this would provide more generalized features as well as preserve our aspects for the aspect-based classification. We believe the system implemented by [70] does not take the properties of the domain in focus into consideration.

Joshi and Rose [40] "backed-off" certain words in the dependency relation pair in order to generate more generalizable features for aspect based product review classification. They alternately replaced each part of the dependency relation pair with its part-of-speech tag, and concluded that "backing-off" the head words in the dependency pair produced more generalizable features, and created useful patterns. They report the best result from their classification with SVM light classifier to be from the feature set comprising of unigrams and dependency relations with the head words replaced with their part-of-speech tag. We found that replacing the head word with its part-of-speech tag also outperformed replacing the dependent word with its part-of-speech, but this difference was not statistically significant. An observation of the feature vectors did show that those with the head words replaced with the part-of-speech tagged were slightly less sparse than those with the dependent words replaced.

Kennedy and Inkpen [43] incorporated contextual valence shifters in their feature set by appending identifiers to bigrams to signify if they were intensifiers, diminishers or negations. They combined these with unigrams and reported that there was a slight improvement from incorporating these features, over just using unigrams with regular bigrams. They reported that the improvement was not statistically significant but showed that these features did have an effect on the polarity classification of texts. Their findings are in line with our findings on incorporating contextual information. Our composite features outperform the ngrams only baseline.

From our results, it can be observed that in some of the instances, incorporating domain specific information does have an effect on the classification accuracy, and also that the results can be slightly better depending on if the head word is wildcarded, or if the dependent word is wildcarded. These findings are in-line with some of the work reported here, and in contrast to others. We believe that composite features lead to a more generalized feature set, which

reduces the sparseness of feature vectors.

6.13 Summary

In this Chapter, we present our hybrid approach to polarity classification. We present transitive relations extracted around opinion targets (aspects), and also introduce certain composite features which we utilize in showing the roles of certain features in syntactic based classification approaches, and to incorporate contextual polarity. We go on to introduce our domain specific lexicon, which we created as a means of tackling the domain dependency problem of sentiment terms.

We have analysed the roles of feature selection, dependencies, transitive dependencies, as well as composite features. We also assess the performance of our domain specific lexicon, as well as the role and effects of incorporating part of speech information in the classifier. We found that our domain specific lexicon performed at par with the generic opinion lexicon, SentiWordNet.

We have observed the effects of weighting schemes, like term presence, and its ability to improve the calculation accuracy. With the composite features, we have been able to observe the contribution which is made by bigrams and trigrams, as higher order dependencies.

We obtain results which are comparable with what has been reported in the literature on this particular movie review dataset. In terms of aspect-focused classification, we have also obtained an accuracy of 87.5%, through our use of transitive dependencies. We find that this is an impressive accuracy score, when compared with what is obtainable in the literature, where aspect-based classification has been implemented.

We note that our approach performs much better than the baseline, which was made up of frequent higher order ngrams. Our composite features also performed better than the baseline, but did not offer improvements in the value of accuracy over the performance of introducing syntactic features into the higher order ngrams classifier.

A further test of validation performed using datasets from other domains showed that

feature selection, if not done properly could negatively influence the accuracy of the classification. The different characteristics of the datasets should be taken into consideration. The results from these other datasets upholds the claim that a classifier that is trained on one domain would not necessarily give the same performance if transferred as it is, to another domain.

A combination of transitive dependencies with unigrams also leads to improvements in the computer and video games domain, over feature selection on higher order ngrams, and dependency relations. We believe that this has to do with the characteristics of the dataset, such as its size and writing style.

We elaborate more on these in the Discussion Chapter of this work.

Chapter 7

Discussion

This work sought to answer a number of research questions, including which sentences within a document could be extracted as a summary of the document, and express the same sentiment polarity as the document.

We did not focus on summarization in this work, though we implemented some aspects of it as a prerequisite experiment to assess if certain segments of a document were good enough to be used as representatives of the whole. The results we obtained were not favourable, so we moved our focus from summarization.

We explored the lexicon-based and machine learning based approaches, and confirmed that using the SentiWordNet lexicon as a means of performing pure lexicon-based polarity classification gives a classification accuracy in line with what is obtainable in the literature. We also found that the last sentences performed better in capturing the polarity of the document set, than the first sentences did. This was in-line with our hypothesis that reviewers tend to summarize their sentiments at the end of a review, which has also been noted in some cases in the literature. We found that the last sentences expressed this sentiment when used together, as our experiments on determining if any single one of them was responsible, all gave accuracies lower than that obtained when they were used together as a summary.

For the pure learning based approach, our intuition that using higher order ngrams would greatly improve the accuracy, was found to not be true, as we found that the weighting

scheme used with the higher order ngrams had more of an influence on the classification accuracy than the ngrams themselves. We found that classifying summaries extracted with a benchmark tool that is used for extracting efficient summaries for topic classification, did not produce a result which performed in parallel with the classification result of full documents. This proved our intuition, and what is reported in the literature, that sentiment classification and topic classification follow very different rules.

In our novel hybrid approach, we addressed our research question on identifying and extracting the aspects within a document about which sentiment is expressed, through the use of transitive relations extracted from lexicalized dependencies. Our intuition was that taking account of such intrinsic sentiment polarities will lead to notable improvements in the accuracy of sentiment classification. We hypothesized that transitive relations which were extracted from lexicalized dependencies obtained from the dependency tree, would lead to very notable improvements in accuracy over the lexicalized dependencies. We found this to not hold, as the change in accuracy was not statistically significant.

Still, our approach of introducing various linguistic features into a learning-based classifier lead to significant improvements over the baseline, which was a learning based classifier utilizing higher order ngrams with term presence weighting. We use Wilcoxon's signed rank test to determine the statistical significance of the classification accuracy, with $\alpha = 0.05$.

We observed the best result from using unigrams with transitive dependencies. The improvement in the accuracy rose from 84.9% to 87.5%, was significant with a p-value of 0.012. The same was observed for introducing dependency relations into the higher order ngrams classifier, with feature selection, the accuracy level rose from 84.9% to 87%, a significant increase, with a z-value of -2.2509.

There was a slight increase in the accuracy from introducing the transitive dependencies, from 87% to 87.5%, but this increase was not statistically significant. We believe that introducing a more sophisticated approach to choosing the opinion targets could lead to the transitive dependencies making more of a contribution to the accuracy level. We used frequent nouns, with a support level of at least of '5' documents. We believe that using

an algorithm like an association rule miner could potentially help extract aspects more efficiently, in relation to the dataset. We also believe that using noun phrases could be explored as well.

We have however observed that the transitive dependencies do have an effect on the classification accuracy. We have also observed that the number of aspects about which we extract transitive dependencies around has an influence on the classification accuracy. This is to be expected, as the number of opinion targets is a variable. This further points to utilizing a more complex algorithm to identify these aspects, and then extracting the transitive dependencies around them.

Obtaining a large improvement in accuracy was not the only motivation for this approach. We note that an extension can be made to the use of transitive dependencies, to generate aspect focused summaries, which can further be utilized in recommender systems. From an inspection of the extracted transitive dependencies, we have noted that certain relations between aspects and sentiment bearing terms which modify them have been extracted. These relations were not extracted directly with the lexicalized dependencies.

One of the research questions we set out to answer was to explore what features can influence the sentiment classification process. In addressing this, we examined the influence of a number of features, and feature combinations. We utilized composite features which are obtained from generalizing syntactic features, syntactic features themselves, and ngrams. We found that unigrams and dependency relations appear to strongly influence the accuracy of sentiment classification. We find this to be in-line with what is reported in a number of published work in the literature, especially the influence of unigrams.

Additionally, we found that feature selection plays a key role in the sentiment classification, especially when a learning based classifier is used. Feature selection led to a high rise in the accuracy value in our movie reviews classification. This is inline with what is reported in the literature about the importance of feature selection to learning based classifiers. We observed that the accuracy rose from 84.9%, which was the baseline score to 86.7%, with a p-value of 0.02.

We also have observed that the weighting scheme used in the determining the features for the document vectors can greatly influence the performance accuracy of the classifier. We observed that using term presence made significant gains over using normalized frequencies, especially when higher order ngrams are used. This goes to prove what has been reported in the literature about the importance of the occurrence of a sentiment term.

To address domain specificity, we implement wildcarding of the extracted dependencies and transitive dependencies, replacing words with their polarity and part-of-speech information. These features formed what we refer to as composite features, and are more generalized. To ensure that context information is incorporated, we created a domain specific lexicon and extracted the polarity of terms for our features from this. This addresses our research question on word correlations which could influence polarity determination. As we had expected, the feature vectors with these composite features were less sparse, but we also observed that the accuracy fell from what we obtained from using transitive dependencies without composite features. A possible reason for this could be due to overfitting, or could be that the features were not distinguishable enough for the classifier.

We obtained an accuracy of 86.9% from the classification using composite features, which was significantly better than the baseline. Additionally, we observed that our domain specific lexicon performed at par with the widely used generic opinion lexicon SentiWordNet.

An important research question we posed was on developing an approach that could generalize well to other problem domains. To implement this, we designed our approach to be focused on the given corpus. We steered clear of introducing any external domain specific resources. This, we believe ensures that our approach is portable, and can be easily implemented on another domain, if necessary.

Testing the classifier on different datasets leads to performances which differ from the movie reviews dataset. This is to be expected, as the MDSA dataset has different characteristics, is from a different domain, and a look at the dataset shows that it contains shorter reviews. The shortness of these reviews would potentially lead to feature vectors which are more sparse.

The books dataset showed an improvement in accuracy from 78.6%, when no feature selection is used, to 79.4%, on using feature selection with higher order dependencies. It however does not show a significant change in accuracy when dependencies are added, or when unigrams are used with transitive dependencies. Like the movie reviews, we expect that the books reviews will also be difficult to classify as it contains plot descriptions as well. We suspect that the feature selection did not make as much difference as it did in the movie reviews domain because we used the exact same feature values for the higher order ngrams and the dependencies, as we did for the movie reviews, which are longer reviews. We also suspect that transitive dependencies did not make a difference because we kept the opinion targets/aspect value at '50', and they should be less features in a book review, as books are very different from movies, and would hardly have reviews where there are many identifiable features.

The classification of the computer and video games dataset recorded a higher accuracy where higher order ngrams are used without feature selection, than when feature selection is applied. We again believe that this is due to the fact that these reviews have differing characteristics, and hence would probably fare better if feature selection values determined on them are utilized, than using the values determined from a classifier implemented on another domain.

We also observed an increase in accuracy when the unigrams were used with transitive dependencies. This dataset, like the movie reviews dataset has a number of aspects, hence using transitive dependencies appears to introduce new relevant information into the classifier than the higher order ngrams with feature selection. This shows that transitive dependencies, if properly extracted, with regard to the characteristics of each dataset would influence the classification accuracy positively.

Overall, this work makes the following contributions:

- We propose and implement a novel hybrid sentiment classification approach which carries out aspect-based sentiment classification through the use of transitive dependencies. Transitive dependencies increase the chances of extracting a relation between

an aspect and a sentiment word. To the best of our knowledge, this has not been reported in any literature on document level sentiment classification.

- We show that these transitive relations can lead to an improvement in the classification accuracy, especially if the aspects are properly extracted.

- We introduce and implement various composite features whose aim is to incorporate more linguistic knowledge to the classification approach. We show that these features attain an accuracy higher than the baseline, and generate a more context rich feature set.

- We develop a domain specific lexicon from which we determine the polarities of our composite features, as a means of incorporating contextual polarity in our classification approach. We observe that this lexicon provides comparable classification with generic lexicons.

- We implement a sentiment classification approach that can easily be adaptable to other domains, other than those tested in this work.

These contributions also address the prevalent challenges in sentiment classification that have been raised in Chapter One of this work. Rather than considering the document as a bag-of-words, we have designed an approach that is more structured, and takes into account the sentiments that are associated with the aspects within the document. We also take advantage of the syntactic information incorporated in sentences, through our use of transitive dependencies. Transitive dependencies also assist in extracting relations between aspects and sentiment, which may not be extracted with regular lexicalized dependencies. This addresses the challenge of sentiment terms not being in close proximity with the words they modify.

We have shown that sentiment classification differs from topic classification by the results we obtained from classifying summaries extracted with OTS, which has been used for benchmarking in topic summarization.

We have also included contextual sentiment in our classification approach, through the incorporation of contextual polarity using composite features.

Finally, we have explored a number of features and feature combinations for classification, and analysed their performances in the design of our approach.

7.1 Further Work

As further work, we believe that utilizing a more sophisticated algorithm to extract the aspects would improve the aspect identification process and positively influence the transitive dependency relations extracted.

The domain specific lexicon which we have developed can also be used to carry out sentiment intensity testing, as we note that it holds highly polar words, and we can extract documents with a high number of these words and determine their intensity level based on the frequency of these words in them.

We also hope to look into determining feature selection values for the MDSA datasets, setting the values per dataset, before extracting transitive dependencies around them and utilizing these to implement sentiment classification.

Additionally, we hope to extend our aspect-focused classification by extracting aspects and their associated sentiments in order to create aspect-sentiment summaries which can further be used in recommendation systems.

References

[1] Alekh Agarwal and Pushpak Bhattacharyya. Sentiment analysis: A new approach for effective use of linguistic knowledge and exploiting similarities in a set of documents to be classified. *Proceedings of the International Conference on Natural Language Processing (ICON)*, 2005. 22

[2] Alina Andreevskaia and Sabine Bergler. When Specialists and Generalists Work Together: Overcoming Domain Dependence in Sentiment Tagging. *ACL*, pages 290–298, 2008. 39, 40

[3] N Anitha, B Anitha, and S Pradeepa. Sentiment classification approaches–A Review. *International Journal of Innovations in Engineering and Technology (IJIET)*, 3(1):22—-31, 2013. 30

[4] Michelle Annett and Grzegorz Kondrak. A comparison of sentiment analysis techniques: Polarizing movie blogs. In *Advances in artificial intelligence*, pages 25–35. Springer, 2008. 32, 46, 93, 104, 105, 123, 124

[5] Siddharth Aravindan and Asif Ekbal. Feature Extraction and Opinion Mining in Online Product Reviews. *2014 International Conference on Information Technology*, pages 94–99, 2014. 23, 38, 40, 48, 154, 156

[6] Shilpa Arora, Elijah Mayfield, Carolyn Penstein-Rosé, and Eric Nyberg. Sentiment classification using automatically extracted subgraph features. *Proceedings of the*

NAACL HLT 2010 Workshop on Computational Approaches to Analysis and Generation of Emotion in Text, pages 131—-139, 2010. 39, 44, 137, 154, 157, 159

[7] Giuseppe Attardi and Maria Simi. Blog Mining Through Opinionated Words. In *TREC*, 2006. 22

[8] Anthony Aue and Michael Gamon. Customizing Sentiment Classifiers to New Domains : A Case Study. *Proceedings of Recent Advances in Natural Language Processing RANLP*, 49(2):207–18, 2005. 45

[9] Lukasz Augustyniak. Simpler is better? Lexicon-based ensemble sentiment classification beats supervised methods. *Advances in Social Networks Analysis and Mining (ASONAM), 2014 IEEE/ACM International Conference*, 2014. 39, 40

[10] Akshat Bakliwal, Pi yush Arora, and Vasudeva Varma. Entity Centric Opinion Mining from Blogs. In *Proc. of 24th Int. Conf. on Computational Linguistics*, pages 53–64. Citeseer, 2012. 22, 96

[11] Pavel Blinov, Maria Klekovkina, Eugeny Kotelnikov, and Oleg Pestov. Research of lexical approach and machine learning methods for sentiment analysis. *Computational Linguistics and Intellectual Technologies*, 2(12):48—-58, 2013. 17, 18, 24, 25, 37, 63, 65, 107, 108, 127, 132

[12] John Blitzer, Mark Dredze, and Fernando Pereira. Biographies, bollywood, boomboxes and blenders: Domain adaptation for sentiment classification. *ACL*, 7:440–447, 2007. 40, 120, 144

[13] Marco Bonzanini, Miguel Martinez-Alvarez, and Thomas Roelleke. Opinion summarisation through sentence extraction: an investigation with movie reviews. In *Proceedings of the 35th international ACM SIGIR conference on Research and development in information retrieval*, pages 1121–1122. ACM, 2012. 96, 97, 99, 114, 115

[14] Oisin Boydell and Barry Smyth. From social bookmarking to social summarization. In *Proceedings of the 12th international conference on Intelligent user interfaces - IUI '07*, page 42, New York, New York, USA, jan 2007. ACM Press. 116

[15] Pranali Chilekar, Swati Ubale, Pragati Sonkambale, Reema Panarkar, and Gopal Upadhye. Sentiment analysis on news articles using Natural Language Processing and Machine Learning Approach . *International journal of Emerging Trend in Engineering and Basic Sciences*, 2(1):464–468, 2015. 15

[16] TC Chinsha and Shibily Joseph. A syntactic approach for aspect based opinion mining. *Semantic Computing (ICSC), 2015 IEEE International Conference on*, pages 24–31, 2015. 10, 12, 48

[17] Hang Cui, Vibhu Mittal, and Mayur Datar. Comparative experiments on sentiment classification for online product reviews. *AAAI*, 6:1265–1270, 2006. 30

[18] Yan Dang, Yulei Zhang, and Hsinchun Chen. A Lexicon-Enhanced Method for Sentiment Classification: An Experiment on Online Product Reviews. *IEEE Intelligent Systems*, 25(4):46–53, jul 2010. 17, 19, 20, 33, 37, 83, 108, 127, 132

[19] Taner Danisman and Adil Alpkocak. Feeler: Emotion classification of text using vector space model. *AISB 2008 Convention Communication, Interaction and Social Intelligence*, 1:53, 2008. 25, 36

[20] Sanjiv R Das and Mike Y Chen. Yahoo! for amazon: Sentiment extraction from small talk on the web. *Management Science*, 53(9):1375–1388, 2007. 31

[21] Maitrayee Dasgupta. Study of different algorithms in Sentiment Analysis and the existing Issues. *American Journal Of Advanced Computing*, 2(2):55–60, 2015. 29

[22] Sajib Dasgupta and Vincent Ng. Topic-wise, sentiment-wise, or otherwise?: Identifying the hidden dimension for unsupervised text classification. *Proceedings of the 2009*

Conference on Empirical Methods in Natural Language Processing: Volume 2-Volume 2, 2009. 32, 124

[23] Kushal Dave, Steve Lawrence, and David M. Pennock. Mining the peanut gallery. *Proceedings of the twelfth international conference on World Wide Web - WWW '03*, page 519, 2003. 45, 121

[24] Kerstin Denecke. Are SentiWordNet scores suited for multi-domain sentiment classification? In *Digital Information Management, 2009. ICDIM 2009. Fourth International Conference on*, pages 1–6. IEEE, 2009. 22, 23, 104, 105

[25] Adnan Duric and Fei Song. Feature selection for sentiment analysis based on content and syntax models. *Decision Support Systems*, 53(4):704–711, 2012. 11, 13, 23

[26] Andrea Esuli and Fabrizio Sebastiani. SENTIWORDNET: A high-coverage lexical resource for opinion mining. *Evaluation*, pages 1–26, 2007. 22

[27] Andrea Esuli, Fabrizio Sebastiani, and Via Giuseppe Moruzzi. SENTIWORDNET : A Publicly Available Lexical Resource for Opinion Mining. *Proceedings of LREC 2006*, pages 417–422, 2006. 4, 21, 22, 40, 139

[28] Zhongchao Fei, Xuanjing Huang, and Lide Wu. Mining the relation between sentiment expression and target using dependency of words. In *Proc. 20th Pacific Asia Conf. on Language, Information and Computation (PACLIC20), Wuhan, China*, pages 257–264, 2006. 24, 42, 44, 48, 60, 61, 65, 68, 113, 131, 132, 135, 136, 141

[29] Zhongchao Fei, Xuanjing Huang, and Lide Wu. Mining the relation between sentiment expression and target using dependency of words. *PACLIC20: Coling*, 2006. 156

[30] Michael Gamon. Sentiment classification on customer feedback data: noisy data, large feature vectors, and the role of linguistic analysis. *Proceedings of the 20th international conference on Computational Linguistics*, page 841, 2004. 45, 59

[31] Krutika M Gandecha, Vikas S Gondane, and Vivek R Shelke. A Survey on Opinion Mining. *ijrest.net*, 3(1):531–539, 2013. 13, 14, 15, 46

[32] Gang Li and Fei Liu. A clustering-based approach on sentiment analysis. In *2010 IEEE International Conference on Intelligent Systems and Knowledge Engineering*, pages 331–337. IEEE, nov 2010. 22, 24, 26, 35, 36, 140

[33] Gizem Gezici, Berrin Yanikoglu, Dilek Tapucu, and Yücel Saygin. New features for sentiment analysis: Do sentences matter? In *CEUR Workshop Proceedings*, volume 917, pages 5–15, 2012. 38, 39, 40, 153, 154

[34] Alaa Hamouda and Mohamed Rohaim. Reviews classification using sentiwordnet lexicon. *World Congress on Computer Science and Information Technology*, 2011. 23, 104, 105

[35] Ali Harb, Michel Plantié, Gerard Dray, Mathieu Roche, François Trousset, and Pascal Poncelet. Web opinion mining. In *Proceedings of the 5th international conference on Soft computing as transdisciplinary science and technology - CSTST '08*, page 211, New York, New York, USA, oct 2008. ACM Press. 104, 105

[36] Vasileios Hatzivassiloglou and Kathleen R. McKeown. Predicting the semantic orientation of adjectives. In *Proceedings of the eighth conference on European chapter of the Association for Computational Linguistics* -, pages 174–181, Morristown, NJ, USA, jul 1997. Association for Computational Linguistics. 19, 20, 25, 26, 67, 131

[37] I. Hemalatha, Dr. G. P. Saradhi Varma, and Dr. A. Govardhan. Sentiment Analysis Tool using Machine Learning Algorithms. *International Journal of Emerging Trends & Technology in Computer Science (IJETTCS)*, 2(2):105–109, 2013. 12, 28, 29, 30

[38] Minqing Hu and Bing Liu. Mining and summarizing customer reviews. In *Proceedings of the tenth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 168–177. ACM, 2004. 20, 22, 24, 25, 48, 65, 68, 113, 114, 134, 139, 141, 146, 154

[39] Thorsten Joachims. Svmlight: Support vector machine. *SVM-Light Support Vector Machine http://svmlight. joachims. org/, University of Dortmund*, 19(4), 1999. 110, 111

[40] Mahesh Joshi and Carolyn Penstein-Rosé. Generalizing Dependency Features for Opinion Mining. *Proceedings of the ACLIJCNLP 2009 Conference Short Papers on ACLIJCNLP 09*, (August):313–316, 2009. 57, 59, 135, 137, 154, 160

[41] Yongyos Kaewpitakkun, Kiyoaki Shirai, and Masnizah Mohd. Sentiment Lexicon Interpolation and Polarity Estimation of Objective and Out-Of-Vocabulary Words to Improve Sentiment Classification on Microblogging. *Proceedings of the 2014 Pacific Asia conference on Language, Information and Computation*, pages 204–213, 2014. 38, 40, 153, 154

[42] Jaap Kamps, M J Marx, Robert J Mokken, and Maarten De Rijke. Using Wordnet to Measure Semantic Orientations of Adjectives. In *In Proceedings of LREC-04, 4th international conference on language resources and evaluation, Lisbon, PT*, volume 4, page 11151118. European Language Resources Association (ELRA), 2004. 20

[43] Alistair Kennedy and Diana Inkpen. Sentiment Classification of Movie Reviews Using Contextual Valence Shifters. *Computational Intelligence*, 22(2):110–125, may 2006. 38, 40, 154, 160

[44] Aurangzeb Khan. Sentiment classification by sentence level semantic orientation using sentiwordnet from online reviews and Blogs. *International Journal of Computer Science \& Emerging Technologies*, 2(4), 2011. 21, 23, 47

[45] Jungi Kim, Jin-Ji Li, and Jong-Hyeok Lee. Discovering the discriminative views: measuring term weights for sentiment analysis. 1:253–261, 2009. 36, 122

[46] Soo-Min Kim and Eduard Hovy. Determining the Sentiment of Opinions. In *Proceedings of the 20th international conference on Computational Linguistics - COLING '04*,

pages 1367–es, Morristown, NJ, USA, 2004. Association for Computational Linguistics. 47

[47] Soo-Min Kim and Eduard Hovy. Identifying and analyzing judgment opinions. *Proceedings of the main conference on Human Language Technology Conference of the North American Chapter of the Association of Computational Linguistics*, 2006. 20

[48] DK Kirange and Ratnadeep R Deshmukh. Emotion Classification of Restaurant and Laptop Review Dataset : Semeval 2014 Task 4. *International Journal of Computer Applications*, 113(6):17–20, 2015. 10

[49] Seema Kolkur, Gayatri Dantal, and Reena Mahe. Study of Different Levels for Sentiment Analysis. *International Journal of Current Engineering and Technology*, 5(2):768–770, 2015. 46

[50] Julia Kreutzer and Neele Witte. Opinion Mining Using SentiWordNet. *stp.lingfil.uu.se Uppsala University*, 2013. 83

[51] David D Lewis and Marc Ringuette. A comparison of two learning algorithms for text categorization. *Third annual symposium on document analysis and information retrieval*, 33:81—-93, 1994. 29

[52] Peifeng Li, Qiaoming Zhu, and Wei Zhang. A Dependency Tree Based Approach for Sentence-Level Sentiment Classification. In *Software Engineering, Artificial Intelligence, Networking and Parallel/Distributed Computing (SNPD), 2011 12th ACIS International Conference on*, pages 166–171. IEEE, 2011. 43, 47

[53] Bing Liu. Sentiment Analysis and Subjectivity. *Handbook of Natural Language Processing*, (1):1–38, 2010. 46, 47

[54] Bing Liu, Minqing Hu, and Junsheng Cheng. Opinion observer: analyzing and comparing opinions on the web. *Proceedings of the 14th international conference on World Wide Web*, pages 342—-351, 2005. 47, 118

[55] Shotaro Matsumoto, Hiroya Takamura, and Manabu Okumura. Sentiment classification using word sub-sequences and dependency sub-trees. *Advances in Knowledge Discovery and Data Mining*, pages 301–311, 2005. 38, 40, 44, 47, 154, 155

[56] Ryan McDonald, Kerry Hannan, Tyler Neylon, Mike Wells, and Jeff Reynar. Structured models for fine-to-coarse sentiment analysis. *Annual Meeting-Association For Computational Linguistics*, 45(1):432, 2007. 59

[57] George A Miller. WordNet: a lexical database for English. *Communications of the ACM*, 38(11):39–41, 1995. 21

[58] Antonio Moreno-Ortiz and Chantal Pérez Hernández. Lexicon-based sentiment analysis of twitter messages in spanish. *Procesamiento del lenguaje natural*, 50:93–100, 2013. 37

[59] Andrius Mudinas, Dell Zhang, and Mark Levene. Combining lexicon and learning based approaches for concept-level sentiment analysis. *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining - WISDOM '12*, pages 1–8, 2012. 27, 37, 65, 107, 108, 127, 139

[60] Andrius Mudinas, Dell Zhang, and Mark Levene. Combining lexicon and learning based approaches for concept-level sentiment analysis. In *Proceedings of the First International Workshop on Issues of Sentiment Discovery and Opinion Mining - WISDOM '12*, pages 1–8, New York, New York, USA, aug 2012. ACM Press. 39, 40, 154, 157

[61] Subhabrata Mukherjee and Pushpak Bhattacharyya. Feature specific sentiment analysis for product reviews. *Computational Linguistics and Intelligent Text Processing*, pages 475—-487, 2012. 54, 60, 65, 131

[62] Tony Mullen and Nigel Collier. Sentiment analysis using support vector machines with diverse information sources. In *Conference on Empirical Methods in Natural Language Processing*, 2004. 39, 40, 154, 157

[63] Tetsuji Nakagawa, Kentaro Inui, and Sadao Kurohashi. Dependency tree-based sentiment classification using CRFs with hidden variables. *Human Language Technologies: The 2010 Annual Conference of the North American Chapter of the Association for Computational Linguistics*, pages 786—-794, 2010. 47

[64] Tetsuya Nasukawa and Jeonghee Yi. Sentiment analysis: Capturing favorability using natural language processing. *Proceedings of the 2nd international conference on Knowledge capture*, pages 70–77, 2003. 61

[65] Andrew Y Ng, Michael I Jordan, and Yair Weiss. On spectral clustering: Analysis and an algorithm. *Advances in neural information*, 2:849–856, 2002. 124

[66] Vincent Ng, Sajib Dasgupta, and S M Arifin. Examining the role of linguistic knowledge sources in the automatic identification and classification of reviews. In *Proceedings of the COLING/ACL on Main conference poster sessions*, pages 611–618. Association for Computational Linguistics, 2006. 34, 35, 37, 39, 41, 42, 54, 57, 68, 113, 120, 121, 122, 131, 132, 133, 134, 136, 139, 154, 155, 159

[67] Bruno Ohana and Brendan Tierney. Sentiment classification of reviews using SentiWordNet. *9th. IT & T Conference*, 2009. 26, 63, 82, 83, 97, 104, 105, 114, 140

[68] Sara Owsley, Sanjay Sood, and Kristian J Hammond. Domain Specific Affective Classification of Documents. *AAAI Spring Symposium: Computational Approaches to Analyzing Weblogs*, pages 181—-183, 2006. 42, 139

[69] Alexander Pak and Patrick Paroubek. Twitter as a Corpus for Sentiment Analysis and Opinion Mining. *Lrec*, pages 1320–1326, 2010. 15

[70] Alexander Pak and Patrick Paroubek. Text representation using dependency tree subgraphs for sentiment analysis. *Database Systems for Adanced Applications*, pages 323–332, 2011. 37, 44, 45, 55, 57, 137, 154, 156, 159, 160

[71] Prabu Palanisamy, Vineet Yadav, and Harsha Elchuri. Serendio : Simple and Practical lexicon based approach to Sentiment Analysis. In *Second Joint Conference on Lexical and Computational Semantics (*SEM), Volume 2: Seventh International Workshop on Semantic Evaluation (SemEval 2013)*, volume 2, pages 543–548, 2013. 17, 82

[72] Bo Pang and Lillian Lee. A sentimental education: Sentiment analysis using subjectivity summarization based on minimum cuts. In *Proceedings of the 42nd annual meeting on Association for Computational Linguistics*, page 271, 2004. 32, 37, 45, 46, 62, 113, 123, 124, 125

[73] Bo Pang and Lillian Lee. Seeing stars: Exploiting class relationships for sentiment categorization with respect to rating scales. In *Proceedings of the 43rd Annual Meeting on Association for Computational Linguistics*, pages 115–124. Association for Computational Linguistics, 2005. 3, 32, 45, 46

[74] Bo Pang, Lillian Lee, and Shivakumar Vaithyanathan. Thumbs up?: sentiment classification using machine learning techniques. In *Proceedings of the ACL-02 conference on Empirical methods in natural language processing-Volume 10*, pages 79–86. Association for Computational Linguistics, 2002. 25, 26, 27, 28, 29, 30, 31, 35, 45, 64, 108, 109, 110, 111, 114, 121, 122, 123, 124, 125, 144

[75] K Paramesha and KC Ravishankar. Optimization Of Cross Domain Sentiment Analysis Using Sentiwordnet. *arXiv preprint arXiv:1401.3230*, 2013. 66, 139

[76] Sungrae Park, Wonsung Lee, and Il-Chul Moon. Efficient extraction of domain specific sentiment lexicon with active learning. *Pattern Recognition Letters*, 56:38–44, 2015. 21, 23

[77] Viraj Parkhe and Bhaskar Biswas. Aspect Based Sentiment Analysis of Movie Reviews: Finding the Polarity Directing Aspects. *2014 International Conference on Soft Computing and Machine Intelligence*, pages 28–32, 2014. 45, 49

[78] Sangita Patel and Jignya Choksi. A Survey of Sentiment Classification Techniques. *IJSER*, 3(3):1–6, 2015. 12, 28

[79] Braja Gopal Patra, Amitava Kundu, Dipankar Das, and Sivaji Bandyopadhyay. Classification of Interviews - A Case Study on Cancer Patients. *Proceedings of the 2nd Workshop on Sentiment Analysis where AI meets Psychology*, 1(December 2012):27–36, 2012. 22, 32

[80] R Mohana Priya, R Vikas Pareek, and V Saravanaprabhu. Sentiment Analysis and Opinion mining using SentiwordNet: A Survey. *History*, 30(131):283–288, 2015. 11, 15, 21, 45

[81] J. Ross Quinlan. Induction of decision trees. *Machine learning*, 1(1):81–106, 1986. 29

[82] SSK Rastogi, Rohit Singhal, and Anil Kumar. An Improved Sentiment Classification using Lexicon into SVM. 95(1):37–42, 2014. 18, 19, 37, 39, 40, 66, 127, 139

[83] Ellen Riloff and Janyce Wiebe. Learning extraction patterns for subjective expressions. *Proceedings of the 2003 conference on Empirical methods in natural language processing* -, 10:105–112, 2003. 19, 21, 40

[84] Ellen Riloff, Janyce Wiebe, and Theresa Wilson. Learning subjective nouns using extraction pattern bootstrapping. *Proceedings of the seventh conference on Natural language learning at HLT-NAACL*, 4, 2003. 25

[85] Nadav Rotem. Open text summarizer, 2003. 116

[86] Hassan Saif, Yulan He, Miriam Fernandez, and Harith Alani. Contextual semantics for sentiment analysis of Twitter. *Information Processing & Management*, pages 1–15, 2015. 13, 15, 18, 21, 48

[87] Shrutiranjan Satapathy and Sumit Bhagwani. Capturing Emotions in Sentences. *Retrieved March*, 15:2012, 2012. 34, 54, 131, 136

[88] Richa Sharma, Shweta Nigam, and Rekha Jain. Opinion mining of movie reviews at document level. *arXiv preprint arXiv:1408.3829*, 2014. 46

[89] Philip J Stone, Dexter C Dunphy, and Marshall S Smith. The General Inquirer: A Computer Approach to Content Analysis. 1966. 21

[90] Carlo Strapparava and Alessandro Valitutti. WordNet Affect: an Affective Extension of WordNet. *LREC*, 4:1083–1086, 2004. 40

[91] Maite Taboada, Caroline Anthony, and Kimberly Voll. Methods for creating semantic orientation dictionaries. *Proceedings of the 5th International Conference on Language Resources and Evaluation (LREC), Genova, Italy*, 2006. 21, 26

[92] Maite Taboada, Julian Brooke, Milan Tofiloski, Kimberly Voll, and Manfred Stede. Lexicon-Based Methods for Sentiment Analysis. *Computational Linguistics*, 37(2):267–307, jun 2011. 17, 24, 25, 26, 37, 67

[93] Songbo Tan and Jin Zhang. An empirical study of sentiment analysis for chinese documents. *Expert Systems with Applications*, 34(4):2622–2629, 2008. 35

[94] Tun Thura Thet, Jin-Cheon Na, and Christopher S G Khoo. Aspect-based sentiment analysis of movie reviews on discussion boards. *Journal of Information Science*, 36(6):823–848, 2010. 26, 34, 41, 48, 58, 65, 67, 139, 143

[95] Gong Tianxia. Processing Sentiments and Opinions in Text : A Survey. *World*, 2007. 4

[96] Dan Tufis and Oliver Mason. Tagging romanian texts: a case study for qtag, a language independent probabilistic tagger. *Proceedings of the First International Conference on Language Resources and Evaluation (LREC)*, 1:589—-596, 1998. 143

[97] Peter D Turney. Thumbs up or thumbs down?: semantic orientation applied to unsupervised classification of reviews. In *Proceedings of the 40th annual meeting on as-*

sociation for computational linguistics, pages 417–424. Association for Computational Linguistics, 2002. 19, 25, 33, 45, 46, 47, 61, 67, 104, 105, 131, 144

[98] Sneha Pradeep Vanjari and V D Thombre. Classification Techniques: A Survey. *International Journal of Science and Research*, 4(2):2317–2320, 2015. x, 17, 20, 45, 46

[99] Casey Whitelaw, Casey Whitelaw, Navendu Garg, Navendu Garg, Shlomo Argamon, and Shlomo Argamon. Using appraisal groups for sentiment analysis. *Proceedings of the 14th ACM international conference on Information and knowledge management - CIKM '05*, page 625, 2005. 45

[100] Janyce Wiebe, Theresa Wilson, Rebecca Bruce, Matthew Bell, and Melanie Martin. Learning Subjective Language. *Computational Linguistics*, 30(3):277–308, sep 2004. 35, 109

[101] Fu Xianghua, Liu Guo, Guo Yanyan, and Wang Zhiqiang. Multi-aspect sentiment analysis for Chinese online social reviews based on topic modeling and HowNet lexicon. *Knowledge-Based Systems*, 37:186–195, 2013. 33

[102] Jyotika Yadav. A Survey on Sentiment Classification of Movie Reviews. 3(1):340–343, 2014. 48

[103] Shailesh Kumar Yadav. Sentiment Analysis and Classification: A Survey. pages 113–121, 2015. 46

[104] Yang Yang, Ruifan Li, and Yanquan Zhou. A hybrid approach to identifying sentiment polarity for new words. *Wireless Communications, Vehicular Technology, Information Theory and Aerospace \& Electronic Systems (VITAE), 2014 4th International Conference on*, pages 1—-5, 2014. 19, 20, 21, 23, 48

[105] Yiming Yang and Jan O Pedersen. A comparative study on feature selection in text categorization. *ICML*, 97:412–420, 1997. 35

[106] V. A. Yatsko and T. N. Vishnyakov. A method for evaluating modern systems of automatic text summarization. *Automatic Documentation and Mathematical Linguistics*, 41(3):93–103, jun 2007. 116

[107] Jeonghee Yi, Tetsuya Nasukawa, Razvan Bunescu, and Wayne Niblack. Sentiment analyzer: Extracting sentiments about a given topic using natural language processing techniques. *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 427—-434, 2003. 3

[108] Hong Yu and Vasileios Hatzivassiloglou. Towards answering opinion questions. *Proceedings of the 2003 conference on Empirical methods in natural language processing* -, (3):129–136, 2003. 47

[109] Jianxing Yu, Zheng-Jun Zha, Meng Wang, and Tat-Seng Chua. Aspect Ranking : Identifying Important Product Aspects from Online Consumer Reviews. *Computational Linguistics*, pages 1496–1505, 2011. 48

[110] Sheng Yu and Subhash Kak. A survey of prediction using social media. *arXiv preprint arXiv:1203.1647*, 2012. 29

[111] Zheng-Jun Zha, Jianxing Yu, Jinhui Tang, Meng Wang, and Tat-Seng Chua. Product aspect ranking and its applications. *Knowledge and Data Engineering, IEEE Transactions on*, 26(5):1211—-1224, 2014. 48

[112] Ethan Zhang and Yi Zhang. UCSC on TREC 2006 blog opinion mining. In *Text Retrieval Conference*, 2006. 22

[113] Hailong Zhang, Wenyan Gan, and Bo Jiang. Machine Learning and Lexicon based Methods for Sentiment Classification : A Survey. pages 262–265, 2014. 19, 20, 28, 32, 93

[114] Yan Zhao, Suyu Dong, and Jing Yang. Effect Research of Aspects Extraction for Chinese Hotel Reviews Based on Machine Learning Method . 9(3):23–34, 2015. 31

[115] Li Zhuang, Feng Jing, and Xiao-Yan Zhu. Movie review mining and summarization. In *Proceedings of the 15th ACM international conference on Information and knowledge management - CIKM '06*, pages 43–50. ACM, 2006. 24, 37, 42, 67, 134, 139, 140, 141

[116] Sapna Zol and Preeti Mulay. Analyzing Sentiments for Generating Opinions (ASGO) -A New Approach. *Indian Journal of Science and Technology*, 8(February):206–211, 2015. 14, 15, 29, 30